Better Predictions when Models are Wrong or Underspecified

Thijs van Ommen

Better Predictions when Models are Wrong or Underspecified

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus prof.mr. C.J.J.M. Stolker,
volgens besluit van het College voor Promoties
te verdedigen op woensdag 10 juni 2015
klokke 16.15 uur

door

Matthijs van Ommen

geboren te Rotterdam
in 1984

Promotiecommissie

Promotor:
prof.dr. P.D. Grünwald

Overige leden:
prof.dr. R.D. Gill
prof.dr. N.L. Hjort (University of Oslo)
prof.dr. A.W. van der Vaart

These investigations were performed at the Centrum Wiskunde & Informatica (CWI) and were supported by Vici grant 639.073.04 from the Netherlands Organization for Scientific Research (NWO). Part of the work was done while the author was visiting UC San Diego.

Cover design by Gracia Murris
Printed and bound by Ipskamp Drukkers
ISBN: 978-94-6259-689-4

This dissertation is based on the following publications and manuscripts:

- Chapter 2 is based on

  > T. van Ommen. Combining predictions from linear models when training and test inputs differ. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence (UAI 2014)*, pages 653–662, 2014.

- Chapters 3, 4 and 5 are based on the technical report

  > P. D. Grünwald and T. van Ommen. Inconsistency of Bayesian inference for misspecified linear models, and a proposal for repairing it. *arXiv preprint arXiv:1412.3730*, 2014.

- Chapters 6 and 7 are joint, as yet unpublished, work with T. Feenstra, P. D. Grünwald and W. M. Koolen. Chapter 6 is a significant extension of

  > T. E. Feenstra. Conditional prediction without a coarsening at random condition. Master's thesis, Leiden University, 2012. Thesis adviser: P. D. Grünwald.

- Chapter 8 is based on work that is not currently available elsewhere.

Contents

List of Figures

Chapter 1

Introduction

Both statistics and machine learning deal with the question of how humans and computers can *learn* from data. In large amounts of data, we hope to find patterns that express the data's most important characteristics, and that can be used to make *predictions* about future data, or to help us *understand* the process that generated the data better.

To approach such a task, we usually need to draw a line between aspects of reality that we will take into consideration and aspects we will ignore. Once this is done, we can make a *model*: a simplified description of the part of reality we are interested in. For example, a model used to predict how well a patient will respond to a certain medicine might use that person's age, gender, and clinical measurements such as blood pressure. Then based on previous observations of other patients and how well they responded to the medicine (the *data*), the model can be used to predict this for a new, previously unobserved patient. To predict how well the same person will like a movie (such as done by Netflix), we would likely employ a very different model. This model may also use the person's age and gender. But for such a model, information about other movies that person likes will be very relevant, while white blood cell count will not be so useful.

Ideally, the model includes all aspects of reality we believe might be relevant for the question we want to answer. In practice, this is not always possible. Further simplification may be necessary if the phenomenon under study is too complex to capture entirely, or if with a bigger model, it would take far too long to compute the answer to our question of interest. This dissertation is about understanding what it means for our learning effort if some relevant aspects of reality are not included in the model, or are included in an overly simplified manner.

We already mentioned that one task for which a model might be used is prediction. In this dissertation, we will measure the success at our learning task by our ability to predict new data coming from the same source. Let us look at two more examples illustrating how the ability to predict can be a good measure of learning:

As a first example, one way to understand how a program such as `zip` can compress files is through prediction. While reading through the file to be compressed, before seeing each character in the file, the program decides for each possibility how it will be encoded (with what sequence of bits in the compressed file). It makes this decision using a prediction of what characters are more likely to occur next: these will be encoded using shorter sequences of bits. If the character that actually appears was likely according to the compressor's prediction, this saves a few bits in the compressed file (Cover and Thomas, 1991).

As a second, much more general example, the predictions of a good scientific theory will correspond to observed data. If new data are observed that differ significantly from the theory's predictions, this is a motivation to find improvements to the theory, or even altogether new theories, that predict the data better. For example, Newton's laws of gravitation can be used to predict the motion of the planets around the sun accurately. But for the planet Mercury, which is closest to the sun, the predictions were very slightly off. In 1915, Einstein's general theory of relativity provided new predictions, which did match the observations of Mercury's orbit; this theory has continued to predict the effects of gravity in the century since then (Misner et al., 1973).

The models we consider in this dissertation are *statistical* models. Such models give a description of part of reality in the language of mathematics, so that the predictions we are interested in can be found by means of computation. A statistical model consists of several possible explanations, or *hypotheses*. Each hypothesis is a *probability distribution* that describes one way in which the data-generating process might work. (One reason for using probabilistic hypotheses is to account for measurement errors in the data.) A model is thus a set of probability distributions.

In addition to the model, we need a *method* that describes how the data and the model should be used: what computations should be performed to find our prediction?

Whenever a model does not incorporate all aspects of reality that are relevant to our learning task, the model is wrong, underspecified, or both. By *wrong*,[1] we mean that the model does not contain a hypothesis which exactly describes the true data-generating process (though it may contain hypotheses that are, in some sense, good approximations to the truth). For example, a medical model like the one we mentioned earlier may predict that a specific medicine is more likely to be effective for patients with higher blood pressures, when in fact this is true only up to a point, above which the medicine becomes less effective again. By *underspecified*, we mean that the hypotheses in the model describe the full process of data generation for only part of the data. For example, the medical model predicts how well a medicine will work *given* the characteristics of a new patient, but that same model might not predict the characteristics of this patient.

In this dissertation, we discuss three scenarios involving models that are wrong or underspecified. In each case, we find that standard methods for

[1] This is also called *misspecified*.

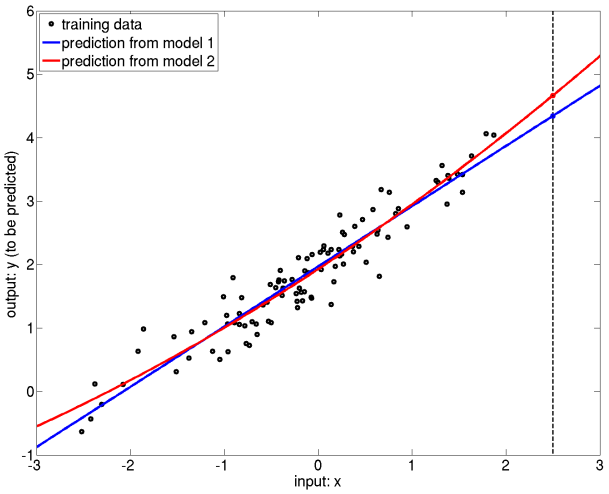


Figure 1.1: A simple example of a regression problem

learning from data and for predicting new data may fail, sometimes dramatically, and we present different methods that continue to perform well at their intended tasks even if the models are wrong or underspecified.

We now discuss these three scenarios in turn. The first two of these scenarios both involve regression problems, explained in Section 1.1. The third scenario, introduced in Section 1.2, addresses a very different situation where we must predict a variable of interest that we do not observe directly, but only through the output of some unknown process.

1.1 Regression

A *regression* problem is a kind of learning problem in which the data consist of two parts: the *input* X and the *output* Y (Hastie et al., 2001). The task at hand is to predict the unknown value of Y given a known value of X. The two examples from the beginning were regression problems: in the medical example, X describes characteristics of a patient and Y is the effectiveness of the medicine for that patient; in the Netflix example, X consists of different characteristics of a user, and Y is the rating that user would assign to some movie.

Figure 1.1 shows a much simpler but illustrative example where X and Y are single numbers. We have observed 100 data points: the *training* data. For learning from these data, two different models have been tried. Model 1, the simpler of the two, considers all linear functions as possible explanations (hypotheses) of how Y is generated given X. The straight blue line is a hypothesis from the model that might be used to predict Y for new data points (the

test data).

Model 2 is more complex than model 1: it includes all hypotheses from model 1, but also curved lines (quadratic functions). If we use this model to predict new values of Y, we might use the red line in the figure.

In the regression problems we consider, the output variable is a single number, and the input variable may be either a single number or a vector of numbers.

We will consider *linear models*. Each hypothesis in a linear model can be described by a vector of parameters β. The hypothesis corresponding to β then states that for a data point with input variable X consisting of the p numbers $X_1, X_2, \ldots, X_p$, the output variable Y is given by

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon.$$

Here ϵ is a random *noise* term, and the other terms on the right hand side define the *regression curve*.

This type of model is much less restrictive than it may at first appear, because we still have a lot of freedom in choosing the values we put in the input vector X. For example, if we believe some real-world quantity S should be taken into account by the model but do not believe that S and Y are linearly related, we may let $X_i = S^i$ for $i \in 1, \ldots, p$: then the model contains a hypothesis for each polynomial up to degree p. Model 2 from the figure is an example of this, so it is also a linear model.

After observing a number of data points, we will see that some hypotheses fit the data better than others. Thus we can *learn* the parameter vector β from the data. One method that can be used for this task is to find the *maximum likelihood* parameter: the value of β for which the probability of observing the data would be largest. This is the main approach in the first of our two scenarios (and the lines shown in Figure 1.1 were found this way). In the second scenario another approach is used, which will be explained in Section 1.1.2.

To fully specify a probability distribution of the output variable given the input variable, the model must also describe the distribution from which the noise term ϵ is drawn. We may for example assume these follow a Gaussian (also called normal) distribution. For this distribution, we still need to specify a variance. Maybe we have a good idea of what this variance is; then we can just let all hypotheses in the model use this fixed value. Alternatively, our model may include a parameter for the variance in addition to the parameter vector β that describes the regression curve; then we can learn the variance from the data using our model.

Besides learning the parameters within a model, we may also face the problem of *model selection*. It is often impossible in practice to formulate and use a model that incorporates all aspects of reality we think might be relevant, so that any model we end up using must be a compromise. It may not be clear in advance how to make this compromise: a model that is too 'simple' may not contain any hypotheses that even resemble the data-generating distribution, while a model that is too 'complex' may contain so many hypotheses that it may be nearly impossible to learn which ones are good. So we may want

to consider a set of candidate models, and learn from the training data which one to use to make our predictions. (Alternatively, we may want to combine the predictions from all these models into a single prediction, giving more weight to some models and less weight to others.) Even with the help of the training data, choosing a model can be a difficult task, as it is in Figure 1.1. The red line was chosen from a larger set of options, and so could be chosen to fit the data we have seen slightly better than the blue line. But how can we tell if model 2 will also be better for data points we have not seen yet?

Again, many different statistical methods exist that answer these questions in as many different ways. In the two regression scenarios, two different methods take the spotlight: AIC in Chapter 2, and Bayesian model averaging in Chapters 3 to 5. We introduce these scenarios in the next two sections.

1.1.1 Extra-sample prediction

You are given 100 data points (X, Y) as shown in Figure 1.1, and are asked to predict y' for a new point with $x' = 2.5$ (marked by the dashed vertical line). Do you base your prediction on the linear or the quadratic model? This is the main question of Chapter 2.

The regression models we consider may be regarded as underspecified, because they only describe how the output is generated given the input, but not how the input is generated. This is enough to have each model give predictions of Y given X (the behaviour of the inputs is not relevant for this task), so this is not a failing of the models. But to assess which model might give the best predictions on future 'test' data, we would need some information (for example in the form of predictions from an auxiliary model, independent of the other models) on the distribution of inputs (for this task, that information *is* relevant). Standard methods for choosing a model or averaging over models do not consider such information.

In Chapter 2, we take the classical method AIC (*Akaike's Information Criterion*;[2] Akaike (1973)), and adapt it to take information on the inputs into consideration. We call our modification XAIC.

When comparing AIC and XAIC, we see that AIC is based on an estimate of the prediction error that would be obtained if the test inputs were randomly chosen from the inputs already seen in the training sample (the *in-sample error*). XAIC does not assume that the test inputs will be so alike the training inputs, and computes the *extra-sample error*. (XAIC stands for *eXtra-sample AIC*.) If the training and test inputs are drawn from the same probability distribution, the difference between the two error measures becomes smaller and smaller as the number of training data points grows. So in this case, AIC eventually performs well. However, the difference between AIC and XAIC never vanishes entirely, even in this case; in a different case where training and test inputs do not come from the same distribution, it becomes very important to estimate the extra-sample and not the in-sample error, and XAIC is strongly recommended over AIC.

[2] According to Akaike himself, AIC stands for 'An Information Criterion'.

XAIC requires some information about test inputs. Predicting these with an auxiliary model may be hard (and is not the problem we are really interested in, namely predicting outputs given inputs). So one special case of XAIC is of special interest: FAIC (*Focused AIC*), which does not require predictions of X, but tailors its model selection choice to each possible value of the test input: for the same training data but different test inputs $X = x$, a different model may be selected. This makes FAIC radically different from most model selection methods,[3] which usually choose a model (or a weighting over models) without any knowledge of the task for which this model will then be used. If instead we take the prediction task as fundamental and view model selection as part of this task, then focused model selection is a natural approach.

For the data shown in Figure 1.1, AIC tells us to use the more complex model 2 for all our predictions, even though the data were actually generated according to the simpler model 1. FAIC will also use model 2 for test inputs near 0 (where the red and blue regression curves are very similar), but will use model 1 for test inputs farther away from 0, like $x' = 2.5$.

The method AIC has existed for over forty years, and in that time, many variations and improvements have been proposed: AIC_C (Hurvich and Tsai, 1989), BPIC (Ando, 2007), DIC (Spiegelhalter et al., 2002), GIC (Konishi and Kitagawa, 1996), NIC (Murata et al., 1994), TIC (see Burnham and Anderson, 2002), WAIC (Watanabe, 2010), ... Thus a reader might think that XAIC is just another slight variation. However, none of these criteria address the issue that XAIC and FAIC address. Thus XAIC is certainly not 'yet another information criterion'.[4]

Some more details on our approach are given in Section 1.1.3. In Chapter 2, we compare XAIC and FAIC to other methods, both theoretically and experimentally.

1.1.2 Bayesian inconsistency

The *Bayesian* approach to dealing with uncertainty is to represent it in terms of a probability distribution. For example, if we think one of the models in a regression problem is correct but we do not know which one, we may make our uncertainty precise by specifying a *prior* distribution over the models. This prior may put the same amount of probability mass on each of the models to represent that we think they are all equally likely to be the correct model, or it may put more mass on some than on others.

After seeing data, we can use *Bayes' theorem* to compute a *posterior* distribution over the models. This distribution tells us exactly what would be rational to believe after seeing the data, if our initial beliefs corresponded to the prior distribution (Bernardo and Smith, 1994).

[3] One exception is the Focused Information Criterion (FIC) of Claeskens and Hjort (2003), which shares the idea of 'focus' on a prediction task of interest, and hence gave FAIC its name; yet it works out this idea in a completely different manner.

[4] That would be YAIC.

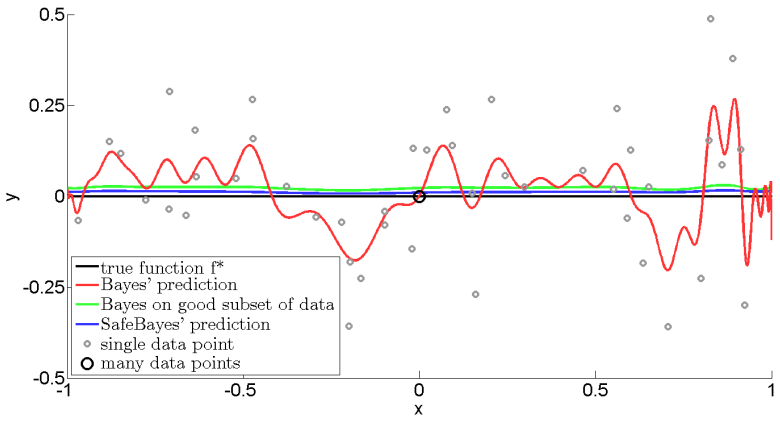


Figure 1.2: Bayesian inconsistency in regression: When the training data contain many 'easy' points (at $(0,0)$), the Bayesian posterior assigns most of its mass to the most complex models available, even though a much simpler model is actually much closer (under a variety of distance measures) to being correct.

Instead of using AIC or XAIC as in Chapter 2, we may use the posterior distribution over the models to select a model for giving predictions, or a weighting over the models. The latter approach is a natural way of taking into account all the uncertainty over the choice of model represented by the posterior. This approach is called *Bayesian model averaging (BMA)*. This is the method for dealing with multiple models we investigate in Chapters 3, 4 and 5.

Similarly to putting a prior on the models, we may put a prior on the hypotheses within each model. Then the posterior over the hypotheses can be used to form the *Bayesian predictive distribution*, which can be used in place of the prediction suggested by the maximum likelihood parameter. The predictive distributions of all our models can be combined into a single overall predictive distribution by BMA.

Instead of considering multiple models, we can also use a single complex model, and use the prior on that model's hypotheses to express that we find 'simpler' a hypothesis (that is, with parameter value β closer to 0) more likely. This *Bayesian ridge regression* setting is also investigated in Chapters 3 to 5.

We conduct two kinds of experiments. In one kind, one of the models is correct: the data are generated by a distribution in one of the models (in Figure 1.2 it is in fact generated by $Y = 0 +$ Gaussian noise, which corresponds to the simplest model under consideration). As predicted by the theory, in these experiments, Bayesian inference methods perform very well. This can be seen in Figure 1.2, where the green line is always close to $Y = 0$.

In the other kind of experiment, half the data are generated by the same model as before, but the other half are all in the same place: $(0,0)$. With the

inclusion of these data points, all models are wrong, because the models predict that all data points will have the same noise level regardless of the input. However, the point $(0, 0)$ is on the same regression curve as the other data (the black line at $Y = 0$ in Figure 1.2) and has no noise, so it would seem that these data points should only make the problem easier. It turns out this is not the case for Bayesian model averaging: it severely *overfits*, which essentially means that it gives far too much attention to other, complex models, as the red line in Figure 1.2 shows.

It is known from other work that Bayesian inference can be made robust against this kind of misbehaviour by selecting a small enough *learning rate* (Walker and Hjort, 2002; Zhang, 2006a; Grünwald, 2012), and using the *generalized Bayesian posterior* with this learning rate (instead of the standard Bayesian posterior, which corresponds to a learning rate of 1). With a properly chosen learning rate, Bayes learns to predict as well as the best[5] hypothesis in the model (if appropriate conditions hold). By comparison, under weak conditions, standard Bayes learns to predict as well as the *true* distribution if it is in the model, but cannot be guaranteed to find a good hypothesis in general if the model is wrong.

To put this theory into practice, we need methods that can determine a good value for the learning rate, tuned for the data and the model. In Chapters 3 to 5, we consider several related methods for this task and evaluate them experimentally in regression problems. We use the name *SafeBayes* for these methods. The blue line in Figure 1.2 shows that SafeBayes predicts very well (close to the true $Y = 0$) in the wrong-model experiment.

1.1.3 Details on XAIC and SafeBayes

In a regression problem, there are infinitely many possible outcomes for Y. No matter how good a prediction method is, it cannot be expected to predict the true outcome exactly. So we cannot measure the quality of predictions by simply counting the number of mistakes. Instead, we need other *loss functions* to compare different predictions.

We distinguish two types of predictions. A *point prediction* simply states a value for the outcome. The closer the predicted value $\hat{y}$ is to the actual value y, the better the prediction is, and the smaller the loss. A standard loss function is *squared error loss*, which assigns a loss of $(y - \hat{y})^2$ to this prediction.

The alternative to a point prediction is a *probabilistic prediction*, which takes the form of a probability distribution over all possible outcomes. Such predictions allow the predictor to express his degree of uncertainty over different possible outcomes. For example, a Gaussian prediction with a small variance represents that the actual outcome is expected to be very close to the mean of the distribution, while a large variance means that the predictor would not be surprised if the outcome turns out to be far away.

A loss function that is commonly used to evaluate probabilistic predictions is *logarithmic loss* (Bernardo and Smith, 1994). If the prediction is given by a

[5] In the sense of KL divergence (see Section 1.1.3).

density function $\hat{f}$ and y is the actual outcome, this loss function assigns a loss of $-\log \hat{f}(y)$ to the prediction. Thus the larger the probability density that the predictor gave to the actual outcome, the smaller the loss. (Loss functions for probabilistic predictions are often called *scoring rules*, but we will usually just call them loss functions.)

For Gaussian predictions with mean $\hat{y}$ and fixed variance, logarithmic loss behaves very much like squared error loss: logarithmic loss is then also a quadratic function of y and $\hat{y}$, minimized when $y = \hat{y}$, so that a prediction that minimizes (an expectation over) one of these two loss functions also minimizes (the same expectation over) the other. This holds for arbitrary expectations: the true density does not need to be a Gaussian. If we consider Gaussian probabilistic predictions with different variances, or even altogether different distributions, the two loss functions become less closely related.

Logarithmic loss arises as a natural choice of loss function for comparing probabilistic predictions in several settings. For example, it is a fundamental quantity in data compression (Cover and Thomas, 1991). It is also related to the information-theoretic concept of *Kullback-Leibler (KL) divergence*. The KL divergence between true density f^* and predicted density $\hat{f}$ is

$$\begin{aligned} D(f^* \| \hat{f}) &:= \mathbf{E}_{Y \sim f^*}[\log f^*(Y) - \log \hat{f}(Y)] \\ &= \mathbf{E}_{Y \sim f^*}[-\log \hat{f}(Y)] - \mathbf{E}_{Y \sim f^*}[-\log f^*(Y)]. \end{aligned} \tag{1.1}$$

This can be thought of as a distance measure between distributions. We see that the first term in (1.1) is the expected logarithmic loss of $\hat{f}$, and the second term does not depend on $\hat{f}$. Thus the smaller the expected logarithmic loss of $\hat{f}$, the closer it is to the 'truth' f^* in terms of KL divergence.

Logarithmic loss and KL divergence play a central role in both of our regression scenarios. We discuss this role in more detail now.

AIC and XAIC Given a set of models, each a parameterized set of densities $\mathcal{M}_i = \{f_\theta \mid \theta \in \Theta_i\}$, AIC (Akaike, 1973) tells us to compute

$$-2 \log f_{\hat{\theta}(X,Y)}(Y \mid X) + 2k \tag{1.2}$$

for each model, and select the model that minimizes this value. Here $\hat{\theta}$ denotes the maximum likelihood estimator and k is the number of free parameters in the model. Under some regularity conditions, the quantity (1.2) is an asymptotically unbiased estimator of the KL divergence from the maximum likelihood estimate to the truth, up to a constant (the second term in (1.1)) that is the same for all models.

XAIC replaces the estimator (1.2) by an estimator of the KL divergence at a given set of (or distribution over) test inputs:

$$-2 \log f_{\hat{\theta}(X,Y)}(Y \mid X) + k + \kappa_{X'}, \tag{1.3}$$

where $\kappa_{X'}$ is a quantity that depends on the model, the training data, and the test input(s) X' or some prediction thereof, arrived at by other means. We refer

to Theorem 2.1 for the definition of $\kappa_{X'}$. In the special case where the set of test inputs is exactly identical to the set of training inputs, $\kappa_{X'} = k$ and XAIC reduces to AIC; if X' contains only one point, we find the special case FAIC.

Bayes and SafeBayes For a prior on Θ with density π relative to measure ρ, the standard Bayesian posterior has density

$$\pi(\theta \mid x^n, y^n) = \frac{f_\theta(y^n \mid x^n)\pi(\theta)}{\int f_\theta(y^n \mid x^n)\pi(\theta)\rho(d\theta)}. \tag{1.4}$$

(Here Θ may be the parameter space of a single model; or, regarding the model index as another parameter, Θ may represent all models combined.) If the posterior converges to a hypothesis in the model, then, under weak conditions, it is to a hypothesis that minimizes the KL divergence to the true distribution. In this case, the Bayesian predictive distribution in our regression models will become more and more similar to a Gaussian distribution with the same variance as the noise, and as a result, logarithmic loss and squared error loss behave similarly.

In the wrong-model experiments of Chapter 3 (of which we saw one example in Figure 1.2), however, the posterior does not concentrate, and Bayes' predictions do not start to resemble Gaussian distributions with a fixed variance. *While these predictions still perform well when evaluated by logarithmic loss, they prove inadequate for other tasks, such as point prediction under squared error loss.* (See Section 3.2 and the rest of Chapters 3 to 5 for details.)

The generalized Bayesian posterior with learning rate η has density

$$\pi(\theta \mid x^n, y^n, \eta) = \frac{(f_\theta(y^n \mid x^n))^\eta \pi(\theta)}{\int (f_\theta(y^n \mid x^n))^\eta \pi(\theta)\rho(d\theta)}. \tag{1.5}$$

SafeBayes considers different values of η and chooses one for which Bayes still performs well when constrained to a probabilistic prediction corresponding to a hypothesis in the model (evaluated using logarithmic loss). Two of the versions of SafeBayes discussed in Chapter 3 may also be interpreted as choosing η by evaluating Bayes' point predictions (using squared error loss).

While AIC and Bayes can both be understood as optimizing KL divergence, they do so in very different ways. AIC usually converges to the KL-optimal predictions more quickly than Bayes. On the other hand, if one of the models contains the true data-generating distribution, AIC may continue to try more complex models instead, regardless of the amount of training data (Yang, 2007a).

In their respective chapters, XAIC and SafeBayes are considered as solutions to two different problems appearing in regression. XAIC (or, more specifically, FAIC) addresses the problem that the accuracy of predictions of a linear regression model may vary with the test input, and introduces the quantity κ_x to measure this effect. SafeBayes addresses the problem of potential Bayesian inconsistency on wrong models; one of several explanations of how this inconsistency may occur in regression involves the way in which the variance of the Bayesian predictive distribution depends on x (see Figure 4.4).

In Section 2.6.1, we see that the variance of the Bayesian predictive distribution in a linear regression model is linearly related to κ_x. Looking further, in Section 3.4.2 we find that κ_x also appears in the versions of SafeBayes that randomize over the posterior (κ_x occurs as the last term in (3.23), and the second to last in (3.24)), but not in the versions that pick a single parameter value in the model. This is remarkable, but it is not clear what conclusions can be drawn from this fact: there are significant differences between the ways in which FAIC and SafeBayes use κ_x. Also, our experiments in Chapters 3 to 5 do not conclusively answer the question which version of SafeBayes is better: one that includes κ_x, or one that excludes it.

So, although the methods we develop bear similarities, we cannot draw hard conclusions from that, and we will consider these methods separately in the chapters to come.

1.2 Probability updating with underspecified distributions

And now for something completely different:

1.2.1 The Monty Hall problem

Monty Hall was the host of the TV show *Let's Make a Deal* (Selvin (1975); see also vos Savant (1990)). He used to play games like the following with the contestants.

The contestant faces three closed doors. Behind one of the doors, a great prize is hidden (say, a car), while the other doors hide less appealing objects (say, goats). After the contestant has picked one of the doors, Monty Hall does not simply tell the contestant whether he won the prize or not. Instead, he opens one of the other two doors, revealing a goat behind it. (He knows what is behind each door, so he never accidentally reveals where the car is.) Now he asks the contestant if he would like to change his mind and switch doors. What should the contestant do?

Most people's first impression is that switching does not change the chances of winning the car. Both doors were equally likely to hide the car before Monty Hall opened a door, so why would one be more likely than the other now?

However, this is not the right answer. The location of the car does not change when Monty Hall opens a door, and so the probability[6] that it is behind the initially chosen door remains $1/3$, so the probability for the other closed door must be $2/3$, and it is wise to switch.

The situation becomes easier to understand intuitively when we consider a game with 100 doors. After the contestant's initial pick, Monty Hall opens 98 doors, revealing 98 goats. Two doors remain shut: the door the contestant

[6] To be precise: this is the *unconditional* probability, which does not take into account *which* door the quizmaster opened.

picked, and another door Monty Hall chose to leave shut. However, the contestant picked his door without knowing where the prize is, while Monty Hall does know where the prize is. Clearly, the two doors are not symmetric, and it would be wise to switch. (This is the explanation given by vos Savant in the article (1990) on the Monty Hall game that made this problem famous.)

1.2.2 Generalizing the problem

In the Monty Hall game, the contestant has an initial probability distribution of where the car is hidden. After seeing a goat behind one of the doors, he may (if he is a probabilist or statistician) want to *update* his probabilities. The standard way to update probabilities after receiving such information is by *conditioning* on the set of remaining options. This would tell us that if the car was equally likely to be behind the two remaining doors initially, then it is still equally likely to be behind those doors now.[7]

As the Monty Hall problem shows, this is not always the right answer. The fact that this does not always work correctly motivates us to study a more general question: how should we update our probabilities in similar situations?

The problems we study correspond to the second half of the Monty Hall game: the car has already been hidden (at random) behind one of the three doors, and the contestant has already made an initial guess. How does the quizmaster now decide which door to open, and how can the contestant interpret this new information? We generalize this problem in the following ways: we allow the set of possible *outcomes* to contain any number of values (instead of just three doors as possible locations of the car in the Monty Hall game); we allow any number of *messages* that may be received by the contestant (corresponding to the two doors the quizmaster may choose to open); and the initial distribution on the outcomes may be any probability distribution.

We are thus looking at a generalization of conditioning where a set y is revealed containing true outcome x. In standard situations, the different sets y that may be revealed are disjoint. (For example, in regression we condition on the precise value of the input variable; these different values do not 'overlap'). But here, the sets y *can* overlap: calling the doors $\{a, b, c\}$, assuming the contestant's initial pick is a, Monty Hall reveals either $\{a, b\}$ or $\{a, c\}$.

We can also cast the contestant's problem into the form of the regression problem / statistical models. Some process generates the outcome X, and another (the quizmaster) generates the message Y given X; this is very similar to the regression setting. Now the contestant observes Y (instead of X), and has to predict X (instead of Y). Another important difference with the regression setting is that in the probability updating problem, there is no previous data for the contestant to learn from.

[7] This is the answer we get by 'naive conditioning' (Grünwald and Halpern, 2003). As we will see below (and in detail in Chapter 6), if we had access to additional information and would formalize the problem in a larger space in which this information can be represented, then conditioning *would* give the right answer. The problem is thus not with conditioning per se, but with the choice of space: in the naive space, conditioning is not right, while in the right space, we do not have sufficient information to condition.

We assume the contestant knows the process generating X. However, the process generating Y given X is unknown to him. The only thing he knows about this process is that some pairs of outcomes and messages cannot occur together: the quizmaster could not reveal a goat behind the door that has the car behind it. The set of all distributions satisfying these constraints form a model. However, with no previous data, we have no way to learn which of these distributions to prefer, as we could in the regression setting. The model only tells us what is possible and what is not. Thus we may say the model is underspecified.

1.2.3 Our approach

We do not know how the quizmaster in the Monty Hall game chooses which door to open; or, in the general setting, how the message Y is generated. We need to know this if we want to know the conditional probability that the contestant should assign to each outcome given some message. To do this, we take a *worst-case* approach: we assume that the contestant and the quizmaster are two players, playing the game against each other. (In the Monty Hall problem, this may very well be what is actually going on.) The contestant wants to predict as well as possible, while the quizmaster's goal is to make the contestant's task as hard as he can.

This worst-case approach is standard in game theory, and has been previously applied to the Monty Hall problem by Gill (2011); Gnedin (2011). But the applicability of this theory is not restricted to games between two opponents. If we do not believe the data-generating process is fully adversarial, but merely want to be careful about the conclusions we draw, a worst-case approach can ensure that our predictions will not be terrible, no matter what happens.

In order to make this approach precise, we need to assign numeric scores to the contestant's predictions. For this purpose, we again use *loss functions*, which were introduced in the beginning of Section 1.1.3. In the probability updating problem, a loss function takes a prediction and the actual outcome, based on which it assigns a loss to the contestant. A smaller loss means the contestant predicted well; a higher loss makes the quizmaster happy. In the terminology of game theory, our game is *zero-sum*: the amount that one player wins always equals the amount that the other player loses.

There are many loss functions that we could choose to use, and in general the players may want to play this game differently for different loss functions. For example, if we use *0-1 loss* as our loss function, then the contestant must pick a single outcome; if this was the actual outcome, his loss is 0, otherwise his loss is 1. (This is essentially the original Monty Hall problem. We did not consider this loss function in the regression setting, because there it would almost surely give loss 1 to every prediction.)

We can also allow the contestant to give a probabilistic prediction, allowing him to express his uncertainty over the true outcome. Then we may again use logarithmic loss to judge these predictions. In terms of Kelly gambling (Cover and Thomas, 1991), this can be thought of as the contestant betting money

on each of the possible outcomes, and winning an amount of money depending on the amount he bet on the true outcome. Many other loss functions are possible. We explore their implications for the probability updating game in Chapter 6.

We wish to find a 'worst-case optimal' strategy in such a game. But what do such strategies look like? To answer this, consider a different game: rock-paper-scissors. If you know your opponent is going to play rock next, you play paper and you win. We say that always playing rock is not a worst-case optimal strategy, because if player A is using this strategy and player B figures it out, player B will have a huge advantage. Now consider a different strategy: your opponent plays rock, paper or scissors each with probability $1/3$. Then even if he tells you what his strategy is, this would not give you any advantage. A strategy like this will be called *worst-case optimal*: it gives a player the best possible result against an opponent with the exact opposite goal, even if the opponent figures out what strategy the player is using and then picks his own strategy to exploit this information as much as possible (the worst case).

Similarly, in the Monty Hall game, if the car happens to be behind the door the contestant guessed, then the quizmaster will have a choice of which door to open. It is a worst-case optimal strategy for the quizmaster to make this decision at random, choosing either door with probability $1/2$. If he uses another strategy, such as always opening the leftmost door out of his two options, then this might give an advantage to a contestant who figures this out: if the quizmaster ever opens the *right*most door, then the contestant can bet all his money that the car is behind the leftmost door; depending on the loss function, this may improve (decrease) his loss on average. (In the Monty Hall game with 0-1 loss, it does not make a difference if the contestant tells the quizmaster beforehand that he will always switch doors. However, it does make a difference in other instances of the generalized game we consider. So in general, we want both players' strategies to be worst-case optimal.)

For many loss functions, worst-case optimal strategies for both players have the property that neither the quizmaster nor the contestant can benefit from knowing the other's strategy. Such a pair of strategies is called a *Nash equilibrium* (Nash, 1951).

Worst-case optimal strategies may be difficult to compute. In particular, it is not always a worst-case optimal strategy for the quizmaster to pick a message uniformly at random, as is sometimes assumed in solutions to the Monty Hall game (when the marginal is also uniform). For example, in the Monty Hall game with a nonuniform marginal (that is, some doors have a larger probability of containing the prize), the worst-case optimal strategy for the quizmaster may require him to open one of the doors with a higher probability than the other. How hard it is to find a worst-case optimal strategy in a probability updating game depends in large part on the arrangement of the possible messages, as we will see in Chapter 7. For some message structures, efficient methods exist that allow a worst-case optimal strategy to be computed quickly. This is the topic of Chapter 8.

1.3 Overview of this dissertation

In Chapter 2, we investigate the problem of model selection for extra-sample prediction. Based on a novel, unbiased expression for KL divergence, we propose XAIC and its special case FAIC as versions of AIC intended for this task, and show that both may significantly improve predictive performance compared to standard methods, including AIC and Bayesian model averaging.

Chapters 3 to 5 concern inconsistency of Bayesian inference for wrong models. Experiments with linear models exhibiting such inconsistency are shown in Chapter 3, both in a model averaging/selection and in a Bayesian ridge regression setting. To remedy the problem, we equip the likelihood in Bayes' theorem with an exponent called the learning rate, and we propose the *SafeBayesian* method to learn the learning rate from the data. SafeBayes tends to select small learning rates as soon the standard posterior is not 'cumulatively concentrated', and its results on our data are quite encouraging.

In Chapter 4, we give several explanations of how this inconsistency may occur under 'bad' misspecification, and why SafeBayes provides a solution to this problem. We also discuss how our inconsistency example and the SafeBayes method relate to other work.

Chapter 5 provides additional regression experiments to test whether the results of Chapter 3 also hold with different priors, models, methods, and data-generating distributions. We find that three versions of SafeBayes consistently perform well, while other methods, including Bayes and AIC, perform badly.

The final three chapters of this dissertation deal with worst-case optimal probability updating, an alternative to conditioning that may be used when the distribution is not fully specified. In Chapter 6, we introduce the problem, and find how optimal solutions may be recognized for different loss functions; our main tool is convex analysis. We find that for logarithmic loss, optimality is characterized by the elegant *RCAR (reverse coarsening at random)* condition.

In Chapter 7, we analyse the combinatorial aspect of the probability updating problem, and present some theoretical tools that may help us compute worst-case optimal solutions to a probability updating problem, as opposed to merely recognizing such solutions. Further, we see that the applicability of the RCAR condition is not restricted to the cases discovered in Chapter 6, and explore the consequences.

In Chapter 8, we give algorithms that automate the task of finding worst-case optimal solutions, for restricted classes of probability updating problems.

Section 8.3 in Chapter 8 is an intermezzo that investigates a notion of fairness in the theory of maximum flows. While needed to understand the subsequent developments of Chapter 8, it may also be of independent interest.

Chapter 2

Extra-Sample Prediction in Linear Models

Methods for combining predictions from a number of models in a supervised learning setting must somehow estimate/predict the quality of a model's predictions at unknown future inputs. Many of these methods (often implicitly) make the assumption that the test inputs are identical to the training inputs, which is seldom reasonable. By failing to take into account that prediction will generally be harder for test inputs that did not occur in the training set, this can sometimes lead to the selection of too complex models.

Based on a novel, unbiased expression for KL divergence, we propose XAIC and its special case FAIC as versions of AIC intended for prediction that use different degrees of knowledge of the test inputs. Both methods substantially differ from and may outperform all the known versions of AIC *even when the training and test inputs are i.i.d.*, and are especially useful for deterministic inputs and under covariate shift. Our experiments on linear models suggest that if the test and training inputs differ substantially, then XAIC and FAIC predictively outperform AIC, BIC and several other methods including Bayesian model averaging.

Terminology In this chapter we freely use machine learning terminology; let us quickly compare the main notions to their counterpart in statistics. By *supervised* learning problems, we mean problems such as regression and classification in which we observe a *training set* (i.e. a sample) of (X, Y) pairs, where X are *inputs* (covariates) and Y is the *output variable*, and we are interested in learning, based on the sample, a (set of) probability distribution(s) or a (set of) decision rule(s) that can help us to make predictions of Y given *test input* X.

2.1 Introduction

In the statistical problem of model selection, we are given a set of models $\{\mathcal{M}_i \mid i \in \mathcal{I}\}$, each of the form $\mathcal{M}_i = \{g_i(\cdot \mid \theta) \mid \theta \in \Theta_i\}$, where the $g_i(\cdot \mid \theta)$ are

density functions on (sequences of) data. We wish to use one of these models to explain our data and/or to make predictions of future data, but do not know which model explains the data best. It is well known that simply selecting the model containing the maximum likelihood distribution from among all the models leads to overfitting, so any expression of the quality of a model must somehow avoid this problem. One way to do this is by estimating each model's ability to predict *unseen* data (this will be made precise below). This approach is used by many methods for model selection, including cross-validation, AIC (Akaike, 1973) and its many variants, Gelfand and Ghosh's D_k (1998), BPIC (Ando, 2007), and WAIC (Watanabe, 2010). However, none of these methods takes into account that for supervised learning problems, the generalization error being estimated will vary with the test input variables. Instead, they implicitly assume that the test inputs will be *identical* to the training inputs.

In this chapter, we derive an estimate of the generalization error that does take the input data into account, and use this to define a new model selection criterion XAIC, its special case FAIC, and the variants XAIC_C and FAIC_C (small sample corrections). We use similar assumptions as AIC, and thus our methods can be seen as relatives of AIC that are adapted to supervised learning when the training and test inputs differ. Our experiments show that our methods have excellent predictive performance, better even than Bayesian model averaging in some cases. Also, we show theoretically that AIC's unawareness of input variables leads to a bias in the selected model order, even in the seemingly safe case where the test inputs are drawn from the same distribution as the training inputs. No existing model selection method seems to address this issue adequately, making XAIC and FAIC more than "yet another version of AIC".

It is in fact quite surprising that, more than forty years after its original invention, all the forms of AIC currently in use are biased in the above sense, and in theoretical analyses, conditional model selection methods are often even compared on a new point x constrained to be one of the x values in the training data (see e.g. Yang, 2005), even though in most practical problems, a new point x will *not* be drawn from this empirical training data distribution, but rather should be regarded as falling in one of the three cases considered in this chapter: (a) it is drawn from the same distribution as the training data (but not necessarily equal to one of the training inputs); (b) it is drawn from a different distribution (covariate shift); (c) it is set to a fixed, observable value, usually not in the training set, but the process that gave rise to this value may not be known.

2.1.1 Goals of model selection

When choosing among or combining predictions from different models, one can have different goals in mind. Whereas BIC and BMS (Bayesian model selection) focus on finding the most probable model, methods like AIC, cross-validation and SRM (structural risk minimization; Vapnik, 1998) aim to find the model that leads to the best *predictions* of future data. While AIC and cross-

validation typically lead to predictions that converge faster to optimal in the sense of KL divergence than those of BIC and BMS, it is also well-known that, unlike BIC and BMS, such methods are not statistically consistent (i.e. they do not find the smallest submodel containing the truth with probability 1 as $n \to \infty$); there is an inherent conflict between these two goals, see for example Yang (2007a); Van Erven et al. (2007, 2012). Like AIC, the XAIC and FAIC methods developed here aim for predictive optimality rather than consistency, thus, if consistency is the main concern, they should not be used. We also stress at the outset that, unlike most other model selection criteria, the model selected by FAIC may *depend* on the new x whose corresponding y value is to be predicted; for different x, a different model may be selected based on the same training data. Since — as in many other model selection criteria — our goal is predictive accuracy rather than 'finding the true model', and since the dependence on the test x helps us to get substantially better predictions, we are not worried by this dependency.

FAIC thus cannot be said to select a 'single' model for a given training set — it merely outputs a *function* from x values to models. As such, it is more comparable with BMA (Bayesian model *averaging*) rather than BMS (*selection*). BMA is of course a highly popular method for data prediction; like FAIC, it adapts its predictions to the test input x (as we will see, FAIC tends to select a simpler model if there are not many training points near x; BMA predicts with a larger variance if there are not many training points near x). BMA leads to the optimal predictions in the idealized setting where one takes expectation under the prior (i.e., in frequentist terms, we imagine nature to draw a model, and then a distribution within the chosen model, both from the prior used in BMA, and then data from the drawn distribution), and usually performs very well in practice as well. It is of considerable interest then that our XAIC and FAIC outperform Bayes by a fair margin in some of our experiments in Section 2.5.

2.1.2 In-sample and extra-sample error

Many methods for model selection work by computing some estimate of how well each model will do at predicting unseen data. This generalization error may be defined in various ways, and methods can further vary in the assumptions used to find an estimate. AIC (Akaike, 1973) is based on the expression for the generalization error

$$-2\,\mathbf{E}_{\mathbf{U}}\,\mathbf{E}_{\mathbf{V}} \log g_i(\mathbf{V} \mid \hat{\theta}_i(\mathbf{U})), \tag{2.1}$$

for model $\mathcal{M}_i = \{g_i(\cdot \mid \theta) \mid \theta \in \Theta_i\}$, where $\hat{\theta}_i(\mathbf{U})$ denotes the element of Θ_i which maximizes the likelihood of data $\mathbf{U}$, and where both random variables are independent samples of n data points each, both following the true distribution of the data. (In this chapter, we use capitals to denote sequences of data points, and boldface for random variables. Throughout this chapter, log denotes the natural logarithm.) Up to an additive term which is the same for all models, the inner expectation is the KL divergence from the true distribution to $g_i(\cdot \mid \hat{\theta}_i(\mathbf{U}))$. An interpretation of (2.1) is that we first estimate

the model's parameters using a random sample $\mathbf{U}$, then judge the quality of this estimate by looking at its performance on an independent, identically distributed sample $\mathbf{V}$. AIC then works by estimating (2.1) for each model by the asymptotically unbiased estimator

$$-2 \log g_i(\mathbf{U} \mid \hat{\theta}(\mathbf{U})) + 2k \tag{2.2}$$

where k is the number of parameters in the model, and selecting the model minimizing this estimate. Thus AIC selects the model whose maximum likelihood estimate is expected to be closest to the truth in terms of KL divergence. In the sequel, we will consider only one model at a time, and therefore omit the model index.

In supervised learning problems such as regression and classification, the data points consist of two parts $u_i = (x_i, y_i)$, and the models are sets of distributions on the *output variable* $\mathbf{y}$ conditional on the *input variable* x (which may or may not be random). We call these *conditional* models. The conditionality expresses that we are not interested in explaining the behaviour of x, only that of $\mathbf{y}$ given x. Then (2.1) can be adapted in two ways: in the terminology of Hastie et al. (2001), as the *extra-sample error*

$$-2 \, \mathbf{E}_{\mathbf{Y}|X} \, \mathbf{E}_{\mathbf{Y}'|X'} \log g(\mathbf{Y}' \mid X', \hat{\theta}(X, \mathbf{Y})), \tag{2.3}$$

and, replacing both X and X' by a single variable X, as the *in-sample error*

$$-2 \, \mathbf{E}_{\mathbf{Y}|X} \, \mathbf{E}_{\mathbf{Y}'|X} \log g(\mathbf{Y}' \mid X, \hat{\theta}(X, \mathbf{Y})), \tag{2.4}$$

where capital letters again denote sequences of data points. Contrary to (2.1), these quantities capture that the expected quality of a prediction regarding $\mathbf{y}$ may vary with x.

An example of a supervised learning setting is given by *linear models*. In a linear model, an input variable x is represented by a *design vector* and a sequence of n inputs by an $n \times p$ *design matrix*; with slight abuse of notation, we use x and X to represent these. Then the densities $g(\mathbf{Y} \mid X, \beta)$ in the model are Gaussian with mean $X\beta$ and covariance matrix $\sigma^2 I_n$ for some fixed σ^2. Because g is of the form $e^{-\text{squared error}}$, taking the negative logarithm as in (2.1) produces an expression whose main component is a sum of squared errors; the residual sum of squared errors $\mathrm{RSS}(\mathbf{Y})$ is the minimum for given data, which is attained by the maximum likelihood estimator. Alternatively, σ^2 may be another parameter in addition to β if the true variance is unknown.

It is standard to apply ordinary AIC to supervised learning problems. For example, for linear models with fixed variance, (2.2) takes the well-known form

$$\frac{1}{\sigma^2} \mathrm{RSS}(\mathbf{Y}) + 2k. \tag{2.5}$$

But because the standard expression behind AIC (2.1) makes no mention of X or X', this corresponds to the tacit assumption that $X = X'$, so that the in-sample error is being estimated.

However, the extra-sample error is more appropriate as a measure of the expected performance on new data. AIC was intended to correct the bias that results from evaluating an estimator on the data from which it was derived, but because it uses the in-sample error, AIC evaluates estimators on new output data, but old input data. So we see that in supervised problems, a bias similar to the one it was intended to correct is still present in AIC.

2.1.3 Contents

The remainder of this chapter is structured as follows. In Section 2.2, we develop our main results about the extra-sample error and propose a new model selection criterion based on this. It involves $\kappa_{X'}$, a term which can be calculated explicitly for linear models; we concentrate on these models in the remainder of the chapter. Special cases of our criterion, including a focused variant, are presented in Section 2.3. In Section 2.4 we discuss the behaviour of our estimate of the extra-sample error, and find that without our modification, AIC's selected model orders are biased. Several experiments on simulated data are described in Section 2.5. Section 2.6 contains some further theoretical discussion regarding Bayesian prediction and covariate shift. Finally, Section 2.7 concludes. Technical regularity conditions and proofs are in the appendix.

2.2 Estimating the extra-sample error

In this section, we will derive an estimate for the extra-sample error. Our assumptions will be similar to those used in AIC to estimate the in-sample error; therefore, we start with some preliminaries about the setting of AIC.

2.2.1 Preliminaries

In the setting of AIC, the data points are independent but not necessarily identically distributed. The number of data points in $\mathbf{Y}$ and $\mathbf{Y}'$ is n. We define the Fisher information matrix $I(\theta)$ as $-\mathbf{E}_{\mathbf{Y}'} \frac{\partial^2}{\partial \theta^2} \log g(\mathbf{Y}' \mid \theta)$, and define the conditional Fisher information matrix $I(\theta \mid X')$ analogously. We write $\operatorname{Cov}(\hat{\theta}(X, \mathbf{Y}) \mid X)$ for the conditional covariance matrix $\mathbf{E}_{\mathbf{Y}|X}[\hat{\theta}(X, \mathbf{Y}) - \mathbf{E}_{\mathbf{Y}|X} \hat{\theta}(X, \mathbf{Y})][\hat{\theta}(X, \mathbf{Y}) - \mathbf{E}_{\mathbf{Y}|X} \hat{\theta}(X, \mathbf{Y})]^\top$.

Under standard regularity assumptions, there exists a unique parameter value θ_o that minimizes the KL divergence from the true distribution, and this is what $\hat{\theta}(\mathbf{Y})$ converges to. Under this and other (not very restrictive) regularity assumptions (Shibata, 1989), it can be shown that (Burnham and Anderson, 2002)

$$-2 \log g(\mathbf{Y} \mid \hat{\theta}(\mathbf{Y})) + 2\widehat{\operatorname{tr}} \left\{ I(\theta_o) \operatorname{Cov}(\hat{\theta}(\mathbf{Y})) \right\} \tag{2.6}$$

(where $\widehat{\operatorname{tr}}$ represents an appropriate estimator of that trace) is an asymptotically unbiased estimator of (2.1). The model selection criterion TIC (Takeuchi's information criterion) selects the model which minimizes (2.6).

The estimator of the trace term that TIC requires has a large variance, making it somewhat unreliable in practice. AIC uses the very simple estimate $2k$ for TIC's trace term. This estimate is generally biased except when the true data-generating distribution is in the model, but obviously has 0 variance. Also, if some models are more misspecified than others, those models will have a worse log-likelihood. This term in AIC grows linearly in the sample size, so that asymptotically, those models will be disqualified by AIC. Thus AIC selects good models even when its penalty term is biased due to misspecification of the models.

This approach corresponds to making the following assumption in the derivation leading to AIC's penalty term:

Assumption 2.1. *The model contains the true data-generating distribution.*

It follows that θ_o specifies this distribution. We emphasize that this assumption is only required for AIC's derivation and does not mean that AIC necessarily works badly if applied to misspecified models. Under this assumption, the two matrices in (2.6) cancel, so the objective function becomes (2.2), the standard formula for AIC (Burnham and Anderson, 2002).

We now move to supervised learning problems, where the true distribution of the data and the distributions g in the models are conditional distributions of output values given input values. In this setting, the data are essentially i.i.d. in the sense that $g(\mathbf{Y} \mid X, \theta) = \prod_{i=1}^{n} g(\mathbf{y}_i \mid x_i, \theta)$. That is, the outputs are independent given the inputs, and if two input variables are equal, the corresponding output variables are identically distributed. Also, the definition of θ_o would need to be modified to depend on the training inputs, but since Assumption 2.1 now implies that $g(\mathbf{y} \mid x, \theta_o)$ defines the true distribution of $\mathbf{y}$ given x for all x, we can take this as the definition of θ_o for supervised learning when Assumption 2.1 holds.

For supervised learning problems, AIC and TIC silently assume that X' either equals X or will be drawn from its empirical distribution. We want to remove this assumption.

2.2.2 Main results

We will need another assumption:

Assumption 2.2. *For training data $(X, \mathbf{Y})$ and (unobserved) test data $(X', \mathbf{Y}')$,*

$$-\frac{1}{n}\,\mathbf{E}_{\mathbf{Y}|X} \log g(\mathbf{Y} \mid X, \theta_o) \;=\; -\frac{1}{n'}\,\mathbf{E}_{\mathbf{Y}'|X'} \log g(\mathbf{Y}' \mid X', \theta_o),$$

where n and n' denote the number of data points in X and X', respectively.

This assumption ensures that the log-likelihood on the test data can be estimated from the training data. If $\mathbf{X}$ and $\mathbf{X}'$ are random and mutually i.i.d., this is automatically satisfied when the expectations are taken over these inputs as well. While this assumption of randomness is standard in machine

learning, there are other situations where X and X' are not random and Assumption 2.2 holds nevertheless. For instance, this is the case if $g(\mathbf{y} \mid x, \theta)$ is such that $\mathbf{y}_i = f_\theta(x_i) + \mathbf{z}_i$, where the noise terms $\mathbf{z}_i$ are zero-mean and i.i.d. (their distribution may depend on θ). This additive noise assumption is common in regression-like settings. Then Assumption 2.1 implies that Assumption 2.2 holds for all X, X'.

To get an estimator of the extra-sample error (2.3), we do not make any assumptions about the process generating X and X' but leave the variables free. We allow $n \neq n'$.

Theorem 2.1. *Under Assumptions 2.1 and 2.2 and some standard regularity conditions (detailed in Assumption 2.3 in the appendix), and for n' either constant or growing with n,*

$$-2\frac{n}{n'}\,\mathbf{E}_{\mathbf{Y}|X}\,\mathbf{E}_{\mathbf{Y}'|X'}\log g(\mathbf{Y}' \mid X', \hat{\theta}(X, \mathbf{Y})) \\ = -2\,\mathbf{E}_{\mathbf{Y}|X}\log g(\mathbf{Y} \mid X, \hat{\theta}(X, \mathbf{Y})) + k + \kappa_{X'} + o(1), \quad (2.7)$$

where $\kappa_{X'} = \frac{n}{n'}\operatorname{tr}\left\{I(\theta_o \mid X')\operatorname{Cov}(\hat{\theta}(X, \mathbf{Y}) \mid X)\right\}$.

Moreover, if the true conditional distribution of $\mathbf{Y}$ *given* X *is Gaussian with fixed variance and the conditional distributions in the models are also Gaussian with that same variance (as is the case in linear models with known variance), then the above approximation becomes exact.*

We wish to use (2.7) as a basis for model selection. To do this, first note that (2.7) can be estimated from our training data using

$$-2\log g(\mathbf{Y} \mid X, \hat{\theta}(X, \mathbf{Y})) + k + \kappa_{X'}. \quad (2.8)$$

Theorem 2.1 expresses that this is an asymptotically unbiased estimator of the extra-sample error. We see that the difference with standard AIC (2.2) is that the penalty $2k$ has been replaced by $k + \kappa_{X'}$. We propose to use (2.8) as the basis for a new model selection criterion *extra-sample AIC (XAIC)*, which chooses the model that minimizes an estimator of (2.8). What remains for this is to evaluate $\kappa_{X'}$, which may depend on the unknown true distribution, and on the test set through X'. This is done below for the case of linear models; further intuition about $\kappa_{x'}$ for linear models and a single test input x' is given in Section 2.4 (where it is related to a norm of x') and Section 2.6.1 (where it is given a Bayesian interpretation).

2.2.3 The $\kappa_{X'}$ and $o(1)$ terms for linear models

If the densities g are Gaussian, then $\kappa_{X'}$ does not depend on the unknown θ_o because the Fisher information is constant, so no additional estimation is necessary to evaluate it. Thus for a linear model with fixed variance, $\kappa_{X'}$ becomes

$$\kappa_{X'} \;=\; \frac{n}{n'}\operatorname{tr}\left\{\left[\frac{1}{\sigma^2}X'^{\top}X'\right]\left[\sigma^2(X^{\top}X)^{-1}\right]\right\} \;=\; \frac{n}{n'}\operatorname{tr}\left[X'^{\top}X'(X^{\top}X)^{-1}\right].$$

If the variance is also to be estimated, it can be easily seen that $\kappa_{X'}$ will become this value plus one. In that case, the approximation in Theorem 2.1 is not exact (as it is in the known variance case), but the $o(1)$ term can be evaluated explicitly:

Theorem 2.2. *For a linear model with unknown variance,*

$$-2\frac{n}{n'}\,\mathbf{E}_{\mathbf{Y}|X}\,\mathbf{E}_{\mathbf{Y}'|X'}\log g(\mathbf{Y}' \mid X', \hat{\theta}(X, \mathbf{Y})) \\ = -2\,\mathbf{E}_{\mathbf{Y}|X}\log g(\mathbf{Y} \mid X, \hat{\theta}(X, \mathbf{Y})) + k + \kappa_{X'} + \frac{(k+\kappa_{X'})(k+1)}{n-k-1},$$

where $\kappa_{X'}$ can be computed from the data and equals $(n/n')\,\mathrm{tr}(X'^{\top}X'(X^{\top}X)^{-1}) + 1$, and k is the number of parameters including σ^2.

Theorem 2.2 presents an extra-sample analogue of the well-known small sample correction $\mathrm{AIC_C}$ (Hurvich and Tsai, 1989), which is derived similarly and uses a penalty of $2k + 2k(k+1)/(n-k-1)$. We define $\mathrm{XAIC_C}$ accordingly. Though the theorem holds exactly only in the specific case described, we believe that the extra penalty term will lead to better results in much more general settings in practice, as is the case with $\mathrm{AIC_C}$ (Burnham and Anderson, 2002).

2.3 Model selection for extra-sample prediction

In this section, we discuss several concrete model selection methods, all based on the XAIC formula (2.8) and thus correcting AIC's bias.

2.3.1 Nonfocused versions of XAIC

Except in trivial cases, the extra-sample error (2.3) and its estimate (2.8) depend on the test inputs X', so some knowledge of X' is required when choosing a model appropriate for extra-sample prediction. In a semi-supervised learning setting where X' itself is known at the time of model selection, we could evaluate (2.8) directly for each model. However, X' might not yet be known when choosing a model.

If $\mathbf{X}'$ is not known but its distribution is, we can replace $\kappa_{\mathbf{X}'}$ by its expectation; for i.i.d. inputs, computing this reduces to computing $\mathbf{E}_{\mathbf{x}'}\, I(\theta_0 \mid \mathbf{x}')$.

If the distribution of $\mathbf{X}'$ is also unknown, we need to estimate it somehow. If it is believed that $\mathbf{X}$ and $\mathbf{X}'$ follow the same distribution, the empirical distribution of $\mathbf{X}$ could be used as an estimate of the distribution of $\mathbf{X}'$. Then AIC is retrieved as a special case. Section 2.4 will show that this is a bad choice even if $\mathbf{X}$ and $\mathbf{X}'$ follow the same distribution, so a smoothed estimate is recommended instead.

Of course, we are not restricted to the case where $\mathbf{X}$ and $\mathbf{X}'$ follow similar distributions. In the setting of covariate shift (Sugiyama and Kawanabe, 2012), the distributions are different but known (or can be estimated). This variant of

XAIC is directly applicable to that setting, yielding an unbiased analogue of AIC.

2.3.2 Focused model selection

It turns out there is a way to apply (2.8) even when nothing is known about the processes generating X and X'. If our goal is prediction, we can set X' to the single point x' for which we need to predict the corresponding $\mathbf{y}'$. Contrary to standard model selection approaches, we thus use x' already at the stage of model selection, rather than only inside the models. We define the model selection criterion *Focused AIC (FAIC)* as this special case of XAIC, and FAIC_C as its small sample correction.

A focused model selection method implements the intuition that those test points whose input is farther away from the training inputs should be predicted with more caution; that is, with less complex models. As discussed in Section 2.1.1, methods that optimize predictive performance often are not consistent; this hurts in particular for test inputs far away from the training inputs. We expect that extra-sample adaptations of such methods (like XAIC) are also inconsistent, but that using the focused special case helps to guard against this small chance of large loss.

Choosing a model specifically for the task at hand potentially lets us end up with a model that performs this task much better than a model found by a non-focused model selection method. However, there are situations in which focus is not a desirable property: the mapping from input values to predictions given by a focused model selection method will be harder to interpret than that of a non-focused method, as it is a combination of the models under consideration rather than a single one of them. Thus, if the experimenter's goal is interpretation/transparency, a focused model selection method is not recommended; these methods are best applied when the goal is prediction.

Evaluating the x'-dependent model selection criterion separately for each x' leads to a regression curve which in general will not be from any one of the candidate models, but only piecewise so. It will usually have discontinuities where it switches between models. If the models contain only continuous functions and such discontinuities are undesirable, Akaike weights (Akaike, 1979; Burnham and Anderson, 2002) may be used to get a continuous analogue of the FAIC regression curve.

2.4 AIC vs. XAIC (k vs. κ_x) in linear models

Intuitively, the quantity κ_x that appears as a penalty term in the XAIC formula (2.8) expresses a measure of dissimilarity between the test input x and the training inputs X. This measure is determined fully by the models and does not have to be chosen some other way. However, its properties are not readily apparent. Because κ_x can be computed exactly for linear models, we investigate some of its properties in that case.

One useful characterization of κ_x is the following: if we express the design vector x of the test point in a basis that is orthonormal to the empirical measure of the training set X, then $\kappa_x = \|x\|^2$.

For given X, x may exist such that κ_x is either greater or smaller than the number of parameters k. An example of $\kappa_x < k$ occurs for the linear model consisting of all linear functions with known variance (so $k = 2$). Then κ_x will be minimized when x lies at the mean of the input values in the training set, where $\kappa_x = 1$.

We will now consider the case where $\mathbf{X}$ and $\mathbf{x}$ are random and i.i.d. We showed that the XAIC expression (2.8) is an unbiased estimator of the extra-sample error. AIC uses k in place of $\kappa_{\mathbf{x}}$, and the above suggests the possibility that maybe the instances where $\kappa_{\mathbf{x}} > k$ and those where $\kappa_{\mathbf{x}} < k$ cancel each other out, so that AIC would also be approximately unbiased as an estimate of the extra-sample error. However, the following proposition shows that, except in a trivial case, $\kappa_{\mathbf{x}}$ is on average greater than k. This means that in those settings, AIC underestimates the model's extra-sample error.

(We should mention here that if $\mathbf{X}$ and $\mathbf{x}$ are random and mutually i.i.d., then as $n \to \infty$, AIC's bias goes to 0. The bias we show below concerns all finite n; additionally, without focus, an extreme $\mathbf{x}$ could result in a very biased AIC value even for large n.)

Proposition 2.3. *Consider a linear model $\mathcal{M}$ with training inputs $\mathbf{X}$ and test input $\mathbf{x}$ i.i.d. such that $\mathbf{X}^\top\mathbf{X}$ is almost surely invertible. Let $\mathcal{M}'$ be the submodel obtained by removing the final entry from every design vector. Then these models are related by $\mathbf{E}\,\kappa_{\mathbf{x}} \geq \mathbf{E}\,\kappa_{\mathbf{x}'} + 1$, with strict inequality if $\mathbf{x}$ has at least two entries.*

It follows by induction on k that for random input data, AIC is biased as an estimate of the extra-sample error except in a special case with $k = 1$. Also, the bias becomes worse for larger models. This last fact is distressing, as it shows that when AIC assesses a sequence of nested models, the amount by which it overestimates their generalization ability grows with the model order. Thus the biases in the AIC scores lead to a bias in the selected model order, which was not evident from earlier work.

The XAIC formula (2.8) contains two terms that depend on the data: minus two times the log-likelihood, and the penalty term $\kappa_{X'}$. The log-likelihood measures distances between output values and is independent of X', while $\kappa_{X'}$ expresses a property of input values and is largely unaffected by output values; in fact, in linear models its computation does not involve any (estimates based on) output values. Hence the variance of XAIC is no greater than that of AIC when comparing the two on fixed X, X', so that XAIC's reduction in bias does not come at the price of an increase in variance. However, focused model selection demands that X' is *not* held fixed, so that FAIC may have a larger variance than AIC. Similarly, if the distribution of $\mathbf{X}'$ is being estimated as part of applying XAIC, the used estimator's quality will affect the accuracy of the estimated generalization error.

2.5 Experiments

We will now experimentally compare XAIC and FAIC (or more precisely, their small-sample corrected versions XAIC_C and FAIC_C) to several other model selection methods, in univariate and multivariate problems.

2.5.1 Description of experiments

In the univariate experiments, linear models $\mathcal{M}_1, \ldots, \mathcal{M}_7$ with unknown variance were considered. Model $\mathcal{M}_i$ contained polynomials of degree $i-1$ (and so had $i+1$ parameters). The input values x of the training data were drawn from a Gaussian distribution with mean 0 and variance 1, while the output values were generated as $\mathbf{y}_i = f(x_i) + \mathbf{z}$ with $\mathbf{z}_i$ i.i.d. Gaussians with mean 0 and variance 0.1, and f some unknown true function. Given 100 training data points, each of the eight model selection methods under consideration had to select a model. The squared risk $(\hat{y} - f(x))^2$ of the chosen model's prediction $\hat{y}$ was computed for each of a range of values of the test point's x, averaged over 100 draws of the training data. This experiment was performed for two different true functions: $f_1(x) = x + 2$ and $f_2(x) = |x|$.

In the multivariate experiments, each input variable was a vector $(u_1, \ldots, u_6)$, and the models corresponded to all possible subsets of these 6 variables. Each model also included an intercept and a variance parameter. The true function was given by $f(u) = 2 + u_1 + 0.1u_2 + 0.03u_3 + 0.001u_4 + 0.003u_5$, and the additive noise was again Gaussian with variance 0.1. A set of $n' = 400$ test inputs was drawn from a standard Gaussian distribution, but the training inputs were generated differently in each experiment: from the same Gaussian distribution as the test inputs; from a uniform distribution on $[-\sqrt{3}, \sqrt{3}]^6$; or from a uniform 'spike-and-slab' mixture of two Gaussians with covariance matrices $(1/5)I_6$ and $(9/5)I_6$. Note that all three distributions have the same mean and covariance as the test input distribution, making these mild cases of covariate shift. For the Gaussian training case, we report the results for $n = 60$ and, after extending the same training set, for $n = 100$. Squared risks were averaged over the test set and further over 50 repeats of these experiments.

The experiments used the version of XAIC that is given a distribution of the test inputs, but not the test inputs themselves. In the multivariate experiments, XAIC used the actual (Gaussian) distribution of the test inputs. In the univariate case, two instances of XAIC were evaluated: one for test inputs drawn from the same distribution as the training inputs (standard Gaussian), and another (labelled $\text{XAIC}_\text{C}2$) for a Gaussian test input distribution with mean 0 and variance 4.

Bayesian model averaging (BMA) differs from the other methods in that it does not select a single model, but formulates its prediction as a weighted average over them; in our case, its prediction corresponds to the posterior mean over all models. Weighted versions exist of other model selection methods as well, such as Akaike weights (Akaike, 1979; Burnham and Anderson, 2002) for AIC and variants. In our experiments we saw that these usually perform sim-

ilar to but somewhat better than their originals. In our univariate experiments, we decided against reporting these, as they are less standard. However, in the multivariate experiments, the weighted versions were all better than their selection counterparts, so both are reported separately to allow fair comparisons.

In our experiments, BMA used a uniform prior over the models. Within the models, Jeffreys' noninformative prior (for which the selected β would correspond to the maximum likelihood $\hat{\beta}$ used by other methods) was used for the variable selection experiments; for the polynomial case, it proved too numerically unstable for the larger models, so there BMA uses a weakly informative Gaussian prior (variance 10^2 on $\beta_2, \ldots, \beta_7$ with respect to the Hermite polynomial basis, and Jeffreys' prior on σ^2).

Of the model selection methods included in our experiments, AIC was extensively discussed in Section 2.2.1; as with XAIC and FAIC, we use here the small sample correction AIC_C (see Section 2.2.3). BIC (Schwarz, 1978) and BMS were mentioned in Section 2.1.1 as methods that attempt to find the most probable model given the data rather than aiming to optimize predictive performance; both are based on BMA, which computes the Bayesian posterior probability of each model. Three other methods were evaluated in our experiments; these are discussed below.

Like AIC, the much more recent focused information criterion (FIC; Claeskens and Hjort, 2003) is designed to make good predictions. Unlike other methods, these predictions are for a *focus parameter* which may be any function of the model's estimate, not just its prediction at some input value (though we only used the latter in our experiments). Unlike FAIC, it uses this focus not just for estimating a model's variance, but also its bias; FAIC on the other hand uses a global estimate of a model's bias based on Assumption 2.2. A model's bias for the focus parameter is evaluated by comparing its estimate to that of the most complex model available.

Another more recent method for model selection is the subspace information criterion (SIC; Sugiyama and Ogawa, 2001), which is applicable to supervised learning problems when our models are subspaces of some Hilbert space of functions, and our objective is to minimize the squared norm. Like FIC, SIC estimates the models' biases by comparing their estimates to that of a larger model, but it includes a term to correct for this large model's variance. In our experiments, we used the corrected SIC (cSIC) which truncates the bias estimate at 0.

Generalized cross-validation (GCV; Golub et al., 1979) can be seen as a computationally efficient approximation of leave-one-out cross-validation for linear models. We included it in our experiments because Leeb (2008) shows that it performs better than other model selection methods when the test input variables are newly sampled.

2.5.2 Results

Results from the two univariate experiments are shown in Figures 2.1 and 2.2 (squared risks) and in Table 2.1 (selected models). Squared risk results for the

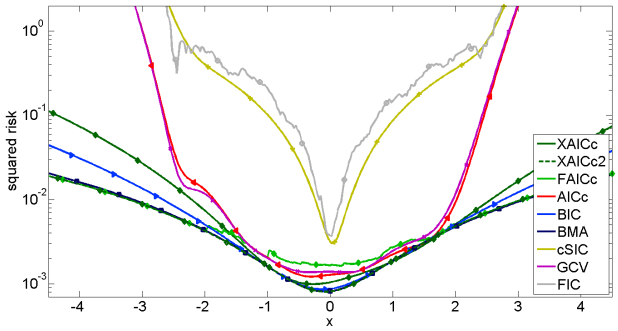


Figure 2.1: Squared risk of different model selection methods as a function of x when the true function is $f_1(x) = x + 2$

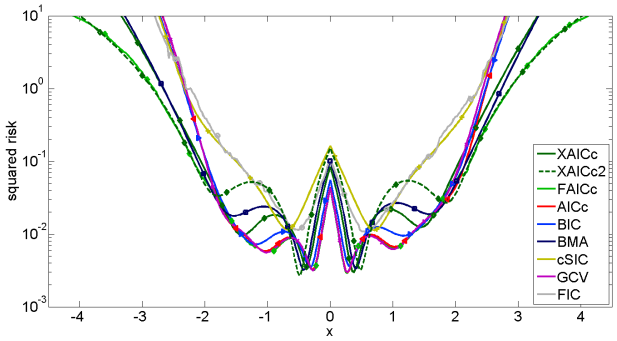


Figure 2.2: Squared risk of different model selection methods as a function of x when the true function is $f_2(x) = |x|$

multivariate experiments are given in Table 2.2 for the model selection methods, and in Table 2.3 for the model weighting/averaging variants.

XAIC and FAIC The characteristic behaviours of our methods are clearly visible in the univariate experiments. Both instances of XAIC perform well overall in both experiment. Of the two, $\text{XAIC}_\text{C}2$ was set up to expect test inputs farther away from the centre. As a result, it selects models more conservatively, and obtains smaller risk at such off-centre test inputs. Its selections were very stable: in both experiments, $\text{XAIC}_\text{C}2$ selected the same model in each of the 100 runs.

We see that in the centre of Figure 2.2, the simple model chosen by $\text{XAIC}_\text{C}2$ was outperformed by more complex models. FAIC exploits this by choosing a model adaptively for each test input. This resulted in good risk performance at all test inputs.

In the multivariate experiments, FAIC was the best method for the spike-

Table 2.1: Average selected model index per method for f_1 and f_2, at test inputs $x' = 0$ and 4 (if different)

	x'	f_1	f_2
XAIC_C		2.10	4.57
$\text{XAIC}_\text{C}2$		2.00	3.00
FAIC_C	0	2.94	6.56
	4	2.00	1.54
AIC_C		2.33	6.38
BIC		2.02	5.70
BMS		2.00	4.05
cSIC		2.94	4.70
GCV		2.38	6.49
FIC	0	2.66	5.29
	4	3.12	5.35

and-slab training data, where there are pronounced differences in training point density surrounding different test points, so that selecting a different model for each pays off. The performance of XAIC was more reliable overall, comparing very favourably to each of its competitors.

AIC Our methods XAIC and FAIC were derived as adaptations of AIC, and share its tendency to go for complex models as soon as there is some indication that their predictions might be worthwhile. This leads to good predictions on average, but also to inconsistency: when a simpler model contains the true distribution, AIC will continue to select more complex models with positive probability, no matter how large n grows. This may sometimes hurt predictive performance, because the accuracy of the estimated parameter will be smaller for more complex models; for details, we refer to Yang (2007a); Van Erven et al. (2007, 2012). XAIC makes a better assessment of the generalization error, even when the training and test inputs follow the same distribution, so that it overfits less than AIC and may achieve much better risks. FAIC differs from AIC in another way: its tendency to choose more complex models is strengthened in areas where many data points are available (so that the potential damage of picking an overly complex model is smaller), while it is suppressed when few data points are available (and the potential damage is much greater).

This tendency is also apparent in Table 2.1. In the first experiment, where a small model contains the true distribution, it causes FAIC to perform worse than AIC near $x = 0$. However, note that the vertical axis is logarithmic, so the difference appears larger than it is: when we average over the training input distribution, we find that FAIC performs better by a factor 20 in terms of squared risk.

In the multivariate experiments, XAIC again performs better than AIC, though the difference eventually disappears as n grows. With the notable ex-

Table 2.2: Multivariate: squared risk for different training sets; model selection

	Gaussian ($n = 60$)	uniform ($n = 60$)	spike-and-slab ($n = 60$)	Gaussian ($n = 100$)
XAIC_C	0.0119	0.0123	0.0144	0.0070
FAIC_C	0.0123	0.0127	0.0133	0.0077
AIC_C	0.0125	0.0126	0.0156	0.0070
BIC	0.0113	0.0128	0.0140	0.0073
BMS	0.0120	0.0126	0.0138	0.0075
cSIC	0.0119	0.0134	0.0138	0.0074
GCV	0.0129	0.0131	0.0153	0.0072
FIC	0.0196	0.0189	0.0241	0.0111

Table 2.3: Multivariate: squared risk for different training sets; model weighting/averaging

	Gaussian ($n = 60$)	uniform ($n = 60$)	spike-and-slab ($n = 60$)	Gaussian ($n = 100$)
XAIC_Cw	0.0099	0.0108	0.0114	0.0063
FAIC_Cw	0.0100	0.0110	0.0110	0.0066
AIC_Cw	0.0101	0.0108	0.0119	0.0063
BICw	0.0096	0.0106	0.0111	0.0062
BMA	0.0100	0.0107	0.0113	0.0061

ception of the spike-and-slab experiment, FAIC does not perform well here: in two of the experiments, it does worse than AIC. Part of the reason must be our observation at the end of Section 2.4: FAIC's estimate of the generalization error, while unbiased, may potentially have a larger variance than (X)AIC's estimate, and this is not always a good trade-off.

BIC and BMS/BMA BIC and BMS do not try to identify the model that will give the best predictions now, but instead attempt to find the most probable model given the data, which usually amounts to the simplest model containing the true distribution. This leads them to be conservative about selecting complex models. For similar reasons, Bayesian model averaging (BMA) puts only small weight on complex models. We see this in Figure 2.1, where BIC and BMA have good performance because they most often select the optimal second model (or in the case of BMA, give it the largest weight). However, for f_2 in Figure 2.2, it may be outperformed by FAIC or XAIC for test inputs away from the centre. In the multivariate experiments, XAIC often performs better than BMS/BMA, and rarely much worse; the only instance of the latter is for the spike-and-slab data, where FAIC outperforms both. (See Section 2.6.1 for further discussion of BMA.)

FIC In all our experiments, FIC obtained large squared risks, and we see in Table 2.1 that its selection behaviour was the opposite of FAIC: for extreme x, FIC often selects a more complex model than near $x = 0$. This seems to happen because FIC uses the most complex model's prediction at a given x to estimate each other model's bias. Because the most complex model will usually have a significant variance, this resulted in FIC being misled in many of the experiments we examined. In particular, in areas with few training inputs, FIC apparently usually believes the simpler models will perform badly because it attributes to them a large bias, so that the same model as elsewhere (or even a more complex one) is selected. Conversely, FIC was often observed to switch to an overly simple model near some input value where this model's estimate happened to coincide with that of the most complex model.

SIC SIC obtained large risks in the univariate experiments due to underfitting. Its results in three of the four multivariate experiments were competitive, however.

GCV Based on Leeb (2008), we expected GCV might be one of the strongest competitors to XAIC. This was not clearly reflected in our experiments, where its performance was very similar to that of AIC.

2.6 Discussion

2.6.1 Relation to the Bayesian predictive distribution

The quantity $\kappa_{x'}$ that occurs in FAIC has an interpretation in the Bayesian framework. If we do linear regression with known variance and a noninformative prior on β, then after observing X, $\mathbf{Y}$ and x', the predictive distribution of $\mathbf{y}'$ is $\mathbf{y}' \mid \mathbf{Y}, X, x' \sim \mathcal{N}(x'^{\top}\hat{\beta}, \sigma^2(1 + x'^{\top}(X^{\top}X)^{-1}x'))$. We see that $\kappa_{x'}$ and the variance of this predictive distribution obey a linear relation. Thus if BMA is allowed to give a distribution over output values as its prediction, then this distribution (a mixture of Gaussians with different variances) will reflect that some models' predictions are more reliable than others. However, if the predictive distribution must be summarized by a point prediction, then such information is likely to be lost. For instance, if the point prediction $\hat{y}'$ is to be evaluated with squared loss and $\hat{y}'$ is chosen to minimize the expected loss under the predictive distribution (as in our experiments in Section 2.5), then $\hat{y}'$ is a weighted average of posterior means for $\mathbf{y}'$ given x' (one mean for each model, weighted by its posterior probability). The predictive variances are not factored into $\hat{y}'$, so that in this scenario, BMA does not use the information captured by $\kappa_{X'}$ that XAIC and FAIC rely on.

This is not to say that BMA *should* use this information: the consideration of finding the most probable model (BMS, BIC) or the full distribution over models (BMA) is not affected by the purpose for which the model will be used, so it should not depend on the input values in the test data through $\kappa_{X'}$. This suggests that there is no XBIC analogue to XAIC. For Bayesian methods that

aim for good predictions, such as DIC (Spiegelhalter et al., 2002), BPIC (Ando, 2007) and WAIC (Watanabe, 2010), on the other hand, extra-sample and focused equivalents may exist. Gelman et al. (2014) give a theoretical and experimental comparison between AIC, DIC and WAIC.

2.6.2 Relation to covariate shift methods

We observed at the end of Section 2.4 that of the two data-dependent terms in XAIC, the log-likelihood is independent of X', while $\kappa_{X'}$ is (largely) unaffected by output values. An important practical consequence of this split between input and output values is that XAIC and FAIC look for models that give a good overall fit, not just a good fit at the test inputs. X' is then used to determine how well we can expect these models to generalize to the test set. So if we have two models and believe each to be able to give a good fit in a different region of the input space, then FAIC is not the proper tool for the task of finding these regions: FAIC considers global fit rather than local fit when evaluating a model, and within the model selects the maximum likelihood estimator, not an estimator specifically chosen for a local fit at input point x.

In this respect, our methods differ from those commonly used in the covariate shift literature (see Sugiyama and Kawanabe (2012); Pan and Yang (2010); some negative results are in Ben-David et al. (2010)), where typically a model (and an estimator within that model) is sought that will perform well on the test set only, using for example importance weighting. This is appropriate if we believe that no available model can give satisfactory results on both training and test inputs simultaneously. In situations where such models are believed to exist, our methods try to find them using all information in the training set.

2.7 Conclusions and future work

We have shown a bias in AIC when it is applied to supervised learning problems, and proposed XAIC and FAIC as versions of AIC which correct this bias. We have experimentally shown that these methods give better predictive performance than other methods in many situations.

We see several directions for future work. First, the practical usefulness of our methods needs to be confirmed by further experiments. Other future work includes considering other model selection methods: determining whether they are affected by the same bias that we found for AIC, whether such a bias can be removed (possibly leading to extra-sample and focused versions of those methods), and how these methods perform in simulation experiments and on real data. In particular, BPIC (Ando, 2007) is a promising candidate, as its derivation starts with a Bayesian equivalent of (2.1). (The same may be true for WAIC (Watanabe, 2010).) An XBPIC method would also be better able to deal with more complex models that a variant of AIC would have difficulty with, such as hierarchical Bayesian models, greatly increasing its practical applicability.

Appendix 2.A Regularity conditions and proofs

Assumption 2.3 (Regularity conditions)**.** *Items 1–4 correspond to the regularity assumptions behind AIC given by Shibata (1989), but rewritten to take the input variables into account. Item 5 is the assumption of asymptotic normality of the maximum likelihood estimator, which is also standard.*

1. $\Theta \subseteq \mathbf{R}^k$ *is open, and for sufficiently large* n *the gradient and Hessian of the log-likelihood function* $\ell(\theta) = \log g(\mathbf{Y} \mid X, \theta)$ *are well-defined for all* $\theta \in \Theta$ *with probability 1, and both are continuous;*

2. *For sufficiently large* n, $\mathbf{E}_{\mathbf{Y}|X} |\frac{\partial}{\partial\theta}\ell(\theta)| < \infty$ *and* $\mathbf{E}_{\mathbf{Y}|X} |\frac{\partial^2}{\partial\theta^2}\ell(\theta)| < \infty$*;*

3. *For sufficiently large* n*, there exists a unique* $\theta_o \in \Theta$ *such that* $\mathbf{E}_{\mathbf{Y}|X} \frac{\partial}{\partial\theta}\ell(\theta_o) = 0$*. For all* $\epsilon > 0$*, it satisfies*

$$\inf_{\theta:\|\theta-\theta_o\|>\epsilon} \ell(\theta_o) - \ell(\theta) \to \infty \qquad \textit{almost surely}$$

   as $n \to \infty$*;*

4. *For all* $\epsilon > 0$*, there is a* $\delta > 0$ *such that for sufficiently large* n*,*

$$\sup_{\|\theta-\theta_o\|<\delta} \Big| \mathbf{E}_{\mathbf{Y}|X}[\hat\theta(\mathbf{Y} \mid X) - \theta_o]^\top I(\theta \mid X)[\hat\theta(\mathbf{Y} \mid X) - \theta_o] - \operatorname{tr}\Big[J(\theta_o \mid X) I(\theta_o \mid X)^{-1}\Big] \Big| < \epsilon,$$

   where $I(\theta \mid X) = -\mathbf{E}_{\mathbf{Y}|X} \frac{\partial^2}{\partial\theta^2}\ell(\theta)$ *and* $J(\theta_o \mid X) = \mathbf{E}_{\mathbf{Y}|X}\Big[\frac{\partial}{\partial\theta}\ell(\theta_o)\Big]\Big[\frac{\partial}{\partial\theta}\ell(\theta_o)\Big]^\top$ *are continuous and positive definite.*

5. $\sqrt{n}(\hat\theta(\mathbf{Y} \mid X) - \theta_o) \xrightarrow{D} \mathcal{N}(0, \Sigma)$ *for some* Σ*.*

Proof of Theorem 2.1. This proof is adapted from the one in Burnham and Anderson (2002), with modifications to take X and X' into account. Derivation of an estimator for (2.3) starts with a Taylor expansion:

$$\begin{aligned}
&-2\log g(\mathbf{Y}' \mid X', \hat\theta(X, \mathbf{Y})) \\
&\quad = -2\log g(\mathbf{Y}' \mid X', \theta_o) - 2\left[\frac{\partial}{\partial\theta}\log g(\mathbf{Y}' \mid X', \theta_o)\right]^\top [\hat\theta(X, \mathbf{Y}) - \theta_o] \\
&\qquad - [\hat\theta(X, \mathbf{Y}) - \theta_o]^\top \left[\frac{\partial^2}{\partial\theta^2}\log g(\mathbf{Y}' \mid X', \theta_o)\right] [\hat\theta(X, \mathbf{Y}) - \theta_o] + r(\hat\theta),
\end{aligned}$$

where $r(\hat\theta)/\|\hat\theta(X, \mathbf{Y}) - \theta_o\|^2 \to 0$ as $\hat\theta(X, \mathbf{Y}) \to \theta_o$.

Next, we take the expectation $\mathbf{E}_{\mathbf{Y}'|X'}$. Given the regularity conditions on the model, θ_o minimizes $\mathbf{E}_{\mathbf{Y}'|X'} \log g(\mathbf{Y}' \mid X', \theta)$, so the linear term vanishes. (Note that we need this vanishing to hold for any X' (or equivalently, for any

single point x); this follows from the assumption that θ_o represents the true conditional data-generating distribution.) The coefficient of the quadratic term now becomes the conditional Fisher information at θ_o given X', so we have

$$-2\,\mathbf{E}_{\mathbf{Y}'|X'} \log g(\mathbf{Y}' \mid X', \hat{\theta}(X, \mathbf{Y})) = -2\,\mathbf{E}_{\mathbf{Y}'|X'} \log g(\mathbf{Y}' \mid X', \theta_o) + [\hat{\theta}(X, \mathbf{Y}) - \theta_o]^\top I(\theta_o \mid X')[\hat{\theta}(X, \mathbf{Y}) - \theta_o] + r(\hat{\theta}).$$

Rearranging the quadratic term and taking the other expectation, we obtain

$$-2\,\mathbf{E}_{\mathbf{Y}|X}\,\mathbf{E}_{\mathbf{Y}'|X'} \log g(\mathbf{Y}' \mid X', \hat{\theta}(X, \mathbf{Y})) = -2\,\mathbf{E}_{\mathbf{Y}'|X'} \log g(\mathbf{Y}' \mid X', \theta_o) + \operatorname{tr}\left\{ I(\theta_o \mid X') \left[\mathbf{E}_{\mathbf{Y}|X} [\hat{\theta}(X, \mathbf{Y}) - \theta_o][\hat{\theta}(X, \mathbf{Y}) - \theta_o]^\top \right] \right\} + \mathbf{E}_{\mathbf{Y}|X}\, r(\hat{\theta}). \quad (2.9)$$

The other matrix in the trace is the conditional covariance matrix of $\hat{\theta}(X, \mathbf{Y})$.

To proceed with the first term on the right hand side, we use Assumption 2.2. Then we have

$$-2\frac{n}{n'}\,\mathbf{E}_{\mathbf{Y}'|X'} \log g(\mathbf{Y}' \mid X', \theta_o) = -2\,\mathbf{E}_{\mathbf{Y}|X} \log g(\mathbf{Y} \mid X, \theta_o)$$

for a sample $(X, \mathbf{Y})$ of size n. (Here X still represents the values of the input variable in the training set, but $\mathbf{Y}$ conceptually represents a new sample.) Now only one X remains, so the rest of the derivation corresponds to that of standard AIC, which gives us

$$-2\,\mathbf{E}_{\mathbf{Y}|X} \log g(\mathbf{Y} \mid X, \theta_o) = -2\,\mathbf{E}_{\mathbf{Y}|X} \log g(\mathbf{Y} \mid X, \hat{\theta}(X, \mathbf{Y})) + k + o(1). \quad (2.10)$$

Multiplying (2.9) by n/n' and plugging in the above, we get

$$-2\frac{n}{n'}\,\mathbf{E}_{\mathbf{Y}|X}\,\mathbf{E}_{\mathbf{Y}'|X'} \log g(\mathbf{Y}' \mid X', \hat{\theta}(X, \mathbf{Y})) = -2\,\mathbf{E}_{\mathbf{Y}|X} \log g(\mathbf{Y} \mid X, \hat{\theta}(X, \mathbf{Y})) + k + \frac{n}{n'} \operatorname{tr}\left\{ I(\theta_o \mid X') \operatorname{Cov}(\hat{\theta}(X, \mathbf{Y}) \mid X) \right\} + \mathbf{E}_{\mathbf{Y}|X}\, \frac{n}{n'} r(\hat{\theta}) + o(1).$$

The term with the trace is what we called $\kappa_{X'}$.

By the assumed asymptotic normality of the maximum likelihood estimator, $\mathbf{E}_{\mathbf{Y}|X}\, n\|\hat{\theta}(X, \mathbf{Y}) - \theta_o\|^2$ converges to a constant, so that the first remainder term $\mathbf{E}_{\mathbf{Y}|X}(n/n')r(\hat{\theta}) = (1/n')o(1)$; because we additionally assumed the test set is either fixed or grows with the training set, this is again $o(1)$. This proves (2.7).

In the case of a linear model with fixed variance σ^2, the second-order Taylor approximation and the approximation in (2.10) are actually exact. □

Proof of Theorem 2.2. This proof follows a different path from the one above. It is adapted from the derivation of $\mathrm{AIC_C}$ in Burnham and Anderson (2002, Section 7.4.1). We first consider the case where the training set size $n' = 1$. Then X' becomes a vector (we choose to make it a column vector) and $\mathbf{Y}'$ a

scalar; we write x and $\mathbf{y}$ for these. Hats denote maximum likelihood estimates. For Gaussian densities, we get

$$\begin{aligned} T &= -2\,\mathbf{E}_{\mathbf{Y}|X}\,\mathbf{E}_{\mathbf{y}|x} \log g(\mathbf{y} \mid x, \hat{\theta}(X, \mathbf{Y})) \\ &= \mathbf{E}_{\mathbf{Y}|X}\,\mathbf{E}_{\mathbf{y}|x}\left[\log 2\pi\hat{\sigma}^2(X,\mathbf{Y}) + \frac{1}{\hat{\sigma}^2(X,\mathbf{Y})}\left(\mathbf{y} - x^\top\hat{\beta}(X,\mathbf{Y})\right)^2\right] \\ &= \mathbf{E}_{\mathbf{Y}|X} \log 2\pi\hat{\sigma}^2(X,\mathbf{Y}) + \mathbf{E}_{\mathbf{Y}|X}\,\frac{1}{\hat{\sigma}^2(X,\mathbf{Y})}\,\mathbf{E}_{\mathbf{y}|x}\left(\mathbf{y} - x^\top\hat{\beta}(X,\mathbf{Y})\right)^2. \end{aligned}$$

We will call the final term T'. Writing y_o for $\mathbf{E}_{\mathbf{y}|x}\,y$ and σ_o^2 for $\mathbf{y}$'s unknown variance, the inner expectation becomes

$$\begin{aligned} &\mathbf{E}_{\mathbf{y}|x}\left(\mathbf{y} - x^\top\hat{\beta}(X,\mathbf{Y})\right)^2 \\ &= \mathbf{E}_{\mathbf{y}|x}(\mathbf{y} - y_o)^2 + 2\left(y_o - x^\top\hat{\beta}(X,\mathbf{Y})\right)\mathbf{E}_{\mathbf{y}|x}(\mathbf{y} - y_o) + \left(y_o - x^\top\hat{\beta}(X,\mathbf{Y})\right)^2 \\ &= \sigma_o^2 + x^\top(\beta_o - \hat{\beta}(X,\mathbf{Y}))(\beta_o - \hat{\beta}(X,\mathbf{Y}))^\top x. \end{aligned}$$

Using the fact that $\hat{\beta}(X,\mathbf{Y})$ and $\hat{\sigma}^2(X,\mathbf{Y})$ are independent in this setting,

$$T' = \left[\mathbf{E}_{\mathbf{Y}|X}\,\frac{1}{\hat{\sigma}^2(X,\mathbf{Y})}\right] \cdot \left[\sigma_o^2 + x^\top \operatorname{Cov}(\hat{\beta}(X,\mathbf{Y}) \mid X)x\right].$$

The covariance matrix equals $\sigma_o^2(X^\top X)^{-1}$. Then we use that $n\hat{\sigma}^2/\sigma_o^2$ follows a chi-squared distribution with $n-k+1$ degrees of freedom (k is the number of free parameters in the model, which includes σ^2), and that $\mathbf{E}\,1/\chi^2_{n-k} = 1/(n-k-1)$:

$$\begin{aligned} T' &= \left[\mathbf{E}_{\mathbf{Y}|X}\,\frac{1}{\hat{\sigma}^2(X,\mathbf{Y})}\right]\left[\sigma_o^2 + \sigma_o^2 x^\top(X^\top X)^{-1}x\right] \\ &= \left[\mathbf{E}_{\mathbf{Y}|X}\,\frac{\sigma_o^2}{n\hat{\sigma}^2(X,\mathbf{Y})}\right]\left[n + nx^\top(X^\top X)^{-1}x\right] \\ &= \frac{n + nx^\top(X^\top X)^{-1}x}{n-k-1} \\ &= 1 + \frac{n + nx^\top(X^\top X)^{-1}x - (n-k-1)}{n-k-1} = 1 + \frac{k+\kappa_x}{n-k-1}, \end{aligned}$$

where $\kappa_x = nx^\top(X^\top X)^{-1}x + 1$. The reason for splitting off the 1 from the fraction is that $n(\mathbf{E}_{\mathbf{Y}|X}\log 2\pi\hat{\sigma}^2(X,\mathbf{Y}) + 1)$ is -2 times the maximized log-likelihood. Then we multiply by n and get the result in the stated form:

$$\begin{aligned} nT &= -2\,\mathbf{E}_{\mathbf{Y}|X} \log g(\mathbf{Y} \mid X, \hat{\theta}(X,\mathbf{Y})) + \frac{n(k+\kappa_x)}{n-k-1} \\ &= -2\,\mathbf{E}_{\mathbf{Y}|X} \log g(\mathbf{Y} \mid X, \hat{\theta}(X,\mathbf{Y})) + k + \kappa_x + \frac{(k+1)(k+\kappa_x)}{n-k-1}. \end{aligned}$$

The result for $n' > 1$ now follows by taking the average over all sample points in the test set on both sides. □

Proof of Proposition 2.3. Assume without loss of generality that the variance is known (as its inclusion does not affect the statement of the theorem) and that the basis is orthonormal with respect to the measure underlying $\mathbf{x}$ (that is, that $\mathbf{E}_{\mathbf{x}}\, \mathbf{x}\mathbf{x}^\top = I_k$). Then

$$\begin{aligned}\mathbf{E}_{\mathbf{x}}\, \kappa_{\mathbf{x}} &= n\, \mathbf{E}_{\mathbf{x}}\, \mathbf{x}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x} \\ &= n \operatorname{tr}(\mathbf{X}^\top \mathbf{X})^{-1} = \operatorname{tr}(\frac{1}{n} \mathbf{X}^\top \mathbf{X})^{-1},\end{aligned}$$

where orthonormality was used in the second equality. To compare the $\kappa_{\mathbf{x}}$ for this model with that of a submodel with one fewer entry in its design vectors, write

$$\frac{1}{n} \mathbf{X}^\top \mathbf{X} = \begin{bmatrix} \mathbf{A} & \mathbf{v} \\ \mathbf{v}^\top & \mathbf{d} \end{bmatrix}.$$

Note that by orthonormality, the expected value of this matrix is the identity matrix. We require that its inverse exists. Then for $\mathbf{d}' = (\mathbf{d} - \mathbf{v}^\top \mathbf{A}^{-1} \mathbf{v})^{-1}$,

$$\begin{aligned}\mathbf{E}\, \kappa_{\mathbf{x}} &= \mathbf{E} \operatorname{tr} \begin{bmatrix} \mathbf{A} & \mathbf{v} \\ \mathbf{v}^\top & \mathbf{d} \end{bmatrix}^{-1} \\ &= \mathbf{E} \operatorname{tr} \begin{bmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1} \mathbf{v} \mathbf{d}' \mathbf{v}^\top \mathbf{A}^{-1} & -\mathbf{A}^{-1} \mathbf{v} \mathbf{d}' \\ -\mathbf{d}' \mathbf{v}^\top \mathbf{A}^{-1} & \mathbf{d}' \end{bmatrix} \\ &= \mathbf{E} \operatorname{tr} \mathbf{A}^{-1} + \mathbf{E}\, \frac{\mathbf{v}^\top \mathbf{A}^{-2} \mathbf{v} + 1}{\mathbf{d} - \mathbf{v}^\top \mathbf{A}^{-1} \mathbf{v}} \\ &\geq \mathbf{E} \operatorname{tr} \mathbf{A}^{-1} + \mathbf{E}\, \frac{1}{\mathbf{d} - \mathbf{v}^\top \mathbf{A}^{-1} \mathbf{v}} \\ &\geq \mathbf{E} \operatorname{tr} \mathbf{A}^{-1} + \frac{1}{1 - \mathbf{E}\, \mathbf{v}^\top \mathbf{A}^{-1} \mathbf{v}} \\ &\geq \mathbf{E} \operatorname{tr} \mathbf{A}^{-1} + 1.\end{aligned}$$

This shows that adding an element to the design vector increases $\mathbf{E}\, \kappa_{\mathbf{x}}$ by at least one. For $k = 1$ (so that $\mathbf{A}$ is a 0×0 matrix), we have equality if and only if $\mathbf{d} = 1$ almost surely, which means that for $\mathbf{x}_1$ (the first and only entry of design vector $\mathbf{x}$), we must have $\mathbf{x}_1 = \pm 1$ almost surely. For $k \geq 2$, because $\mathbf{A}^{-1}$ is positive definite, equality requires that $\mathbf{v}$ is the zero vector almost surely (in addition to the same requirement as above on all $\mathbf{x}_i$). But this can only be satisfied if $\mathbf{x}_i \mathbf{x}_k = 0$ almost surely for all $i < k$, which is incompatible with the conditions on $\mathbf{x}_1$ and $\mathbf{x}_k$. □

Chapter 3

Bayesian Inconsistency under Misspecification

We empirically show that Bayesian inference can be inconsistent under misspecification in simple linear regression problems, both in a model averaging/selection and in a Bayesian ridge regression setting. We use the standard linear model, which assumes homoskedasticity, whereas the data are heteroskedastic (though significantly, there are no outliers), and observe that the posterior puts its mass on ever more high-dimensional models as the sample size increases. To remedy the problem, we equip the likelihood in Bayes' theorem with an exponent called the learning rate, and we propose the *SafeBayesian* method to learn the learning rate from the data. SafeBayes tends to select small learning rates as soon the standard posterior is not 'cumulatively concentrated', and its results on our data are quite encouraging.

In this chapter, we focus on introducing both the problem and the solution we propose, and we provide our main experiments with Bayes and SafeBayes. The discussion of Bayesian inconsistency will be continued in Chapters 4 (analysing the underlying reasons for the behaviour of Bayes and SafeBayes) and 5 (providing several additional experiments to check the generality of our findings). An overview of these three chapters is provided in Section 3.1.1, and an 'executive summary' of all the experiments in Chapters 3 and 5 combined is provided in Section 3.5.5 at the end of Chapter 3.

3.1 Introduction

The problem We empirically demonstrate the inconsistency of Bayes factor model selection, model averaging and Bayesian ridge regression under model misspecification on a simple linear regression problem with random design. We sample data $(X_1, Y_1), (X_2, Y_2), \ldots$ i.i.d. from a distribution P^*, where $X_i = (X_{i1}, \ldots, X_{ip_{\max}})$ are high-dimensional vectors, and we allow $p_{\max} = \infty$. We use nested models $\mathcal{M}_0, \mathcal{M}_1, \ldots$ where $\mathcal{M}_p$ is a standard linear model, consist-

ing of conditional distributions $P(\cdot \mid \beta, \sigma^2)$ expressing that

$$Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij} + \epsilon_i \tag{3.1}$$

is a linear function of $p \leq p_{\max}$ covariates with additive independent Gaussian noise $\epsilon_i \sim N(0, \sigma^2)$. We equip each of these models with standard priors on coefficients and the variance, and also put a discrete prior on the models themselves. $\mathcal{M} := \bigcup_{p=0,\ldots,p_{\max}} \mathcal{M}_p$ does not contain the conditional 'ground truth' $P^*(Y \mid X)$ (hence the model is 'misspecified'), but it does contain a $\tilde{P}$ that is 'best' in several respects: it is closest to P^* in KL (Kullback-Leibler) divergence, it represents the true regression function (leading to the best squared error loss predictions among all $P \in \mathcal{M}$) and it has the true marginal variance (explained in Section 3.2.3). Yet, while $\tilde{P} \in \mathcal{M}_0$ and $\mathcal{M}_0$ receives substantial prior mass, as n increases, the posterior puts most of its mass on complex $\mathcal{M}_p$'s with higher and higher p's, and, conditional on these $\mathcal{M}_p$'s, at distributions which are very far from P^* both in terms of KL divergence and in terms of L_2 risk, leading to bad predictive behaviour in terms of squared error. Figures 3.1 and 3.2 illustrate a particular instantiation of our results, obtained when X_{ij} are polynomial functions of S_i and $S_i \in [-1, 1]$ uniformly i.i.d. We also show comparably bad predictive behaviour for various versions of Bayesian ridge regression, involving just a single, high-but-finite dimensional model. In that case Bayes eventually recovers and concentrates on $\tilde{P}$, but only at a sample size that is incomparably larger than what can be expected if the model is correct.

These findings contradict the folk wisdom that, if the model is incorrect, then "Bayes tends to concentrate on neighbourhoods of the distribution(s) in $\mathcal{M}$ that is/are closest to P^* in KL divergence." Indeed, the strongest actual theorems to this end that we know of, (Kleijn and Van der Vaart, 2006; De Blasi and Walker, 2013; Ramamoorthi et al., 2013), hold, as the authors emphasize, under regularity conditions that are substantially stronger than those needed for consistency when the model is correct (as by e.g. Ghosal et al. (2000) or Zhang (2006a)), and our example shows that consistency may fail to hold even in relatively simple problems.

The solution: Generalized posterior and SafeBayes Bayesian updating can be enhanced with a *learning rate* η, an idea put forward independently by several authors (Vovk, 1990; McAllester, 2003; Barron and Cover, 1991; Walker and Hjort, 2002; Zhang, 2006a) and suggested as a tool for dealing with misspecification by Grünwald (2011; 2012). η trades off the relative weight of the prior and the likelihood in determining the *η-generalized posterior*, where $\eta = 1$ corresponds to standard Bayes and $\eta = 0$ means that the posterior always remains equal to the prior. When choosing the 'right' η, which in our case is significantly smaller than 1 but of course not 0, η-generalized Bayes becomes competitive again. In general, the optimal η depends on the underlying ground truth P^*, and the problem has always been how to determine the optimal η empirically, from the data.

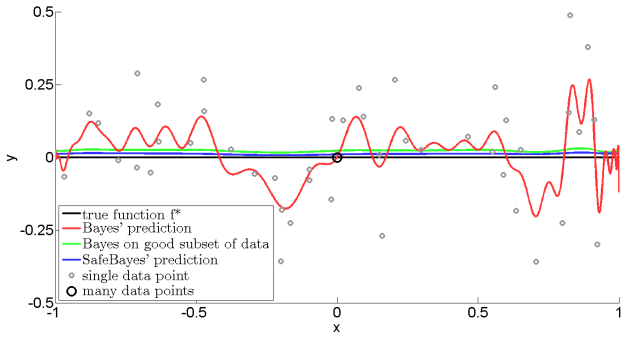


Figure 3.1: The conditional expectation $\mathbf{E}[Y \mid X]$ according to the full Bayesian posterior based on a prior on models $\mathcal{M}_0, \ldots, \mathcal{M}_{50}$ with polynomial basis functions, given 100 data points sampled i.i.d. $\sim P^*$ (about 50 of which are at $(0,0)$). Standard Bayes overfits, not as dramatically as maximum likelihood or unpenalized least squares, but still enough to show dismal predictive behaviour as in Figure 3.2. In contrast, SafeBayes (which chooses learning rate $\eta \approx 0.4$ here) and standard Bayes trained only at the points for which the model is correct (not $(0,0)$) both perform very well.

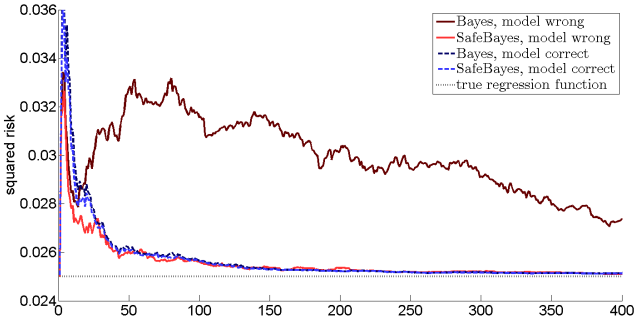


Figure 3.2: The expected squared error risk (defined in (3.3)), obtained when predicting by the full Bayesian posterior (brown curve), the SafeBayesian posterior (red curve) and the optimal predictions (black dotted curve), as a function of sample size for the setting of Figure 3.1. SafeBayes is the R-log-version of SafeBayes defined in Section 3.4.2. Precise definitions and further explanation in Section 3.5.1 and Section 3.5.2.

Recently, Grünwald (2012) proposed the *SafeBayesian* algorithm for learning η, and theoretically showed that it achieves good convergence rates in terms of KL divergence on a variety of problems. Here we show empirically that SafeBayes performs excellently in our regression setting, being competitive with standard Bayes if the model is correct and very significantly outperforming not just standard Bayes, but also cross-validation and approaches such as AIC when the model is incorrect. We do this by providing a wide range of experiments, varying parameters of the problem such as the priors and the true regression function and studying various performance indicators such as the squared error risk, the posterior on the variance etc.

We note that a Bayesian's (and our) first instinct would be to learn η itself in a Bayesian manner instead. Yet this does not solve the problem, as we show in Section 3.5.4, where we consider a setting in which $1/\eta$ turns out to be exactly equivalent to the λ regularization parameter in the Bayesian Lasso and ridge regression approaches. We find that selecting η by (empirical) Bayes, as suggested by e.g. Park and Casella (2008), does not nearly regularize enough in our misspecification experiments. In the Bayesian ridge regression setting with fixed variance, the SafeBayesian algorithm becomes very similar to learning λ by cross-validation with squared-error loss, as is standard in frequentist ridge regression (cross-validation with a logarithmic score does *not* work however). In the varying variance case, there is no such straightforward interpretation of SafeBayes.

The type of misspecification The models are misspecified in that they make the standard assumption of homoskedasticity — σ^2 is independent of X — whereas in reality, under P^*, there is heteroskedasticity, there being a region of X with low and a region with (relatively) high variance. Specifically, in our simplest experiment the 'true' P^* is defined as follows: at each i, toss a fair coin. If the coin lands heads, then sample X_i from a uniform distribution on $[-1, 1]$, and set $Y_i = 0 + \epsilon_i$, where $\epsilon_i \sim N(0, \sigma_0^2)$. If the coin lands tails, then set $(X_i, Y_i) = (0, 0)$, so that there is no variance at all. The 'best' conditional density $\tilde{P}$, closest to $P^*(Y \mid X)$ in KL divergence, representing the true regression function $Y = 0$ and reliable in the sense of Section 3.2.3, is then given by (3.1) with all β's set to 0 and $\tilde{\sigma}^2 = \sigma_0^2/2$. In a typical sample of length n, we will thus have approximately $n/2$ points with X_i uniform and Y_i normal with mean 0, and approximately $n/2$ points with $(X_i, Y_i) = (0, 0)$. These points seem 'easy' since they lie exactly on the regression function one would hope to learn; but they really wreak severe havoc.

The in-liers cause the problem While it is well-known that in the presence of outliers, Gaussian assumptions on the noise lead to problems, both for frequentist and Bayesian procedures, in the present problem we have *'in-liers'* rather than outliers. Also, if we slightly modify the setup so that homoskedasticity holds, standard Bayes starts behaving excellently, as again depicted in Figures 3.1 and 3.2. Finally, while the figure shows what happens for polynomials, we used independent multivariate X's rather than nonlinear basis functions in the main experiments below, getting essentially the same results. All

this indicates that the inconsistency is really caused by misspecification, in particular the presence of in-liers, and not by anything else. The setup is inspired by the work of Grünwald and Langford (2004, 2007), who gave a mathematical proof that Bayesian inference can be inconsistent under misspecification in a related but much more artificial classification setting. Here we show that this can also happen in a much more natural regression setting. The setting being more natural, it is also harder to analyse, and we only demonstrate the inconsistency empirically.

3.1.1 Overview of Chapters 3 to 5

KL-associated inference tasks Section 3.2 introduces our regression setting and the main concepts needed to understand our results. A crucial point here is that, if Bayesian (or other likelihood-based methods) converge at all to a distribution in the model $\mathcal{M}$, this distribution (often called the 'pseudo-truth') is the $\tilde{P} \in \mathcal{M}$ that minimizes KL divergence to the true distribution P^*. While the minimum KL divergence point is often not of intrinsic interest, for some (not all) models, $\tilde{P}$ can be of interest for other reasons as well (Royall and Tsou, 2003): there may be *associated* inference tasks for which $\tilde{P}$ is suitable as well. For standard linear models with fixed σ^2, the main associated task is squared error prediction: the KL-optimal $\tilde{P}$ is also optimal, among all $P \in \mathcal{M}$, in terms of squared error prediction risk. If additionally σ^2 becomes a free parameter, then it is also reliable, which roughly means that it is optimal in determining its own squared error prediction quality (Section 3.2.3; we have a lot more to say about associated inference tasks in Section 4.3). Thus, whenever one is prepared to work with linear models and one is interested in squared error risk or reliability, then Bayesian inference would seem the way to go, even if one suspects misspecification... at least if there is consistency.

The SafeBayesian algorithm Section 3.3 introduces the η-generalized posterior and instantiates it to the linear model. Section 3.4 introduces the 'SafeBayesian' algorithm, which learns η from the data. This is done via Dawid's (1984) *prequential* view on Bayesian inference. We then provide four instantiations of the SafeBayes method to linear models.

Section 3.5 discusses our experiments. We first provide the necessary preparation in Section 3.5.1 and 3.5.2. Section 3.5.3 gives the results of our first experiment, a comparison of Bayesian and SafeBayesian model averaging and selection in two settings, one with a correct model and one with a model corrupted by 50% easy points as above, but with independent Gaussian rather than polynomial inputs. Section 3.5.4 repeats these experiments for a Bayesian ridge regression setting, Section 3.5.5 provides an 'executive summary'. In all experiments SafeBayesian methods behave much better in terms of squared error risk and reliability than standard Bayes if the model is incorrect, and hardly worse (sometimes still better) than standard Bayes if the model is correct.

Good vs. bad misspecification: Nonconcentration and hypercompression In and of itself, the fact that one obtains inconsistency with homoskedastic

models and heteroskedastic data may not be very surprising; indeed, whether similar phenomena occur in real-world data needs further study. The main strength of our example is rather that it clearly shows what can happen in principle, and indicates how one may go about solving it. We explain this in Section 4.1 in Chapter 4, in particular on the basis of Figure 4.1 on page 74, *the essential picture to understand the phenomenon*. Inconsistency can only arise under a 'bad' form of misspecification, depicted by the figure. Under bad misspecification, the posterior may *fail to concentrate*, and this causes trouble. As a theoretical contribution of this chapter, we show in this section that, under some conditions, a Bayesian strongly believes that her posterior will, in some sense, concentrate fast. Indeed, SafeBayes will only select $\eta \ll 1$ if the standard posterior is nonconcentrated, and may thus be (loosely) viewed as a particular 'prior predictive check'.

Posterior nonconcentration in turn can lead to 'hypercompression', the phenomenon that the Bayes predictive distribution behaves *substantially better* under a logarithmic scoring rule than the best distribution $\tilde{P} \in \mathcal{M}$; this can happen because the Bayes predictive distribution — a mixture of elements of $\mathcal{M}$ — behaves substantially differently from any of the elements of $\mathcal{M}$. Somewhat paradoxically (Section 4.1.3), Bayes' overly good log-loss behaviour is exactly what causes it to perform badly for the associated inference tasks (squared error prediction and reliability, in our case). Thus, there can be an inherent tension between behaviour under log-loss and behaviour under its associated tasks, a discrepancy which one can measure by the *mixability gap* (Section 4.1.4), a theoretical concept introduced by Van Erven et al. (2011) and Grünwald (2012). If one is interested in log-loss, standard Bayes is just fine; the SafeBayesian algorithm should be used if one wants to optimize behaviour against the associated tasks. Of course, whether such a task-dependent modification of Bayes is desirable needs discussion, which we provide in Section 4.3.

Additional experiments In Chapter 5 we provide a battery of experiments to check the robustness of our results. Specifically, we investigate what happens if we vary our models and priors (using e.g. a fixed σ^2 and standard priors used in the regression literature), our methods, and if we vary the data-generating distribution using e.g. 'easy' points that are close to, but not exactly $(0, 0)$. Our main conclusion here is that, of the four versions of SafeBayes which we propose, one is uncompetitive and among the other three, there is no clear winner — although they consistently outperform Bayes under misspecification. Furthermore we show that AIC, BIC and cross-validation also have serious problems in our regression setup.

3.2 Preliminaries

3.2.1 Setting, logarithmic risk, optimal distribution

In this chapter we consider data $Z^n = Z_1, Z_2, \ldots, Z_n \sim$ i.i.d. P^*, where each $Z_i = (X_i, Y_i)$ is an independently sampled copy of $Z = (X, Y)$, X taking val-

ues in some set $\mathcal{X}$, Y taking values in $\mathcal{Y}$ and $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. We are given a *model* $\mathcal{M} = \{P_\theta \mid \theta \in \Theta\}$ parameterized by (possibly infinite-dimensional) Θ, and consisting of conditional distributions $P_\theta(Y \mid X)$, extended to n outcomes by independence. For simplicity we assume that all P_θ have corresponding conditional densities f_θ, and similarly, the conditional distribution $P^*(Y \mid X)$ has a conditional f^*, all with respect to the same underlying measure. While we do not assume $P^*(Y \mid X)$ to be in (or even 'close' to) $\mathcal{M}$, we want to learn, from given data Z^n, a 'best' (in a sense to be defined below) element of $\mathcal{M}$, or at least, a distribution on elements of $\mathcal{M}$ that can be used to make predictions about future data. While our experiments focus on linear regression, the discussion in this section holds for general conditional density models. The logarithmic score, henceforth abbreviated to *log-loss*, is defined in the standard manner: the loss incurred when predicting Y based on density $f(\cdot \mid x)$ and Y takes on value y, is given by $-\log f(y \mid x)$. A central quantity in our setup is then the *expected log-loss* or *log-risk*, defined as

$$\textsc{risk}^{\log}(\theta) := \mathbf{E}_{(X,Y)\sim P^*}[-\log f_\theta(Y \mid X)],$$

where here as in the remainder of this chapter, log denotes the natural logarithm.

We let P^*_X be the marginal distribution of X under P^*. The *Kullback-Leibler (KL) divergence* $D(P^* \| P_\theta)$ between P^* and conditional distribution P_θ is defined as the expectation, under $X \sim P^*_X$, of the KL divergence between P_θ and the 'true' conditional $P^*(Y \mid X)$: $D(P^* \| P_\theta) = \mathbf{E}_{X\sim P^*_X}[D(P^*(\cdot \mid X) \,\|\, P_\theta(\cdot \mid X))]$. A simple calculation shows that for any θ, θ',

$$D(P^* \| P_\theta) - D(P^* \| P_{\theta'}) = \textsc{risk}^{\log}(\theta) - \textsc{risk}^{\log}(\theta'),$$

so that the closer P_θ is to P^* in terms of KL divergence, the smaller its log-risk, and the better it is, on average, when used for predicting under the log-loss.

Now suppose that $\mathcal{M}$ contains a unique distribution that is closest, among all $P \in \mathcal{M}$ to P^* in terms of KL divergence. We denote such a distribution, if it exists, by $\tilde{P}$. Then $\tilde{P} = P_\theta$ for at least one $\theta \in \Theta$; we pick any such θ and denote it by $\tilde{\theta}$, i.e. $\tilde{P} = P_{\tilde{\theta}}$, and note that it also minimizes the log-risk:

$$\textsc{risk}^{\log}(\tilde{\theta}) = \min_{\theta\in\Theta} \textsc{risk}^{\log}(\theta) = \min_{\theta\in\Theta} \mathbf{E}_{(X,Y)\sim P^*}[-\log f_\theta(Y \mid X)]. \tag{3.2}$$

We shall call such a $\tilde{\theta}$ *(KL-)optimal*.

Since, in regions of about equal prior density, the log Bayesian posterior density is proportional to the log likelihood ratio, we hope that, given enough data, with high P^*-probability, the posterior puts most mass on distributions that are close to $P_{\tilde{\theta}}$ in KL divergence, i.e. that have log-risk close to optimal. Indeed, all existing consistency theorems for Bayesian inference under misspecification express concentration of the posterior around $P_{\tilde{\theta}}$.

3.2.2 A special case: The linear model

Fix some $p_{\max} \in \{0, 1, \ldots\} \cup \{\infty\}$. We observe data $Z_1, \ldots, Z_n$ where $Z_i = (X_i, Y_i)$, $Y_i \in \mathbf{R}$ and $X_i = (1, X_{i1}, \ldots, X_{ip_{\max}}) \in \mathbf{R}^{p_{\max}+1}$. Note that this is as in (3.1) but from now on we adopt the standard convention to take $X_{0i} \equiv 1$ as a dummy random variable. We denote by $\mathcal{M}_p = \{P_{p,\beta,\sigma^2} \mid (p, \beta, \sigma^2) \in \Theta_p\}$ the standard linear model with parameter space $\Theta_p := \{(p, \beta, \sigma^2) \mid \beta = (\beta_0, \ldots, \beta_p)^\top \in \mathbf{R}^{p+1}, \sigma^2 > 0\}$, where the entry p in (p, β, σ^2) is redundant but included for notational convenience. We let $\Theta = \bigcup_{p=0,\ldots,p_{\max}} \Theta_p$. $\mathcal{M}_p$ states that for all i, (3.1) holds, where $\epsilon_1, \epsilon_2, \ldots \sim$ i.i.d. $N(0, \sigma^2)$. When working with linear models $\mathcal{M}_p$, we are usually interested in finding parameters β that predict well in terms of the *squared error loss function* (henceforth abbreviated to *square-loss*): the square-loss on data (X_i, Y_i) is $(Y_i - \sum_{j=0}^{p} \beta_j X_{ij})^2 = (Y_i - X_i\beta)^2$. We thus want to find the distribution minimizing the expected square-loss, i.e. *squared error risk* (henceforth abbreviated to 'square-risk') relative to the underlying P^*:

$$\text{RISK}^{\text{sq}}(p, \beta) := \mathbf{E}_{(X,Y)\sim P^*}(Y - \mathbf{E}_{p,\beta,\sigma^2}[Y \mid X])^2 = \mathbf{E}_{(X,Y)\sim P^*}(Y - \sum_{j=0}^{p} \beta_j X_j)^2, \tag{3.3}$$

where $\mathbf{E}_{p,\beta,\sigma^2}[Y \mid X]$ abbreviates $\mathbf{E}_{Y\sim P_{p,\beta,\sigma^2}|X}[Y]$. Since this quantity is independent of the variance σ^2, σ^2 is not used as an argument of RISK^{sq}.

3.2.3 KL-associated prediction tasks for the linear model

Suppose that an optimal $\tilde{P} \in \mathcal{M}$ exists in the regression model. We denote by $\tilde{p}$ the smallest p such that $\tilde{P} \in \mathcal{M}_p$, and define $\tilde{\sigma}^2, \tilde{\beta}$ such that $\tilde{P} = P_{\tilde{p},\tilde{\beta},\tilde{\sigma}^2}$. A straightforward computation shows that for all $(p, \beta, \sigma^2) \in \Theta$:

$$\text{RISK}^{\log}((p, \beta, \sigma^2)) = \frac{1}{2\sigma^2}\text{RISK}^{\text{sq}}((p, \beta)) + \frac{1}{2}\log(2\pi\sigma^2), \tag{3.4}$$

so that the (p, β) achieving minimum log-risk for each fixed σ^2 is equal to the (p, β) with the minimum square-risk. In particular, $(\tilde{p}, \tilde{\beta}, \tilde{\sigma}^2)$ must minimize not just log-risk, but also square-risk. Moreover, the conditional expectation $\mathbf{E}_{P^*}[Y \mid X]$ is known as the *true regression function*. It minimizes the square-risk among all conditional distributions for $Y \mid X$. Together with (3.4) this implies that, if there is some (p, β) such that $\mathbf{E}[Y \mid X] = \sum_{j=0}^{p} \beta_j X_j = X\beta$, i.e. (p, β) represents the true regression function, then $(\tilde{p}, \tilde{\beta})$ also represents the true regression function. In all our examples, this will be the case: the model is misspecified only in that the true noise is heteroskedastic; but the model does invariably contain the true regression function.

Moreover, for each fixed (p, β), the σ^2 minimizing $\text{RISK}^{\log}$ is, as follows by differentiation, given by $\sigma^2 = \text{RISK}^{\text{sq}}(p, \beta)$. In particular, this implies that

$$\tilde{\sigma}^2 = \text{RISK}^{\text{sq}}(\tilde{p}, \tilde{\beta}), \tag{3.5}$$

or in words: the KL-optimal model variance $\tilde{\sigma}^2$ is equal to the true expected (marginal, not conditioned on X) square-risk obtained if one predicts with the optimal $(\tilde{p}, \tilde{\beta})$. This means that the optimal $(\tilde{p}, \tilde{\beta}, \tilde{\sigma}^2)$ is *reliable* in the sense of Grünwald (1998, 1999): its self-assessment about its square-loss performance is correct, independently of whether $\tilde{\beta}$ is equal to the true regression function or not. In other words, $(\tilde{p}, \tilde{\beta}, \tilde{\sigma}^2)$ *correctly predicts how well it predicts*.

Summarizing, for misspecified models, $(\tilde{p}, \tilde{\beta}, \tilde{\sigma}^2)$ is optimal not just in KL/log-risk sense, but also in terms of square-risk and in terms of reliability; in our examples, it also represents the true regression function. We say that, for linear models, square-risk optimality, square-risk reliability and regression-function consistency are *KL-associated prediction tasks*: if (as we hope Bayes will do, but as we will see sometimes does not) we can find the KL-optimal $\tilde{\theta}$, we automatically behave well in these associated tasks as well.

3.3 The generalized posterior

General losses The original generalized posterior is a concept going back at least to Vovk (1990) and has been developed mainly within the so-called (frequentist) *PAC-Bayesian* framework (McAllester, 2003; Seeger, 2002; Catoni, 2007; Audibert, 2004; Zhang, 2006b; see also Bissiri et al. (2013) and the discussion in Section 4.3). It is defined relative to a prior on *predictors* rather than probability distributions. Depending on the decision problem at hand, predictors can be e.g. classifiers, regression functions or probability densities. Formally, we are given an abstract space of predictors represented by a set Θ, which obtains its meaning in terms of a loss function $\ell : \mathcal{Z} \times \Theta \to \mathbf{R}$, writing $\ell_\theta(z)$ as shorthand for $\ell(z, \theta)$. Following e.g. Zhang (2006b), for any prior Π on Θ with density π relative to some underlying measure ρ, we define the *generalized Bayesian posterior with learning rate η relative to loss function ℓ*, denoted as $\Pi \mid Z^n, \eta$, as the distribution on Θ with density

$$\pi(\theta \mid z^n, \eta) := \frac{e^{-\eta \sum_{i=1}^n \ell_\theta(z_i)} \pi(\theta)}{\int e^{-\eta \sum_{i=1}^n \ell_\theta(z_i)} \pi(\theta) \rho(d\theta)} = \frac{e^{-\eta \sum_{i=1}^n \ell_\theta(z_i)} \pi(\theta)}{\mathbf{E}_{\theta \sim \Pi}[e^{-\eta \sum_{i=1}^n \ell_\theta(z_i)}]}. \tag{3.6}$$

Thus, if θ_1 fits the data better than θ_2 by a difference of ϵ according to loss function ℓ, then their posterior ratio is larger than their prior ratio by an amount exponential in ϵ, where the larger η, the larger the influence of the data as compared to the prior.

If $z_i = (x_i, y_i)$ with $y_i \in \mathbf{R}$ and $x_i = (1, x_{i1}, \ldots, x_{ip})$, and the goal is to predict y_i given x_i, then we may take as our prediction model e.g. the set of linear predictors that predict y_i by $\sum \beta_j x_{ij} = x_i \beta$, and as our loss function the squared error loss, $\ell_\beta(x_i, y_i) = (y_i - x_i \beta)^2$. We may then study the behaviour of such a procedure in its own right, irrespective of a Bayesian misspecification interpretation; the experiments we perform in Section 5.1.1 can be interpreted in this manner.

Log-loss and likelihood Now if the set Θ represents a model of (conditional) distributions $\mathcal{M} = \{P_\theta \mid \theta \in \Theta\}$, we may set, for $z_i = (x_i, y_i)$, $\ell_\theta(z_i) = -\log f_\theta(y_i \mid x_i)$ to be the log-loss as defined above. In this special case, the definition of η-generalized posterior specializes to the definition of 'generalized posterior' as known within the Bayesian literature (Walker and Hjort, 2002; Zhang, 2006a):

$$\pi(\theta \mid z^n, \eta) = \frac{(f(y^n \mid x^n, \theta))^\eta \pi(\theta)}{\int (f(y^n \mid x^n, \theta))^\eta \pi(\theta) \rho(d\theta)} = \frac{(f(y^n \mid x^n, \theta))^\eta \pi(\theta)}{\mathbf{E}_{\theta \sim \Pi}[(f(y^n \mid x^n, \theta))^\eta]}. \quad (3.7)$$

Again, the larger η, the larger the influence of the likelihood. Obviously $\eta = 1$ corresponds to standard Bayesian inference, whereas if $\eta = 0$ the posterior is equal to the prior and nothing is ever learned. Our algorithm for learning η will usually end up with values in between. It has long been known that in model selection and nonparametric settings, there is an issue with consistency proofs for full Bayes, Bayes MAP and MDL if we take the standard $\eta = 1$, and indeed, this is part of the reason why the generalized posterior in the form (3.7) was derived in the first place: for example, Barron and Cover (1991) give general consistency theorems for 2-part MDL (closely related to Bayes MAP) and note that they hold for any $\eta < 1$; but for $\eta = 1$, additional assumptions must be made. Zhang (2006a) gives an explicit example in which the posterior shows anomalous behaviour at $\eta = 1$. A connection to misspecification was first made by Grünwald (2011) (see Section 4.3.1) and Grünwald (2012).

Generalized predictive distribution We also define the predictive distribution based on the η-generalized posterior (3.7) as a generalization of the standard definition as follows: for $m \geq 0, m' \geq m$, we set

$$\begin{aligned} &\bar{f}(y_i, \ldots, y_{i+m} \mid x_i, \ldots, x_{i+m'}, z^{i-1}, \eta) \\ &\quad := \mathbf{E}_{\theta \sim \Pi \mid z^{i-1}, \eta}[f(y_i, \ldots, y_{i+m} \mid x_i, \ldots, x_{i+m'}, \theta)] \\ &\quad = \mathbf{E}_{\theta \sim \Pi \mid z^{i-1}, \eta}[f(y_i, \ldots, y_{i+m} \mid x_i, \ldots, x_{i+m}, \theta)]. \end{aligned} \quad (3.8)$$

where the first equality is a definition and the second follows by our i.i.d. assumption. We always use the bar-notation $\bar{f}$ to indicate marginal and predictive distributions, i.e. distributions on data that are arrived at by integrating out parameters. If $\eta = 1$ then $\bar{f}$ and π become the standard Bayesian predictive density and posterior, and if it is clear from the context that we consider $\eta = 1$, we leave out the η in the notation.

> The generalized posterior is created by exponentiating the likelihood according to individual elements $\theta \in \Theta = \bigcup_p \Theta_p$ in the model and renormalizing, which is not the same as exponentiating marginal likelihoods and renormalizing. In particular, $\pi(p \mid z^n, \eta)$ as given by (3.10) is in general *not* proportional to $(\bar{f}(y^n \mid x^n, p))^\eta \pi(p)$. Similarly, for generalized marginal distributions, as soon as $\eta \neq 1$, we have that in general
>
> $$\bar{f}(y_i, y_{i+1} \mid x_i, x_{i+1}, z^{i-1}, \eta) \neq \bar{f}(y_i \mid x_i, z^{i-1}, \eta) \cdot \bar{f}(y_{i+1} \mid x_{i+1}, z^i, \eta),$$

unlike for the standard Bayesian marginal distribution for which equality holds (in Section 4.2 we encounter a further modification of the generalized posterior whose marginals do satisfy this product rule).

3.3.1 Instantiation to linear model selection and averaging

Now consider again a linear model $\mathcal{M}_p$ as defined in Section 3.2.3. We instantiate the generalized posterior and its marginals for this model. With prior $\pi(\beta, \sigma^2 \mid p)$ taken relative to Lebesgue measure, (3.7) specializes to:

$$\pi(\beta, \sigma \mid z^n, p, \eta) = \frac{(2\pi\sigma^2)^{-n\eta/2} e^{-\frac{\eta}{2\sigma^2}\sum_{i=1}^n (y_i - x_i\beta)^2} \pi(\beta, \sigma \mid p)}{\int (2\pi\sigma^2)^{-n\eta/2} e^{-\frac{\eta}{2\sigma^2}\sum_{i=1}^n (y_i - x_i\beta)^2} \pi(\beta, \sigma \mid p)\, d\beta\, d\sigma}.$$

Note that in the numerator $1/\sigma^2$ and η are interchangeable in the exponent, but not in the factor in front: their role is subtly different. For Bayesian inference with a sequence of models $\mathcal{M} = \bigcup_{p=0,\ldots,p_{\max}} \mathcal{M}_p$, with $\pi(p)$ a probability mass function on $p \in \{0, \ldots, p_{\max}\}$, we get:

$$\begin{aligned}\pi(\theta \mid z^n, \eta) &= \frac{f(y^n \mid x^n, \theta)^\eta \pi(\theta)}{\int_{\theta\in\Theta} f(y^n \mid x^n, \theta)^\eta \pi(\theta) \rho(d\theta)} \quad \text{with} \quad \theta = (\beta, \sigma^2, p) \\ &= \frac{(2\pi\sigma^2)^{-n\eta/2} e^{-\frac{\eta}{2\sigma^2}\sum_{i=1}^n (y_i - x_i\beta)^2} \pi(\beta, \sigma \mid p)\pi(p)}{\sum_{p=0}^{p_{\max}} \int (2\pi\sigma^2)^{-n\eta/2} e^{-\frac{\eta}{2\sigma^2}\sum_{i=1}^n (y_i - x_i\beta)^2} \pi(\beta, \sigma \mid p)\pi(p)\, d\beta\, d\sigma}\end{aligned} \tag{3.9}$$

The total generalized posterior probability of model $\mathcal{M}_p$ then becomes:

$$\pi(p \mid z^n, \eta) = \int \pi(\beta, \sigma, p \mid z^n, \eta)\, d\beta\, d\sigma. \tag{3.10}$$

Analogously to (3.8), for given p, we define the η-generalized Bayesian predictive distribution as:

$$\begin{aligned}\bar{f}(y_i^{i+m} \mid x_i^{i+m'}, z^{i-1}, p, \eta) &:= \mathbf{E}_{\beta,\sigma^2\sim\Pi|z^{i-1},p,\eta}[f(y_i^{i+m} \mid x_i^{i+m'}, \beta, \sigma^2, p)] \\ &= \mathbf{E}_{\beta,\sigma^2\sim\Pi|z^{i-1},p,\eta}[f(y_i^{i+m} \mid x_i^{i+m}, \beta, \sigma^2, p)]\end{aligned} \tag{3.11}$$

(writing a_i^j as shorthand for $a_i, \ldots, a_j$). The previous displays held for general priors. The experiments in this chapter adopt widely used priors (see e.g. Raftery et al., 1997): normal priors on the β's and inverse gamma priors on the variance. These conjugate priors allow explicit analytical formulas for all relevant quantities for arbitrary η, provided below. We only consider the simple case of a fixed $\mathcal{M}_p$ here; the more complicated formulas with an additional prior on p will be given in Appendix 4.A.1 in the next chapter.

Fixed p and σ^2 Let $\mathbf{X}_n = (x_1^\top, \ldots, x_n^\top)^\top$ be the design matrix. For a linear model $\mathcal{M}_p$ with fixed variance σ^2 and initial Gaussian prior on β given by $N(\bar{\beta}_0, \sigma^2\Sigma_0)$, the generalized posterior on β is again Gaussian with mean

$$\bar{\beta}_{n,\eta} := \mathbf{E}_{\beta\sim\Pi|z^n,p,\eta}\, \beta = \Sigma_{n,\eta}(\Sigma_0^{-1}\bar{\beta}_0 + \eta\mathbf{X}_n^\top y^n) \tag{3.12}$$

and covariance matrix $\sigma^2\Sigma_{n,\eta}$, where $\Sigma_{n,\eta} = (\Sigma_0^{-1} + \eta \mathbf{X}_n^\top \mathbf{X}_n)^{-1}$.

Fixed p, varying σ^2 Now consider linear models with a Gaussian prior on β conditional on σ^2 as above, and a conjugate (inverse gamma) prior on σ^2, i.e. $\pi(\sigma^2) = \text{Inv-gamma}(\sigma^2 \mid a_0, b_0)$ for some a_0 and b_0. Here we use the following parameterization of the inverse gamma distribution:

$$\text{Inv-gamma}(\sigma^2 \mid a, b) = \sigma^{-2(a+1)} e^{-b/\sigma^2} b^a / \Gamma(a). \tag{3.13}$$

The posterior $\pi(\sigma^2, z^n, p)$ is then given by $\text{Inv-gamma}(\sigma^2 \mid a_{n,\eta}, b_{n,\eta})$ where

$$a_{n,\eta} = a_0 + \eta n/2 \ ; \ b_{n,\eta} = b_0 + \frac{\eta}{2} \sum_{i=1}^{n} (y_i - x_i \bar{\beta}_{n,\eta})^2. \tag{3.14}$$

The posterior expectation of σ^2 can be calculated as

$$\bar{\sigma}^2_{n,\eta} := \frac{b_{n,\eta}}{a_{n,\eta} - 1}. \tag{3.15}$$

Note that the posterior mean of β given σ^2 does not depend on σ^2.

3.4 The SafeBayesian algorithm

3.4.1 Introducing SafeBayes via the prequential view

We introduce SafeBayes via Dawid's prequential interpretation of Bayes factor model selection. As was first noticed by Dawid (1984) and Rissanen (1984), we can think of Bayes factor model selection as picking the model with index p that, when used for sequential prediction with a logarithmic scoring rule, minimizes the cumulative loss. To see this, note that for any distribution whatsoever, we have that, by definition of conditional probability,

$$-\log f(y^n) = -\log \prod_{i=1}^{n} f(y_i \mid y^{i-1}) = \sum_{i=1}^{n} -\log f(y_i \mid y^{i-1}).$$

In particular, for the standard Bayesian marginal distribution $\bar{f}(\cdot \mid p) = \bar{f}(\cdot \mid p, \eta = 1)$ as defined above, for each fixed p, we have

$$-\log \bar{f}(y^n \mid x^n, p) = \sum_{i=1}^{n} -\log \bar{f}(y_i \mid x^n, y^{i-1}, p) = \sum_{i=1}^{n} -\log \bar{f}(y_i \mid x_i, z^{i-1}, p), \tag{3.16}$$

where the second equality holds by (3.11). If we assume a uniform prior on model index p, then Bayes factor model selection picks the model maximizing $\pi(p \mid z^n)$, which by Bayes' theorem coincides with the model minimizing (3.16), i.e. minimizing cumulative log-loss. Similarly, in 'empirical Bayes' approaches, one picks the value of some nuisance parameter ρ that maximizes

the marginal Bayesian probability $\bar{f}(y^n \mid x^n, \rho)$ of the data. By (3.16), which still holds with p replaced by ρ, this is again equivalent to the ρ minimizing the cumulative log-loss. This is the *prequential* interpretation of Bayes factor model selection and empirical Bayes approaches, showing that Bayesian inference can be interpreted as a sort of *forward* (rather than cross-) validation (Dawid, 1984; Rissanen, 1984; Hjorth, 1982).

We will now see whether we can use this approach with ρ in the role of the η for the η-generalized posterior that we want to learn from the data. We continue to rewrite (3.16) as follows (with ρ instead of p that can either stand for a continuous-valued parameter or for a model index but not yet for η), using the fact that the Bayes predictive distribution given ρ and z^{i-1} can be rewritten as a posterior-weighted average of f_θ:

$$\begin{aligned}\check{\rho} &:= \arg\max_\rho \bar{f}(y^n \mid x^n, \rho) = \arg\min_\rho \sum_{i=1}^n \left(-\log \bar{f}(y_i \mid x_i, z^{i-1}, \rho)\right) \\ &= \arg\min_\rho \sum_{i=1}^n \left(-\log \mathbf{E}_{\theta\sim\Pi|z^{i-1},\rho}[f(y_i \mid x_i, \theta)]\right). \end{aligned} \tag{3.17}$$

This choice for $\check{\rho}$ being entirely consistent with the Bayesian approach, our first idea is to choose $\hat{\eta}$ in the same way: we simply pick the η achieving (3.17), with ρ substituted by η. However as Figure 4.5 will show (the blue line there depicts (3.17) for one of our experiments), this will tend to pick η close to 1 and does not improve predictions under misspecification. Indeed, we introduced η to deal with the case in which the Bayesian model assumptions are violated, so we cannot expect that learning it in a Bayes-like way such as (3.17) will resolve the issue. But it turns out that a *slight* modification of (3.17) does the trick: we simply interchange the order of logarithm and expectation in (3.17) and pick the η minimizing

$$\sum_{i=1}^n \mathbf{E}_{\theta\sim\Pi|z^{i-1},\eta}\left[-\log f(y_i \mid x_i, \theta)\right]. \tag{3.18}$$

In words, we pick the η minimizing the *P*osterior-*E*xpected *P*osterior-*R*andomized log-loss, i.e. the log-loss we expect to obtain, according to the η-generalized posterior, if we actually sample from this posterior. This modified loss function has also been called *Gibbs error* (Cuong et al., 2013), and while the abbreviation *PEPR*-log-loss would be more correct, we simply call it the η-*R*-log-loss from now on.

A detailed explanation of why this works will be given in Sections 4.1.3 and 4.2; for now we just notice that by Jensen's inequality, for any fixed η, for every sequence of data we must have

$$\mathbf{E}_{\theta\sim\Pi|z^{i-1},\eta}\left[-\log f(y_i \mid x_i, \theta)\right] \geq -\log \mathbf{E}_{\theta\sim\Pi|z^{i-1},\eta}\left[f(y_i \mid x_i, \theta)\right], \tag{3.19}$$

yet, the difference between both sides is small if the posterior is *concentrated* for (x_i, y_i), i.e. for small ϵ and small positive δ, it puts $1-\delta$ of its mass on distributions which assign the same density to y_i given x_i up to a factor $1+\epsilon$ — clearly,

if $\delta = \epsilon = 0$ then both sides are the same. Thus, at values for η at which the generalized posterior is 'cumulatively concentrated', i.e. concentrated at most sample points, the objective function will be similar to the standard Bayesian one. This is the clue to further analysis of the algorithm to follow later.

In practice, it is computationally infeasible to try all values of η and we simply have to try out a number of values. For convenience we give a detailed description of the resulting algorithm below, copied from Grünwald (2012). In this chapter, we will invariably apply it with $z_i = (x_i, y_i)$ as before, and $\ell_\theta(z_i)$ set to the (conditional) log-loss as defined before, although it sometimes also has a second interpretation with ℓ_θ as square-loss.

Algorithm 3.1: The (R-)SafeBayesian algorithm

Input: data $z_1, \ldots, z_n$, model $\mathcal{M} = \{f(\cdot \mid \theta) \mid \theta \in \Theta\}$, prior Π on Θ, step-size κ_{STEP}, max. exponent $\kappa_{\max}$, loss function $\ell_\theta(z)$

Output: Learning rate $\hat{\eta}$

$\mathcal{S}_n := \{1, 2^{-\kappa_{\text{STEP}}}, 2^{-2\kappa_{\text{STEP}}}, 2^{-3\kappa_{\text{STEP}}}, \ldots, 2^{-\kappa_{\max}}\}$;

for all $\eta \in \mathcal{S}_n$ **do**

  $s_\eta := 0$;

  for $i = 1 \ldots n$ **do**

    Determine generalized posterior $\Pi(\cdot \mid z^{i-1}, \eta)$ of Bayes with learning rate η.

    Calculate "posterior-expected posterior-randomized loss" of predicting actual next outcome:

$$r := \ell_{\Pi|z^{i-1},\eta}(z_i) = \mathbf{E}_{\theta\sim\Pi|z^{i-1},\eta}\left[\ell_\theta(z_i)\right] \tag{3.20}$$

    $s_\eta := s_\eta + r$;

  end

end

Choose $\hat{\eta} := \arg\min_{\eta\in\mathcal{S}_n}\{s_\eta\}$ (if min achieved for several $\eta \in \mathcal{S}_n$, pick largest);

Variation As we will see in Section 4.1.4, the crucial property to make inference about η work is that the expression inside the sum in (3.17) is replaced by

$$\mathbf{E}_{\theta\sim\Pi'}[-\log f_\theta(Y_i \mid X_i)], \tag{3.21}$$

where Π' should be chosen such that the resulting log-loss is as small as possible. In (3.18) we set $\Pi' = \Pi$, but Π' is allowed to be *any* distribution on θ under which the expected log-loss is small. The heuristic analysis of Section 4.1.4 suggests that the smaller the loss that can be formed this way (see also the open problems, Section 4.3.3), the better the resulting method is expected to work.

Now the *η-in-model-log-loss* (or just *η-I-log-loss*), defined as

$$\sum_{i=1}^{n} \left[-\log f(y_i \mid x_i, \mathbf{E}_{\theta \sim \Pi | z^{i-1}, \eta}[\theta]) \right], \tag{3.22}$$

is (by Jensen's inequality) always smaller than (3.18) for the linear models that we consider. This means that, instead of finding the η minimizing (3.18), we may want to find the η minimizing (3.22), which is of the form (3.21) with Π' equal to a point mass on $\bar{\theta}_{i,\eta} := \mathbf{E}_{\theta \sim \Pi | z^{i-1}, \eta} f_\theta$. We call the version of SafeBayes which minimizes the alternative objective function (3.22) *in-model SafeBayes*, abbreviated to *I*-SafeBayes, and from now on use *R*-SafeBayes for the original version based on the *R*-log-loss. We did not realize the potential benefits of using in-model SafeBayes at the time of writing Grünwald (2012), and while the theoretical results of Grünwald can be adjusted to deal with such modifications, we cannot get any better theoretical convergence bounds as yet, but this may be an artefact of our proof techniques. A secondary goal of the experiments in this chapter is thus to see whether one can really improve SafeBayes by using the 'in-model' version.

3.4.2 Instantiating SafeBayes to the linear model

Our experiments concern four instantiations of SafeBayes: *R*-SafeBayes and *I*-SafeBayes for models with fixed variance, denoted *R-square-SafeBayes and I-square-SafeBayes* for reasons that will become clear below, are the topic of experiments in Section 5.1.1. The main text instead investigates, in Section 3.5, *R*-SafeBayes and *I*-SafeBayes for models with varying variance, denoted *R-log-SafeBayes and I-log-SafeBayes*. Below we give explicit formulas for each when conditioned on a fixed model $\mathcal{M}_p$; the case with a posterior on p itself can easily be derived from these.

Fixed σ^2: *R*-square- and *I*-square-SafeBayes When conditioned on a fixed p and σ^2 (a situation with which we experiment in Section 5.1.1.2), SafeBayes tries to minimize the *R*-log-loss, which, as an easy calculation shows, is just the sum, from $i = 0$ to $n-1$, of

$$\begin{aligned} &\mathbf{E}_{\beta \sim \Pi | z^i, p, \eta} \left[-\log f(y_{i+1} \mid x_{i+1}, \beta, \sigma^2) \right] \\ &\qquad = \frac{1}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} (y_{i+1} - x_{i+1} \bar{\beta}_{i,\eta})^2 + \frac{1}{2} x_{i+1} \Sigma_{i,\eta} x_{i+1}^\top, \end{aligned} \tag{3.23}$$

where $\bar{\beta}_{i,\eta}$ and $\Sigma_{i,\eta}$ are given as in and below (3.12). Note that $\bar{\beta}_{i,\eta}$ depends on η but not on σ, and note also that, since $\mathbf{X}_n^\top \mathbf{X}_n$ (as in (3.12)) tends to increase linearly in n and p, the final term is of order $p/(n\eta)$.

In the corresponding in-model version of SafeBayes, we use the in-model-loss as given by $-\log f(y_{i+1} \mid x_{i+1}, \bar{\beta}_{i,\eta}, \sigma^2)$, which is equal to (3.23) without the final term. Since the first term of (3.23) does not depend on the data, this version of SafeBayes thus amounts to picking the $\hat{\eta}$ minimizing just the sum

of square-loss prediction errors, *which does not depend on the chosen* σ^2. It thus becomes a standard version of 'prequential model selection' as based on the square-loss, which in turn is similar to (though having different asymptotics than) leave-one-out cross validation based on the square-loss.

Indeed, the fixed σ^2 versions of SafeBayes can be interpreted in two ways: first, as we did until now, in terms of SafeBayes with ℓ_θ in (3.20) set to the log-loss, i.e. as a tool for dealing with misspecification; and second, with ℓ_θ in (3.20) set proportionally to the square-loss, as a generic tool to learn good square-loss predictors (not distributions) in a pseudo-Bayesian way. More precisely, I-SafeBayes with the log-loss for fixed σ^2 is equivalent to the version of I-SafeBayes we would get if we set $\ell_{\beta,\sigma^2}(x,y) := C(y - x\beta)^2$, for any constant $C > 0$. Similarly, R-SafeBayes with the log-loss for fixed σ^2 is equivalent to the version of R-SafeBayes we would get if we set $\ell_{\beta,\sigma^2}(x,y) := C(y - x\beta)^2$, although now equivalence only holds if we set $C = 1/2\sigma^2$. For this reason we will now refer to them as *I-square-SafeBayes and R-square-SafeBayes*, respectively.

Varying σ^2: R-log- and I-log-SafeBayes Next consider the situation with fixed p and varying σ^2, with posterior on σ^2 an inverse gamma distribution with parameters $a_{n,\eta}$ and $b_{n,\eta}$ as given by (3.14). Then the R-log-loss is given by

$$\begin{aligned}
&\mathbf{E}_{\sigma^2,\beta\sim\Pi|z^i,p,\eta}\left[-\log f(y_{i+1} \mid x_{i+1},\beta,\sigma^2)\right] \\
&\quad= \frac{1}{2}\log 2\pi b_{i,\eta} - \frac{1}{2}\psi(a_{i,\eta}) + \frac{1}{2}\frac{(y_{i+1} - x_{i+1}\bar{\beta}_{i,\eta})^2}{b_{i,\eta}/a_{i,\eta}} + \frac{1}{2}x_{i+1}\Sigma_{i,\eta}x_{i+1}^\top \\
&\quad= \frac{1}{2}\log 2\pi\bar{\sigma}^2_{i,\eta} + \frac{1}{2}\frac{(y_{i+1} - x_{i+1}\bar{\beta}_{i,\eta})^2}{\bar{\sigma}^2_{i,\eta}} + \frac{1}{2}x_{i+1}\Sigma_{i,\eta}x_{i+1}^\top + r(i,\eta), \quad (3.24)
\end{aligned}$$

where ψ is the digamma function, $\bar{\sigma}^2_{i,\eta}$ is the η-posterior expectation of σ^2 as given by (3.15) and $r(i,\eta)$ is a remainder function which is $O(1/i)$ whenever $\sum_{i=1}^n (y_i - x_i\beta_{n,\eta})^2$ increases linearly in i. This final approximation follows by (3.15) and because we have $\psi(x) \in [\log(x-1), \log x]$. R-SafeBayes for varying σ^2 minimizes (3.24), and, because there is now only a log-loss and not a direct square-loss interpretation, we will call it *R-log-SafeBayes* from now on.

To calculate the corresponding in-model version of SafeBayes, *I-log-SafeBayes*, note that it minimizes the sum of

$$-\log f(y_{i+1} \mid x_{i+1},\bar{\beta}_{i,\eta},\bar{\sigma}^2_{i,\eta}) = \frac{1}{2}\log 2\pi\bar{\sigma}^2_{i,\eta} + \frac{1}{2}\frac{(y_{i+1} - x_{i+1}\beta_{i,\eta})^2}{\bar{\sigma}^2_{i,\eta}}. \quad (3.25)$$

Comparing the four versions of SafeBayes, we see that both R-SafeBayeses have an additional term which decreases in η, increases in model dimensionality p (via the size of the matrix $\Sigma_{i,\eta}$), but becomes negligible for $n \gg p$.

3.4.3 SafeBayes learns to predict as well as the optimal distribution

We first define the *Cesàro-averaged* posterior given data Z^n by setting, for any subset $\Theta' \subset \Theta$,

$$\Pi_{\text{Ces}}(\Theta' \mid Z^n, \eta) := \frac{1}{n} \sum_{i=1}^{n} \Pi(\Theta' \mid Z^i, \eta) \tag{3.26}$$

to be the posterior probability of Θ' averaged over the n posterior distributions obtained so far. Predicting based on Cesàro-averaged posteriors was introduced independently by several authors (Barron, 1987; Helmbold and Warmuth, 1992; Yang, 2000; Catoni, 1997) and has received a lot of attention in the machine learning literature in recent years, also under the name "on-line to batch conversion of Bayes" or *progressive mixture rule* (Audibert, 2007) or *mirror averaging* (Juditsky et al., 2008; Dalalyan and Tsybakov, 2012), but is of course unnatural from a Bayesian perspective.

The main result of Grünwald (2012) essentially states the following: suppose that, under P^*, the density ratios are uniformly bounded, i.e. there is a finite v such that for all $\theta, \theta' \in \Theta$, $P^*(f_\theta(Y \mid X)/f_{\theta'}(Y \mid X) \leq v) = 1$. Suppose further that the prior Π assigns 'sufficient mass' in KL-neighbourhoods of $P_{\tilde{\theta}}$. Then Π_{Ces} applied with the $\hat{\eta}$ learned by the SafeBayesian algorithm concentrates on the optimal $P_{\tilde{\theta}}$. That is, let Θ_δ be the subset of all $\theta \in \Theta$ with $D(P^* \| P_\theta) \geq D(P^* \| P_{\tilde{\theta}}) + \delta$. Then for all $\delta > 0$, with P^*-probability 1, as $n \to \infty$, we have that $\Pi_{\text{Ces}}(\Theta_\delta \mid Z^n, \hat{\eta})$ goes to 0. Grünwald goes on to show that in several settings, one can design priors such that the rate at which the posterior concentrates is minimax optimal, i.e. no algorithm can do better in general. On the negative side, the requirement of bounded density ratio is strong, and the replacement of the standard posterior by the Cesàro one is awkward. On the positive side, the theorem has no further conditions and can be applied to parametric and nonparametric cases alike.

In recent, as yet unpublished work, Grünwald (2014) extends the result to deal with unbounded density ratios as in the regression setting considered here, and to the 'standard' η-generalized rather than the Cesàro-averaged η-generalized posterior. In both cases, convergence can still be proved but the bounds given on the concentration rate worsen by a $\log n$ factor. We suspect that in many situations, this is an artefact of the proof technique, and to see whether there is any practical difference, below we include experimental results both for the Cesàro-averaged η-generalized posterior $\Pi_{\text{Ces}}(\cdot \mid Z^n, \hat{\eta})$ and for the standard η-generalized posterior $\Pi(\cdot \mid Z^n, \hat{\eta})$.

3.5 Main experiment: Varying σ^2

In this section we provide our main experimental results, based on linear models $\mathcal{M}_p$ as defined in Section 3.2.2 with a prior on both the mean and the variance. Figures 3.3–3.6 depict, and Section 3.5.3 discusses the results of model selection and averaging experiments, which choose or average between the

models $0, \ldots, p_{\max}$, where we consider first an incorrectly and then a correctly specified model, both with $p_{\max} = 50$ and later with $p_{\max} = 100$. Section 3.5.4 contains and interprets additional experiments on Bayesian ridge regression, with a fixed p; a multitude of additional experiments is provided in Chapter 5. Section 3.5.5 at the end of this chapter summarizes the relevant findings of these additional experiments.

3.5.1 Preparing the main experiments: Model, priors, method, 'truth'

In this subsection we prepare the experiments: Section 3.5.1.1 describes our priors π; Section 3.5.1.2 concerns the sampling ('true') distributions P^* with which we experiment; and finally, Section 3.5.2 describes the data statistics that we will report.

3.5.1.1 The priors

Prior on models In our model selection/averaging experiments, we use a fat-tailed prior on the models given by

$$\pi(p) \propto \frac{1}{(p+2)(\log(p+2))^2}.$$

This prior was chosen because it remains well-defined for an infinite collection of models, even though we only use finitely many in our experiments.

Variation As a sanity check we did repeat some of our experiments with a uniform prior on $0, \ldots, p_{\max}$ instead; the results were indistinguishable.

Prior on parameters given models Each model $\mathcal{M}_p$ has parameters β, σ^2, on which we put the standard conjugate priors as described in Section 3.3.1. We set the mean of the prior on β to $\bar{\beta}_0 = \mathbf{0}$, and its covariance matrix to $\sigma^2\Sigma_0$. Our main experiments below are based on an *informative* instantiation of Σ_0, using the identity matrix $\Sigma_0 = \mathbf{I}_{p+1}$; this prior equals the posterior we would get by starting with an improper Jeffreys' prior on β and then observing, for each coefficient β_j, one extra point $z = (x, 0)$ with $x_j = 1$ and $x_i = 0$ for $i \neq j$.

Variations We also ran experiments with a 'slightly informative' Σ_0, where we set $\Sigma_0 = 1000 \cdot \mathbf{I}_{p+1}$, comparable to observing points $z = (x, 0)$ with $x_j = 1/\sqrt{1000}$. Finally, following the standard reference Raftery et al. (1997), we also used a prior with a level of informativeness depending on the submodel, described in more detail in Section 5.1.

As to the prior on σ^2: Jeffreys' prior is obtained for the choice $a_0 = b_0 = 0$ in (3.13). We do not use this improper prior, because of the well-known issues with Bayes factors under improper priors (O'Hagan, 1995). Moreover, to calculate the posterior's reliability (defined in Section 3.5.2 and shown in Figure 3.3) and also for the I-log-loss, we need to calculate the posterior expectation of the variance σ^2 quantity as given by (3.15), which is only well-defined and finite for $a_n > 1$. We want to make $\pi(\sigma^2)$ as uninformative as possible while ensuring

that (for any positive learning rate) this variance exists for the posterior based on at least one sample. This is accomplished by choosing $a_0 = 1$: for standard Bayes, the posterior after one observation has $a_1 = a_0 + 1/2$; for generalized Bayes, $a_1 = a_0 + \eta/2$. To set b_0, we use that b_0/a_0 represents the sample variance of a virtual initial data sequence (Gelman et al., 2013, Section 14.8). We choose $b_0 = 1/40$ so that $b_0/a_0 = 1/40$, the true variance of the noise in our data, as we describe next.

3.5.1.2 The "truth" (sampling distribution)

Our experiments fall into two categories: correct-model and wrong-model experiments.

Correct-model experiments Here $X_1, X_2, \ldots$ are sampled i.i.d., with, for each individual $X_i = (X_{i1}, \ldots, X_{ip_{\max}})$, $X_{i1} \ldots, X_{ip_{\max}}$ i.i.d. $\sim N(0,1)$. Given each X_i, Y_i is generated as

$$Y_i = .1 \cdot (X_{i1} + \ldots + X_{i4}) + \epsilon_i, \tag{3.27}$$

where the ϵ_i are i.i.d. $\sim N(0, \sigma^{*2})$ with variance $\sigma^{*2} = 1/40$.

Wrong-model experiments Now at each time point i, a fair coin is tossed independently of everything else. If the coin lands heads, then the point is 'easy', and $(X_i, Y_i) := (\mathbf{0}, 0)$. If the coin lands tails, then X_i is generated as for the correct model, and Y_i is generated as (3.27), but now the noise random variables have variance $\sigma_0^2 = 2\sigma^{*2} = 1/20$. Thus, $Z_i = (X_i, Y_i)$ is generated as in the true model case but with a larger variance; this larger variance has been chosen so that the marginal variance of each Y_i is the same value σ^{*2} in both experiments.

From the results in Section 3.2.3 we immediately see that, for both experiments, the optimal model is $\mathcal{M}_{\tilde{p}}$ for $\tilde{p} = 4$, and the optimal distribution in $\mathcal{M}$ and $\mathcal{M}_{\tilde{p}}$ is parameterized by $\tilde{\theta} = (\tilde{p}, \tilde{\beta}, \tilde{\sigma}^2)$ with $\tilde{p} = 4$, $\tilde{\beta} = (\tilde{\beta}_0, \ldots \tilde{\beta}_4) = (0, .1, .1, .1, .1), \tilde{\sigma}^2 = 1/40$ (in the correct model experiment, $\tilde{\sigma}^2 = \sigma^{*2}$; in the wrong model experiment, since $\tilde{\sigma}^2$ must be reliable, it must be equal to the square-risk obtained with $(\tilde{p}, \tilde{\beta})$, which is $(1/2) \cdot (1/20) = 1/40$). $f(x) := x\tilde{\beta}$ is then equal to the *true* regression function $\mathbf{E}_{P^*}[Y \mid X]$.

Variations We have already seen a variation of these two experiments depicted in Figures 3.1 and 3.2. In the correct-model version of that experiment, P^* is defined as follows: set $X_j = P_j(S)$, where P_j is the Legendre polynomial of degree j and S is uniformly distributed on $[-1, 1]$, and set $Y = 0 + \epsilon$, where $\epsilon \sim N(0, \sigma^{*2})$, with $\sigma^{*2} = 1/40$; $(X_1, Y_1), \ldots$ are then sampled as i.i.d. copies of (X, Y). Note that the true regression function is 0 here. In Section 5.3 we briefly consider this and several other variations of these ground truths.

3.5.2 The statistics we report

Figure 3.3 reports the results of the wrong-model, $p = 50$ experiment; Figure 3.4 shows correct-model, $p = 50$; Figure 3.5 is about wrong-model, $p =$

100; and Figure 3.6 depicts the correct-model, $p = 100$ setting. For all four experiments we measure three aspects of the performance of Bayes and SafeBayes, each summarized in a separate graph. First, we show the behaviour of several prediction methods based on SafeBayes relative to square-risk; second, we measure whether the methods provide a good assessment of their own predictive capabilities in terms of square-loss, i.e. whether they are reliable and not 'overconfident'. Third, we check a form of model identification consistency. Below we explain these three performance measures in detail. We postpone all experiments with log-loss rather than square-loss to Section 4.1.4. We also provide a fourth graph in each case indicating what $\hat{\eta}$'s are typically selected by the two versions of SafeBayes.

Square-risk For a given distribution W on (p, β, σ^2), the *regression function based on* W, a function mapping covariate X to $\mathbf{R}$, abbreviated to $\mathbf{E}_W[Y \mid X]$, is defined as

$$\mathbf{E}_W[Y \mid X] := \mathbf{E}_{(p,\beta,\sigma)\sim W} \mathbf{E}_{Y\sim P_{p,\beta,\sigma}|X}[Y] = \mathbf{E}_{(p,\beta,\sigma)\sim W}\left[\sum_{j=0}^{p} \beta_j X_j\right]. \tag{3.28}$$

If we take W to be the η-generalized posterior, then (3.28) is also simply called the η-posterior regression function. The *square-risk* relative to P^* based on predicting by W is then defined as an extension of (3.3) as

$$\text{RISK}^{\text{sq}}(W) := \mathbf{E}_{(X,Y)\sim P^*}(Y - \mathbf{E}_W[Y \mid X])^2. \tag{3.29}$$

In the experiments below we measure the square-risk relative to P^* at sample size $i-1$ achieved by, respectively, (1), the η-generalized posterior, (2), the η-generalized posterior conditioned on the MAP (maximum a posteriori) model, and, (3), the η-generalized Cesàro-averaged posteriors, i.e.

$$\begin{aligned} &\mathbf{E}_{Z^{i-1}\sim P^*}[\text{RISK}^{\text{sq}}(W)], \text{ with} \\ &W = \Pi \mid Z^{i-1}, \eta \; ; \; W = \Pi \mid Z^{i-1}, \eta, \breve{p}_{\text{map}(Z^{i-1},\eta)} \; ; \; W = \Pi_{\text{CES}} \mid Z^{i-1}, \eta, \end{aligned} \tag{3.30}$$

respectively, where the MAP model $\breve{p}_{\text{map}(Z^{i-1},\eta)}$ is defined as the p achieving $\max_{p\in 0,\ldots,p_{\max}} \pi(p \mid Z^{i-1}, \eta)$, with $\pi(p \mid Z^{i-1}, \eta)$ defined as in (3.10), and Π_{CES} is the Cesàro-averaged posterior as defined as in (3.26). We do this for three values of η: (a) $\eta = 1$, corresponding to the standard Bayesian posterior, (b) $\eta := \hat{\eta}(Z^{i-1})$ set by the R-log SafeBayesian algorithm run on the past data Z^{i-1}, and (c) η set by the I-log SafeBayesian algorithm. In the figures of Section 3.5.3, 1(a) is abbreviated to *Bayes*, 1(b) is *R-log-SafeBayes*, 1(c) is *I-log-SafeBayes*, 2(a) is *Bayes MAP*, 2(b) is *R-log-SafeBayes MAP*, 2(c) is *I-log-SafeBayes MAP*, and results with Cesàro-averaging are discussed but not explicitly shown. In Section 3.5.4, additionally 3(a) is *Bayes Cesàro*, 3(b) is *R-log-SafeBayes Cesàro*, and 3(c) is *I-log-SafeBayes Cesàro*.

Concerning the three square-risks that we record: The first choice is the most natural, corresponding to the prediction (regression function) according to the 'standard' η-generalized posterior; the second corresponds to the

situation where one first selects a single submodel $\check{p}_{\text{map}}$ and then bases all predictions on that model; it has been included because such methods are often adopted in practice. The third choice, the *Cesàro-averaged generalized posterior* is included because, when $\eta = \hat{\eta}$ is set by SafeBayes, this is the choice that Grünwald (2012) provides theoretical convergence results for (as we discussed, Grünwald (2014) provides results for the non-averaged η-generalized posterior as well, but these are worse by a log-factor). But we are also interested in the results for the Cesàro-average for $\eta = 1$, because this has been proposed earlier — albeit somewhat implicitly and with different models — to stabilize Bayesian predictions in adversarial circumstances (Helmbold and Warmuth, 1992), so we include these as well.

In Figure 3.3 and subsequent figures below, we depict these quantities by sequentially sampling data $Z_1, Z_2, \ldots, Z_{\text{max}}$ i.i.d. from a P^* as defined above in Section 3.5.1.2, where max is some large number. At each i, after the first $i-1$ points Z^{i-1} have been sampled, we compute the three square-risks given above. We repeat the whole procedure a number of times (called 'runs'); the graphs show the average risks over these runs.

MAP-model identification / Occam's razor When the goal of inference is model identification, 'consistency' of a method is often defined as its ability to identify the smallest model $\mathcal{M}_{\tilde{p}}$ containing the 'pseudo-truth' $(\tilde{\beta}, \tilde{\sigma}^2)$. To see whether standard Bayes and/or SafeBayes are consistent in this sense, we check whether the MAP model $\check{p}_{\text{map}(Z^{i-1},\eta)}$ is equal to $\tilde{p}$.

Reliability vs. overconfidence Does Bayes learn how good it is in terms of squared error? To answer this question, we define, for a predictive distribution W as in (3.29) above, $U_i^{[W]}$ (a function of X_i, Y_i and (through W) of Z^{i-1}), as

$$U_i^{[W]} = (Y_i - \mathbf{E}_W[Y_i \mid X_i])^2.$$

This is the error we make if we predict Y_i using the regression function based on prediction method W. In the graphs in the next sections we plot the *self-confidence ratio* $\mathbf{E}_{X_i,Y_i\sim P^*}[U_i^{[W]}] / \mathbf{E}_{X_i\sim P^*}\mathbf{E}_{Y_i\sim W|X_i}[U_i^{[W]}]$ as a function of i for the three prediction methods / choices of W defined above. We may think of this as the ratio between the actual expected prediction error (measured in square-loss) one gets by using a predictor who based predictions on W and the marginal (averaged over X) subjectively expected prediction error by this predictor. We previously, in Section 3.2.3, showed that the KL-optimal $(\tilde{p}, \tilde{\beta}, \tilde{\sigma}^2)$ is *reliable*: this means that, if we would take W the point mass on $(\tilde{p}, \tilde{\beta}, \tilde{\sigma}^2)$ and thus, irrespective of past data Z^{i-1}, would predict by $\mathbf{E}_{(\tilde{p},\tilde{\beta},\tilde{\sigma}^2)}[Y_i \mid X_i] = \sum_{j=0}^{\tilde{p}} \tilde{\beta}_j X_{ij}$, then the ratio would be 1. For the W learned from data considered above, a value larger than 1 indicates that W does not implement a 'reliable' method in the sense of Section 3.2.3, but rather overconfident: it predicts its predictions to be better than they actually are, in terms of square-risk.

3.5.3 Main model selection/averaging experiment

We run the SafeBayesian algorithm of Section 3.4 with $z_i = (x_i, y_i)$ and $\ell_\theta(z_i) = -\log f_\theta(y_i \mid x_i)$ is the (conditional) log-loss as described in that section. As to the parameters of the algorithm (page 52), in all experiments we set the step-size $\kappa_{\text{STEP}} = 1/3$ and $\kappa_{\max} := 8$, i.e. we tried the following values of η: $1, 2^{-1/3}, 2^{-2/3}, \ldots, 2^{-8}$. The result of the wrong-model and correct-model experiment as described above with $p_{\max} = 50$ and $p_{\max} = 100$, respectively, are given in Figures 3.3–3.6.

Conclusion 1: Bayes performs well in model-correct, and dismally in model-incorrect experiment The four figures show that standard Bayes behaves excellently in terms of all quality measures (square-risk, MAP model identification and reliability) when the model is correct, and dismally if the model is incorrect.

Conclusion 2: If (and only if) model incorrect, then the higher $p_{\max}$, the worse Bayes gets We see from Figures 3.4 and 3.6 that standard Bayes behaves excellently in terms of all quality measures (square-risk, MAP model identification and reliability) when the model is correct, both if $p_{\max} = 50$ and if $p_{\max} = 100$, the behaviour at $p_{\max} = 100$ being essentially indistinguishable from the case with $p_{\max} = 50$. These and other (unreported) experiments strongly suggests that, when the data are sampled from a low-dimensional model, then, when the model is correct, standard Bayes is unaffected (does not get confused) by adding additional high-dimensional models to the model space. Indeed, the same is suggested by various existing Bayesian consistency theorems, such as those by Doob (1949); Ghosal et al. (2000); Zhang (2006a).

At the same time, from Figures 3.3 and 3.5 we infer that standard Bayes behaves very badly in all three quality measures in our (admittedly very 'evilly chosen') model-wrong experiment. At very large sample sizes, Bayes eventually recovers, but the main point here to notice is that the n at which a given level of recovery (measured in, say, square-loss) takes place is much higher for the case $p_{\max} = 100$ (Figure 3.5) than for the case $p_{\max} = 50$ (Figure 3.3). This strongly suggests that, when the model is incorrect but the best approximation lies in a low-dimensional submodel, then standard Bayes gets confused by adding additional high-dimensional models to the model space — recovery takes place at a sample size that increases with $p_{\max}$. Indeed, the graphs strongly suggest that in the case that $p_{\max} = \infty$ (with which we cannot experiment), Bayes will be inconsistent in the sense that the risk of the posterior predictive will never ever reach the risk attainable with the best submodel. Grünwald and Langford (2007) showed that this can indeed happen with a simple, but much more unnatural classification model; the present result indicates (but does not prove) that it can happen with our standard model as well.

Conclusion 3: R-log-SafeBayes and I-log-SafeBayes generally perform well Comparing the four graphs for SafeBayes and I-log-SafeBayes, we see that

they behave quite well for *both* the model-correct and the model-wrong experiments, being slightly worse than, though still competitive to, standard Bayes when the model is correct and incomparably better when the model is wrong. Indeed, in the wrong-model experiments, about half of the data points are identical and therefore do not provide very much information, so one would expect that if a 'good' method achieves a given level of square-risk at sample size n in the correct-model experiment, it achieves the same level at about $2n$ in the incorrect-model experiment, and this is indeed what happens. Also, we see from comparing Figures 3.5 and 3.6 on the one hand to Figures 3.3 and 3.4 on the other that adding additional high-dimensional models to the model space hardly affects the results — like standard Bayes when the model is correct, SafeBayes does not get confused by the additional, larger model space.

Secondary conclusions We see that both types of SafeBayes converge quickly to the right (pseudo-true) model order, which is pleasing since they were not specifically designed to achieve this. Whether this is an artefact of our setting or holds more generally would, of course, require further experimentation. We note that at small sample sizes, when both types of SafeBayes still tend to select an overly simple model, I-log-SafeBayes has significantly more variability in the model chosen-on-average; it is not clear though whether this is 'good' or 'bad'. We also note that the η's chosen by both versions are very similar for all but the smallest sample sizes, and are consistently smaller than 1. When instead of the full η-generalized posteriors, the η-generalized posterior conditioned on the MAP $\check{p}_{\text{map}}$ is used, the behaviour of all method consistently deteriorates a little, but never by much.

For lack of space in the graphs, we did not show the Cesàro-versions of Bayes, R-log-SafeBayes and I-log-SafeBayes (methods 3(a), 3(b), 3(c) in Section 3.5.2). Briefly, the curves look as follows: Cesàro-Bayes performs significantly better than standard Bayes in all three quality measures in the wrong-model experiments, but is still far from competitive with the two (full-posterior) SafeBayes versions. When Cesàroified, the SafeBayes methods become a bit smoother but not necessarily better. Very similar behaviour of Cesàro (making bad methods significantly better but still not competitive, and good methods smoother, sometimes a bit worse and sometimes a bit better) has been explicitly depicted in the ridge regression with varying σ^2 in Section 3.5.4 below.

3.5.4 Second experiment: Ridge regression, varying σ^2

We repeat the model-wrong and model-correct experiments of Figures 3.3 and 3.4, with just one major difference: all posteriors are conditioned on $p := p_{\max} = 50$. Thus, we effectively consider just a fixed, high-dimensional model, whereas the best approximation $\tilde{\theta} = (50, \tilde{\beta}, \tilde{\sigma}^2)$ viewed as an element of $\mathcal{M}_p$ is 'sparse' in that it has only $\beta_1, \ldots, \beta_4$ not equal to 0. We note that the MAP model index graphs of Figures 3.3 and 3.4 are meaningless in this context (they

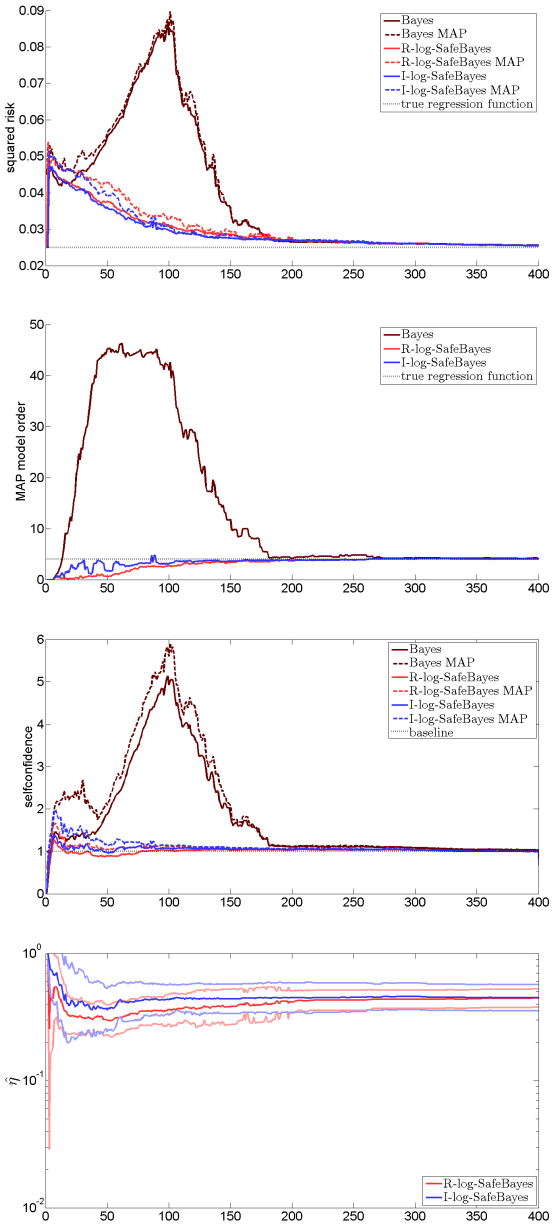


Figure 3.3: Four graphs showing respectively the square-risk, MAP model order, overconfidence (lack of reliability), and selected $\hat{\eta}$ at each sample size, each averaged over 30 runs, for the wrong-model experiment with $p_{\max} = 50$, for the methods indicated in Section 3.5.2. For the selected-$\hat{\eta}$ graph, the pale lines are one standard deviation apart from the average; all lines in this graph were computed over $\hat{\eta}$ indices (so that the lines depict the geometric mean over the values of $\hat{\eta}$ themselves).

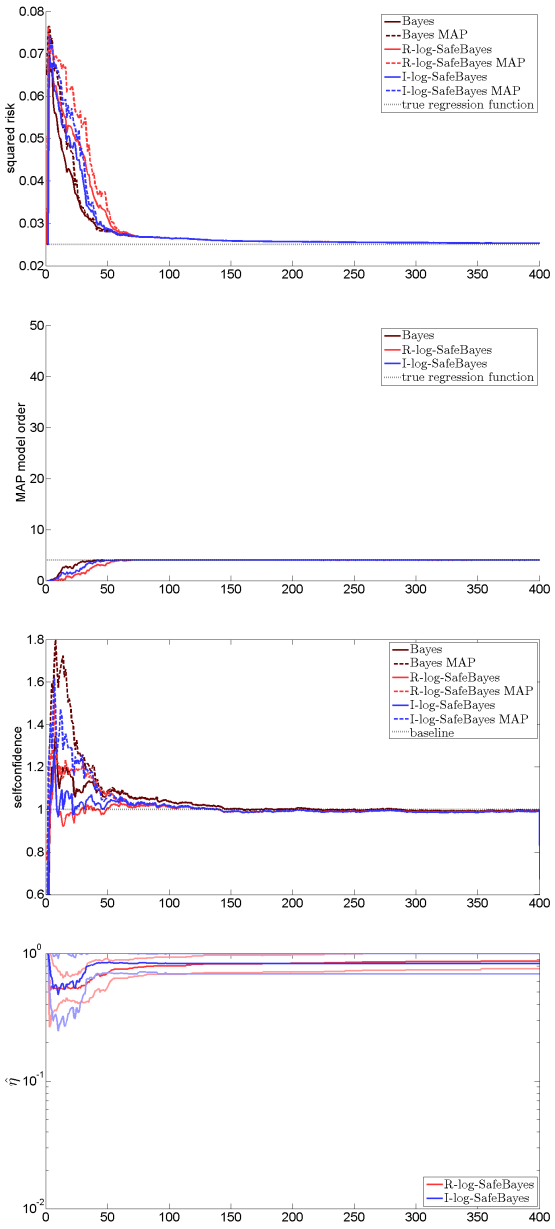


Figure 3.4: Same graphs as in Figure 3.3 for the correct-model experiment with $p_{\max} = 50$

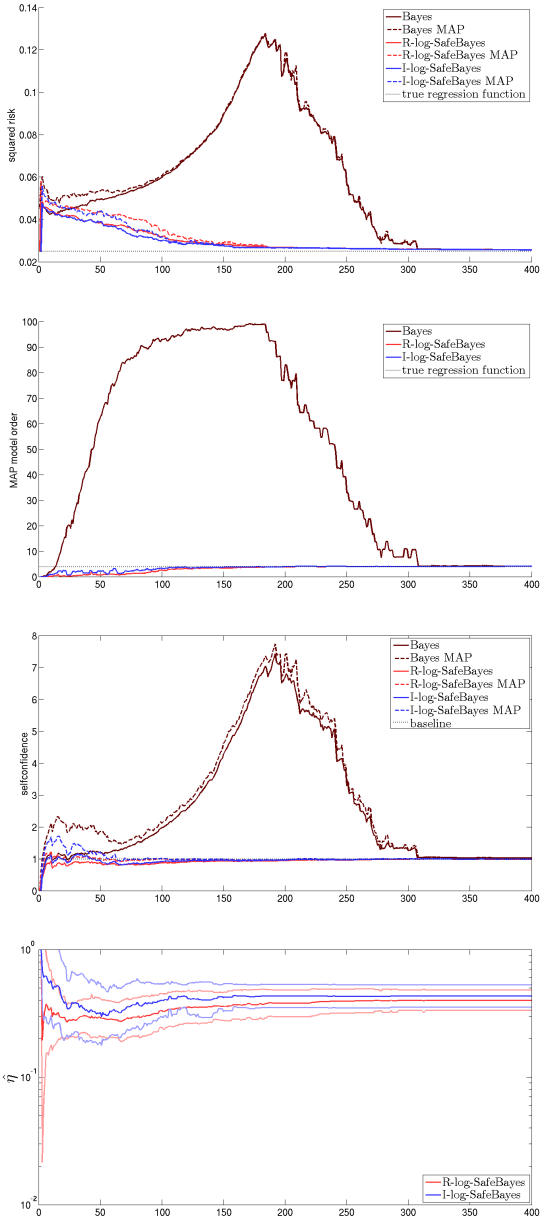


Figure 3.5: Same four graphs as in Figure 3.3, for the wrong-model experiment with $p_{\max} = 100$

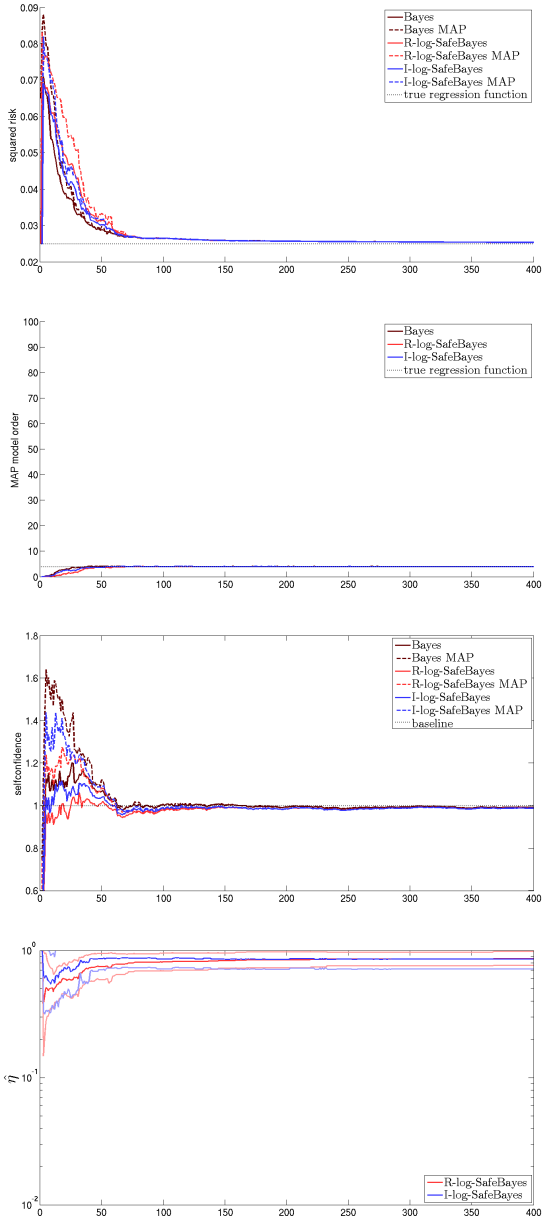


Figure 3.6: Same graphs as in Figure 3.3 for the correct-model experiment with $p_{\max} = 100$

would be equal to the constant 50) so they are left out of the new Figures 3.7 and 3.8.

Instantiating SafeBayes Since we noticed in preliminary experiments that some versions of SafeBayes now have a tendency to select much smaller values of η than in the previous experiments, we now set $\kappa_{\max} = 16$ (large enough so that in no experiment the optimal $\eta < 2^{-\kappa_{\max}}$); for computational reasons we also increased the step size and set $\kappa_{\text{STEP}} = 1$.

Connection to Bayesian (b)ridge regression From (3.12) we see that the posterior mean parameter $\bar{\beta}_{i,\eta}$ is equal to the posterior MAP parameter and depends on η but not on σ^2, since σ^2 enters the prior in the same way as the likelihood. Therefore, the square-loss obtained when using the generalized posterior for prediction is always given by $(y_i - x_i\bar{\beta}_{i,\eta})^2$ irrespective of whether we use the posterior mean, or MAP, or the value of σ^2. Interestingly, if we fix some λ and perform standard (nongeneralized) Bayes with a modified prior, proportional to the original prior raised to the power $\lambda := \eta^{-1}$, then the prior becomes normal $N(\bar{\beta}_0, \sigma^2\Sigma_0')$ where $\Sigma_0' = \eta\Sigma_0$ and the standard posterior given z^i is then (by (3.12)) Gaussian with mean

$$\left(\left(\Sigma_0'\right)^{-1} + \mathbf{X}_n^\top \mathbf{X}\right)^{-1} \cdot \left(\left(\Sigma_0'\right)^{-1} \bar{\beta}_0 + \mathbf{X}_n^\top y^n\right) = \bar{\beta}_{i,\eta}. \tag{3.31}$$

Thus we see that in this special case, the (square-risk of the) η-generalized Bayes posterior mean coincides with the (square-risk of the) standard Bayes posterior mean with prior $N(\bar{\beta}_0, \sigma^2\eta\Sigma_0)$. But this means that the square-loss obtained by η-generalized Bayes on a data sequence is precisely equal to the square-loss obtained by *Bayesian ridge regression* with penalty parameter $\lambda = \eta^{-1}$, as defined, by, e.g., Park and Casella (2008) (to be precise, they call this method Bayesian 'bridge' regression with $q = 2$; the choice of $q = 1$ in their formula gives their celebrated 'Bayesian Lasso'). It is thus of interest to see what happens if η (equivalently, λ) is determined by *empirical Bayes*, which is one of the methods Park and Casella (2008) suggest. In addition to the graphs discussed earlier in Section 3.5.2, we thus also show the results for η set in this alternative way. Whereas this empirical-Bayesian ridge regression is usually a very competitive method (indeed in our model-correct experiment, Figure 3.8, it performs best in all respects), we will see in Figure 3.7 (the green line) that, just like other versions of Bayes, it breaks down under our type of misspecification.

We hasten to add that the correspondence between the η-generalized posterior means and the standard posterior means with prior raised to power $1/\eta$ only holds for the $\bar{\beta}_{i,\eta}$ parameters. It does not hold for the $\bar{\sigma}^2_{i,\eta}$ parameters, and thus, for fixed η, the self-confidence ratio of both methods may be quite different.

Conclusions for model-wrong experiment For most curves, the overall picture of Figure 3.7 is comparable to the corresponding model averaging experi-

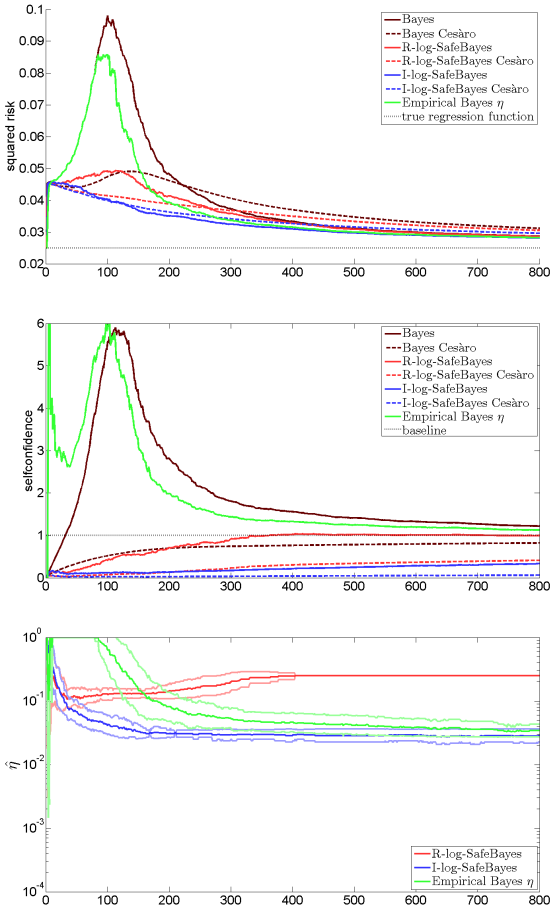


Figure 3.7: Bayesian ridge regression: Model-wrong experiment conditioned on $p := p_{\max} = 50$. The graphs (square-risk, self-confidence ratio and chosen η as function of sample size) are as in Figures 3.3–3.6, except for the third graph there (MAP model order), which has no meaning here. The meaning of the curves is given in Section 3.5.2 except for *empirical Bayes*, explained in Section 3.5.4.

ment, Figure 3.3: when the model is wrong, standard Bayes shows dismal performance in terms of risk and reliability up to a certain sample size and then very slowly recovers, whereas both versions of SafeBayes perform quite well even for small sample sizes. We do not show variations of the graph for $p = p_{\max} = 100$ (i.e. the analogue of Figure 3.5), since it relates to Figure 3.7 in exactly the same way as Figure 3.5 relates to Figure 3.3: with $p = 100$, bad square-risk and reliability behaviour of Bayes goes on for much longer (recovery takes place at much larger sample size) and remains equally good as for

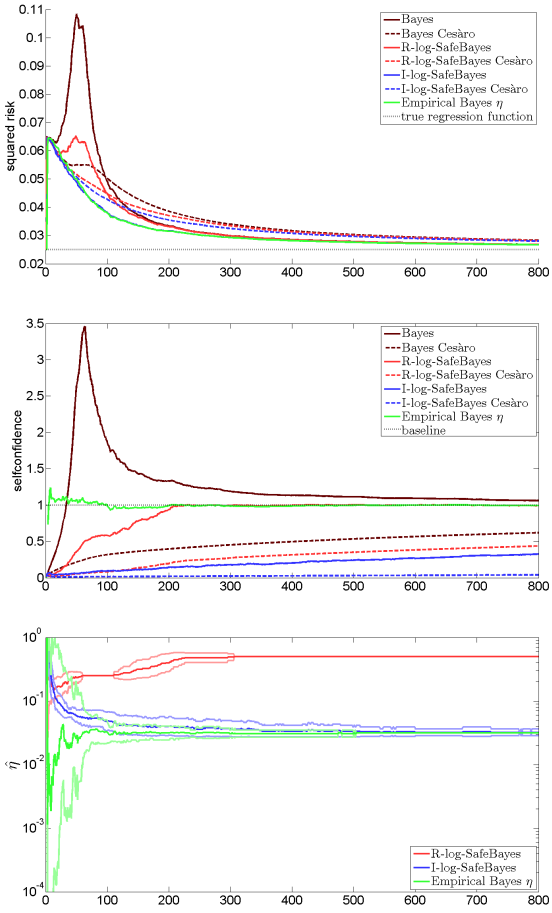


Figure 3.8: Bayesian ridge regression: Same graphs as in Figure 3.7, but for the model-correct experiment conditioned on $p := p_{\max} = 50$.

$p = 50$ with the two versions of SafeBayes.

The results for the Cesàro-versions of our methods are exactly as discussed at the end of Section 3.5.3.

We also see that, as we already indicated in the introduction, choosing the learning rate by empirical Bayes (thus implementing one version of Bayesian bridge regression) behaves terribly. This complies with our general theme that, to 'save Bayes' in general misspecification problems, the parameter η cannot be chosen in a standard Bayesian manner.

Conclusions for model-correct experiment The model-correct experiment for ridge regression (Figure 3.8) offers a surprise: we had expected Bayes to perform best, and were surprised to find that the SafeBayeses obtained smaller

risk. Some followup experiments (not shown here), with different true regression functions and different priors, shed more light on the situation. Consider the setting in which the coefficients of the true function are drawn randomly according to the prior. In this setting standard Bayes performs at least as good in expectation as any other method including SafeBayes (the Bayesian posterior now represents exactly what an experimenter might ideally know). SafeBayes (still in this setting) usually chooses $\eta = 1/2$ or $1/4$, and the difference in risks compared to Bayes is small. On the other hand, if the true coefficients are drawn from a distribution with substantially smaller variance than a priori expected by the prior (a factor 1000 in the 'correct'-model experiment of Figure 3.8), then SafeBayes performs much better than Bayes. Here Bayes can no longer necessarily be expected to have the best performance (the model is correct, but the prior is "wrong"), and it is possible that a slightly reduced learning rate gives (significantly) better results. It seems that this situation, where the variance of the true function is much smaller than its prior expectation, is not exceptional: for example, Raftery et al. (1997) suggest choosing the variance of the prior in such a way that a large region of parameter values receives substantial prior mass. Following that suggestion in our experiments already gives a variance that is large enough compared to the true coefficients that SafeBayes performs better than Bayes even if the model is correct.

A joint observation for the model-wrong and model-correct experiments Finally we note that we see an interesting difference between the two SafeBayes versions here: *I*-log-SafeBayes seems better for risk, giving a smooth decreasing curve in both experiments. *R*-log-SafeBayes inherits a trace of standard Bayes' bad behaviour in both experiments, with a nonmonotonicity in the learning curve. On the other hand, in terms of reliability, *R*-log-SafeBayes is consistently better than *I*-log-SafeBayes (but note that the latter is underconfident, which is arguably preferable over being overconfident, as Bayes is). All in all, there is no clear winner between the two methods.

3.5.5 Executive summary: Joint conclusions from main and additional experiments

Standard Bayes In almost all our experiments (both here and in Chapter 5), standard Bayesian inference fails in its KL-associated prediction tasks (squared error risk, reliability) when the model is wrong. Adopting a different prior (such as the *g*-prior) does not help, with two exceptions in model averaging: (a) when Raftery's prior (Section 5.1.3) is used, then Bayes works quite well, but there it fails dramatically again (in contrast to SafeBayes) once the percentage of easy points is increased; (b) when it is run with a fixed variance that is significantly larger than the 'best' (pseudo-true) variance $\tilde{\sigma}^2$. Moreover, in the ridge regression experiment with fixed σ^2, we find that standard Bayes can even perform much worse than SafeBayes when the model is correct — so all in all we tentatively conclude that SafeBayes is safer to use for linear regression.

SafeBayes R-square-SafeBayes is not competitive with the other SafeBayes methods and can even get worse than Bayes sometimes; this is due to an unwanted dependence on the specified scale σ^2 as explained in Section 5.1. The other three SafeBayes methods behave reasonably well in all our experiments, and there is no clear winner among them. I-square-SafeBayes usually behaves excellently for the square-risk, but cannot directly be used to assess its own performance. I-log-SafeBayes usually behaves excellently in terms of square-risk as well but is underconfident about its own performance (which is perhaps acceptable, overconfidence being a lot more dangerous). R-log-SafeBayes is usually good in terms of square-risk though not as good as I-log-SafeBayes, yet it is highly reliable. However, in Section 5.2.1, we describe an initial idea for discounting the importance of the first few outcomes and explain why this might improve performance. When combined with this discounting idea, R-log-SafeBayes may actually always be competitive with the other two methods in terms of square-risk as well.

Learning η in Bayes- or likelihood way fails Despite its intuitive appeal, fitting η to the data by e.g. empirical Bayes fails both in the model-wrong ridge experiment with a prior on σ^2, where it amounts to Bayesian ridge regression (Figure 3.7) and in the model-wrong fixed-variance ridge experiment (where it amounts to a method for learning the variance, see Section 5.1.1.2).

Robustness of experiments It does not matter whether the $X_{i1}, X_{i2}, \ldots$ are independent Gaussian, uniform or represent polynomial basis functions: all phenomena reported here persist for all choices. If the 'easy' points are not precisely $(0, 0)$, but have themselves a small variance in both dimensions, then all phenomena reported here persist, but on a smaller scale.

Centring We repeated several of our experiments with centred data, i.e. preprocessed data so that the empirical average of the Y_i is exactly 0 (Raftery et al., 1997; Hastie et al., 2001). In none of our experiments did this affect any results. We also looked at the case where the true regression function has an intercept far from 0, and data are *not* centred. This hardly affected the SafeBayes methods.

Other methods We also repeated the wrong-model experiment for several other model selection methods: AIC, BIC, and various forms of cross-validation. Briefly, we found that all these have severe problems with our data as well. experiments, the mentioned methods were used to identify a model index p and η played no role, but in our final experiment we used leave-one-out cross-validation to learn η itself. With the squared error loss it worked fine, which is not too surprising given its close similarity to I-square-SafeBayes. However, when we tried it with log-loss (as a likelihoodist or information-theorist might be tempted to do), it behaved terribly.

Chapter 4

Bayesian Inconsistency: Explanations and Discussion

In this chapter, we give several explanations of how the Bayesian inconsistency seen in Chapter 4 may occur under 'bad' misspecification, and why SafeBayes provides a solution to this problem. We also discuss how our inconsistency example and the SafeBayes method relate to other work.

4.1 Bayes' behaviour explained

In this section we explain how anomalous behaviour of the Bayesian posterior may arise, taking a frequentist perspective. Section 4.1.1 is merely provided to give some initial intuition and may be skipped. The proof of Theorem 4.1 is given in Appendix 4.A.2.

4.1.1 Explanation I: Variance issues

Example 4.A. [Bernoulli] Consider the following very simple scenario: our 'model' consists of two Bernoulli distributions, $\mathcal{M} = \{P_\theta \mid \theta \in \{0.2, 0.8\}\}$, with P_θ expressing that $Y_1, Y_2, \ldots \sim$ i.i.d. $\textsc{Ber}(\theta)$. We perform Bayesian inference based on a uniform prior on $\mathcal{M}$. Suppose first that the data are, in fact, sampled i.i.d. from P_{θ^*}, where θ^* is the 'true' parameter. The model is misspecified, in particular we will take a $\theta^* \notin \{0.2, 0.8\}$. The log-likelihood ratio between the two distributions for data Y^n with n_1 ones and $n_0 = n - n_1$ zeroes, measured for convenience in bits (base 2), is given by

$$L = \log_2 \frac{f_{0.8}(Y^n)}{f_{0.2}(Y^n)} = \log_2 \frac{(0.8)^{n_1}(0.2)^{n_0}}{(0.2)^{n_1}(0.8)^{n_0}} = 2(n_1 - n_0). \tag{4.1}$$

With uniform priors, the posterior will prefer $\theta = 0.2$ as soon as $L < 0$.

First suppose $\theta^* = 1/2$. Then both distributions in $\mathcal{M}$ are equally far from θ^* in terms of KL divergence (or any other commonly used measure). By the

central limit theorem, however, we expect that the probability that $|L| > \sqrt{n}/2$ is larger than a constant for all large n; in this particular case we numerically find that, for all n, it is larger than 0.32.

This implies that, at each n, $\min_{\theta \in \{0.2,0.8\}} \pi(\theta \mid Y^n) \approx 2^{-\sqrt{n}/2}$ with 'true' probability at least 0.32. Thus, there is a nonnegligible 'true' probability that the posterior on one of the two distributions is negligibly small, and a naive Bayesian who adopted such a model would be strongly convinced that the other distribution would be better even though both distributions are equally bad. While this already indicates that strange things may happen under misspecification, we are of course more interested in the situation in which $\theta^* \neq 1/2$, so that one of the two distributions in $\mathcal{M}$ is truly 'better'. Now, if the 'true' parameter θ^* is within $O(1/\sqrt{n})$ of $1/2$, then, by the central limit theorem, the probability that $L < 0$ is nonnegligible. For example, if θ^* is exactly $1/2 + 1/\sqrt{n}$, then this probability is larger than 0.16 for all n. Thus, for values of θ^* this close to $1/2$, there is no way we can even expect Bayes to learn the 'best' value. For fixed (independent of n), larger values of θ^*, like 0.6, the posterior will concentrate at 0.8 at an exponential rate, but the sample size at which concentration starts is considerably larger than the sample sized needed when the true parameter is in fact 0.8. For example, at $n = 50$, $P_{0.6}(L < 0) \approx 0.1$, $P_{0.8}(L < 0) \approx 2 \cdot 10^{-5}$; both probabilities go to 0 exponentially fast but their ratio increases exponentially with n. So, under a fixed θ^*, with increasing n, Bayes may take longer to concentrate on the best $\tilde{\theta}$ if $\tilde{\theta} \neq \theta^*$ (misspecification) than if $\tilde{\theta} = \theta^*$, but it eventually 'recovers' (this was seen in the ridge experiments of Section 3.5.4). Now, for larger models, the consequence of slower concentration of the log-likelihood ratio L is that the probability that *some* 'bad' P_θ happens to 'win' is substantially larger than with a correct model. Grünwald and Langford (2007) showed that, in a classification context with an infinite-dimensional model, there are so many of such 'bad' P_θ that Bayes does not recover any more, and the posterior keeps putting most of its mass on a bad model for ever (although the particular bad model on which it puts its mass keeps changing). In Chapter 3 we empirically showed the same in a regression problem.

Now one might conjecture that the issues above are caused by the fact that the model $\mathcal{M}$ is 'disconnected'. In the Bernoulli example above, the problem indeed goes away if instead of the model $\mathcal{M}$, we adopt its 'closure' $\mathcal{M}' = \{P_\theta \mid \theta \in [0.2, 0.8]\}$. However, high-dimensional regression problems exhibit the same phenomenon, even if their parameter spaces are connected. It turns out that in general, to get concentration at the same rates as if the model were correct, the model must be *convex*, i.e. closed under taking any finite mixture of the densities, which is a much stronger requirement than mere connectedness. For standard Gaussian regression problems with $Y \mid X \sim N(0, \sigma^2)$, this would mean that we would have to adopt a model in which $Y \mid X$ can be any Gaussian mixture with arbitrarily many components — which is clearly not practical (note that 'convex' refers to the space of densities, not the space of regression functions (Grünwald and Langford, 2007, Section 6.3.5)).

4.1.2 Explanation II: Good vs. bad misspecification

Barron (1998) showed that sequential Bayesian prediction under a logarithmic score function shows excellent behaviour in a cumulative risk sense; for a related result see (Barron et al., 1999, Lemma 4). Although Barron (1998) focuses on the well-specified case, this assumption is not required for the proof and the result still holds even if the model $\mathcal{M}$ is completely wrong. For a precise description and proof of this result emphasizing that it holds under misspecification, see (Grünwald, 2007, Section 15.2). At first sight, this leads to a paradox, as we now explain.

A paradox? Let $\tilde{\theta}$ index the KL-optimal distribution in Θ as in Section 3.2.1. The result of Barron (1998) essentially says that, for arbitrary models Θ, for all n,

$$\mathbf{E}_{Z^n \sim P^*}\left[\sum_{i=1}^{n} \textsc{risk}^{\log}(\Pi \mid Z^{i-1}) - \textsc{risk}^{\log}(\tilde{\theta})\right] \leq \textsc{red}_n, \tag{4.2}$$

where $\textsc{risk}^{\log}(W)$, for a distribution W on Θ, is defined as the log-risk obtained when predicting by the W-mixture of f_θ, i.e.

$$\textsc{risk}^{\log}(W) = \mathbf{E}_{X,Y \sim P^*}[-\log \mathbf{E}_{\theta \sim W} f_\theta(Y \mid X)]. \tag{4.3}$$

In (4.2), this coincides with log-risk of the Bayes predictive density $\bar{f}(\cdot \mid Z^{i-1})$, as defined by (3.8). Here, as in the remainder of this section, we look at the standard Bayes predictive density, i.e. $\eta = 1$. $\textsc{red}_n$ is the so-called *relative expected stochastic complexity* or *redundancy* (Grünwald, 2007), which depends on the prior and for 'reasonable' priors is typically *small*. The result thus means that, when sequentially predicting using the standard predictive distribution under a log-scoring rule, one does not lose much compared to when predicting with the log-risk optimal $\tilde{\theta}$.

When $\mathcal{M}$ is a union of a finite or countably infinite number of parametric exponential families and $\tilde{p} < \infty$ is well-defined, then, under some further regularity conditions, which hold in the regression example of Chapter 3 (Grünwald, 2007), the redundancy is, up to $O(1)$, equal to the BIC term $(\tilde{k}/2) \log n$, where $\tilde{k}$ is the dimensionality of the smallest model containing $\tilde{\theta}$. In the regression case, $\mathcal{M}_{\tilde{p}}$ has $\tilde{p} + 2$ parameters $(\beta_0, \ldots, \beta_p, \sigma^2)$, so in the two experiments of Section 3.5, $\tilde{k} = 6$. Thus, in our regression example, when sequentially predicting with the standard Bayes predictive $\bar{f}(\cdot \mid Z^{i-1})$, the cumulative log-risk is at most $n \cdot \textsc{risk}^{\log}(\tilde{\theta})$ which is linear in n, plus a logarithmic term that becomes comparatively negligible as n increases. This is confirmed by Figure 4.2 on page 77. Now, for each individual $\theta = (p, \beta, \sigma^2)$ we know from Section 3.2.3 that, if its log-risk is close to that of $\tilde{\theta}$, then its square-risk must also be close to that of $\tilde{\theta}$; and $\tilde{\theta}$ itself has the smallest square-risk among all $\theta \in \Theta$. Hence, one would expect the reasoning for log-risk to transfer to square-risk: it seems that when sequentially predicting with the standard Bayes predictive $\bar{f}(\cdot \mid Z^{i-1})$, the cumulative square-risk should at most be n times the instantaneous square-risk of $\tilde{\theta}$ plus a term that hardly grows with n; in other words, the cumulative

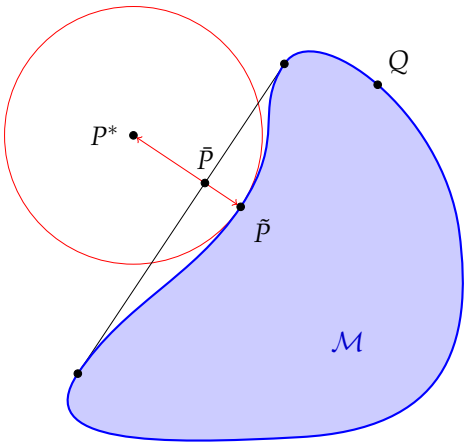


Figure 4.1: Benign vs. bad misspecification: $\tilde{P} = \arg\min_{P \in \mathcal{M}} D(P^* \| P)$ is the distribution in model $\mathcal{M}$ that minimizes KL divergence to the 'true' P^*, but, since the model is nonconvex, the Bayes predictive distribution $\bar{P}$ may happen to be very different from any $P \in \mathcal{M}$. When this happens, we can have 'bad misspecification' and then it may be necessary to decrease the learning rate (in this simplistic drawing $\bar{P}$ is a mixture of just two distributions; in our regression example it mixes infinitely many). Yet if P^* were such that $\inf_{P \in \mathcal{M}} D(P^* \| P)$ does not decrease if the infimum is taken over the convex hull of $\mathcal{M}$ (e.g. if Q rather than $\tilde{P}$ reached the minimum), then any learning rate $\eta < 1$ is fine ('benign' misspecification). In the picture, we even have $D(P^* \| \bar{P}) < D(P^* \| \tilde{P})$; in this case we can get hypercompression.

square-risk from time 1 to n, averaged over time by dividing by n, should rapidly converge to the constant instantaneous risk of $\tilde{\theta}$. Yet the experiments of Section 3.5 clearly show that this is *not* the case: Figure 3.3 shows that, until $n = 100$, it is about 3 times as large.

This 'paradox' is resolved once we realize that the Bayesian predictive density $\bar{f}(\cdot \mid^{i-1})$ is a *mixture* of various f_θ, and not necessarily similar to f_θ for any individual θ — the link between log-risk and square-risk (3.4) only holds for individual $\theta = (p, \beta, \sigma^2)$, not for mixtures of them. Indeed, if at each point in time i, $\bar{f}(\cdot \mid Z^i)$ would be very similar (in terms of e.g. Hellinger distance) to some particular f_{θ_i} with $\theta_i \in \Theta$, then there would really be a contradiction. Thus, the discrepancy between the good log-risk and bad square-risk results in fact *implies* that at a substantial fraction of sample sizes i, $\bar{f}(\cdot \mid Z^i)$ must be substantially different from *every* $\theta \in \Theta$. In other words, *the posterior is not concentrated at such i*. A cartoon picture of this situation is given in Figure 4.1: the Bayes predictive achieves small log-risk because it mixes together several distributions into a single predictive distribution which is very different from any particular single $f_\theta \in \mathcal{M}$. By Barron's bound, (4.2), the resulting $\bar{f}(\cdot \mid Z^i)$ must, averaged over i, have at most a risk almost as small as the risk of $\tilde{\theta}$. We can thus, at least informally, distinguish between "benign" and "bad" misspecifi-

cation. Bad misspecification occurs if there is a nonnegligible probability that for a range of sample sizes, the predictive distribution is substantially different from any of the distributions in $\mathcal{M}$. As Figure 4.1 suggests, 'bad' misspecification cannot occur for convex models $\mathcal{M}$ — and indeed, the results by Li (1999) suggest that for such models consistency holds under weak conditions for any $\eta < 1$, even under misspecification.

4.1.3 Hypercompression

The picture suggests that, if, as in our regression model, the model is nonconvex (i.e. the set of densities $\{f_\theta \mid \theta \in \Theta\}$ is not closed under taking mixtures), then $\bar{f}(\cdot \mid Z^i)$ might in fact be significantly *better* in terms of log-risk than the best $\tilde{\theta}$, and its individual constituents might even all be substantially worse than $\tilde{\theta}$. If this were indeed the case then, with high P^*-probability, we would also get the analogous result for an actual sample (and not just in expectation): the cumulative log-risk obtained by the Bayes predictive should be significantly smaller than the cumulative log-risk achieved with the optimal $\tilde{f}$. Figure 4.2 below shows that this indeed happens with our data, until $n \approx 100$.

The no-hypercompression inequality In fact, Figure 4.2 shows a phenomenon that is virtually impossible if the Bayesian's model and prior are 'correct' in the sense that data Z^n would behave like a typical sample from them: it easily follows from Markov's inequality (for details see Grünwald, 2007, Chapter 3) that, letting Π denote the Bayesian's joint distribution on $\Theta \times \mathcal{Z}^n$, for each $K \geq 0$,

$$\Pi\Big\{(\theta, Z^n) \,:\, \sum_{i=1}^{n} \Big(-\log \bar{f}(Y_i \mid X_i, Z^{i-1})\Big) \\ \leq \sum_{i=1}^{n} \Big(-\log f_\theta(Y_i \mid X_i, Z^{i-1})\Big) - K\Big\} \leq e^{-K},$$

i.e. the probability that the Bayes predictive $\bar{f}$ cumulatively outperforms f_θ, with θ drawn from the prior, by K log-loss units is exponentially small in K. Figure 4.2 below thus shows that at sample size $n \approx 90$, an a priori formulated event has happened of probability less than e^{-30}, clearly indicating that something about our model or prior is quite wrong.

Since the difference in cumulative log-loss between $\bar{f}$ and f_θ can be interpreted as the amount of bits saved when coding the data with a code that would be optimal under $\bar{f}$ rather than f_θ, this result has been called the *no-hypercompression inequality* by Grünwald (2007). The figure shows that for our data, we have substantial hypercompression.

The SafeBayes error measure As seen from (3.18), SafeBayes measures the performance of η-generalized Bayes not by the cumulative log-loss, as standard Bayes does, but instead by the cumulative posterior-expected error when

predicting by drawing from the posterior. One way to interpret this alternative error measure is that, at least in expectation, we cannot get hypercompression. Defining (compare to (4.3)!)

$$\text{RISK}^{\text{R-log}}(W) = \mathbf{E}_{X,Y\sim P^*}\, \mathbf{E}_{\theta\sim W}[-\log f_\theta(Y \mid X)], \tag{4.4}$$

we get by Fubini's theorem,

$$\begin{aligned}&\text{RISK}^{\text{R-log}}(W) - \text{RISK}^{\log}(\tilde{\theta}) \\ &\quad = \mathbf{E}_{\theta\sim W}\, \mathbf{E}_{X,Y\sim P^*}[[-\log f_\theta(Y \mid X)] - [-\log f_{\tilde{\theta}}(Y \mid X)]] \geq 0, \end{aligned} \tag{4.5}$$

where the inequality follows by definition of $\tilde{\theta}$ being log-risk optimal among Θ. There is thus a crucial difference between $\text{RISK}^{\text{R-log}}$ and $\text{RISK}^{\log}$ — for the latter we just argued that, under misspecification, $\text{RISK}^{\log}(W) - \text{RISK}^{\log}(\tilde{\theta}) \leq 0$ is very well possible. Thus, in contrast to predicting with the mixture density $\mathbf{E}_{\theta\sim W}\, f_\theta$, prediction by randomization (first sampling $\theta \sim W$ and then predicting with the sampled f_θ) cannot 'exploit' the fact that mixture densities might have smaller log-risk than their components. Thus, if the difference (4.5) is small, then W must put most of its mass on distributions $\theta \in \Theta$ that have small log-risk themselves. For *individual* θ, we know that small log-risk implies small square-risk. This implies that if (4.5) is small, then the (standard) posterior is concentrated on distributions with small square-risk.

Experimental demonstration of hypercompression for standard Bayes Figure 4.2 and Figure 4.3 show the predictive capabilities of standard Bayes in the wrong model example in terms of *cumulative* and *instantaneous log-loss* on a simulated sample. The graphs clearly demonstrate hypercompression: the Bayes predictive cumulatively performs *better* than the best single model / the best distribution in the model space, until at about $n \approx 100$ there is a phase transition. At individual points, we see that it sometimes performs a little worse, and sometimes (namely at the 'easy' $(0, 0)$ points) much better than the best distribution. We also see that, as anticipated above, randomized and in-model Bayesian prediction do *not* show hypercompression and in fact perform terribly on the log-loss until the phase transition at $n = 100$, when they become almost as good as standard Bayes. We see that for $\eta = 1$, they perform much worse. An important conclusion is that *if we are only interested in log-loss prediction, it is clear that we just want to use Bayes rather than randomized or in-model prediction with different* η.

> As an aside, we note that the first few outcomes have a dramatic effect on cumulative R- and I-log-loss (it disappears from Figure 4.2); this may be due to the fact that our densities — other than those considered by Grünwald (2012) — have unbounded support so that there is no v such that Theorem 4.1 below holds. This observation inspired the idea described in Section 5.2.1 about ignoring the first few outcomes when determining the optimal η. Also, we emphasize that the hypercompression phenomenon takes places more generally, not just in our regression setup — for example, the classification inconsistency noted by Grünwald and Langford (2007) can be understood in terms of hypercompression as well.

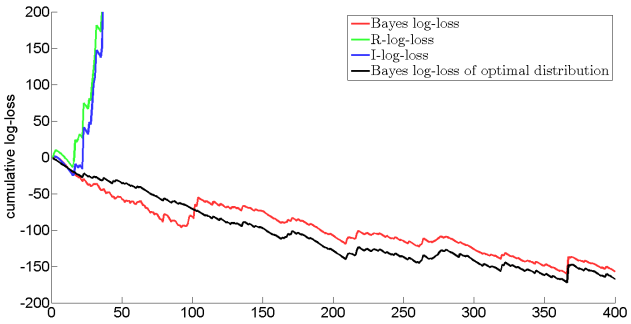


Figure 4.2: Cumulative standard, R-, and I-log-loss as defined in (3.18) and (3.22) respectively of standard Bayesian prediction ($\eta = 1$) on a single run for the model-averaging experiment of Figure 3.3. We clearly see that standard Bayes achieves *hypercompression*, being better than the best single model. And, as predicted by theory, randomized Bayes is never better than standard Bayes, whose curve has negative slope because the densities of outcomes are > 1 on average.

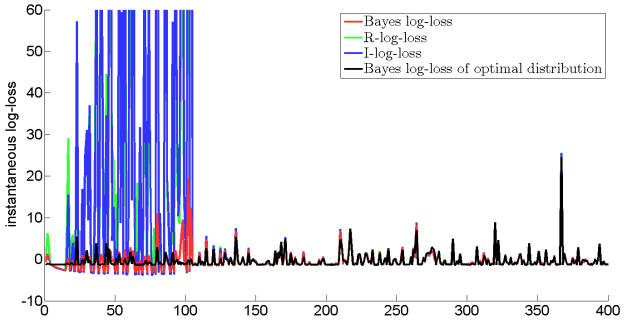


Figure 4.3: Instantaneous standard, R- and I-log-loss of standard Bayesian prediction for the run depicted in Figure 4.2.

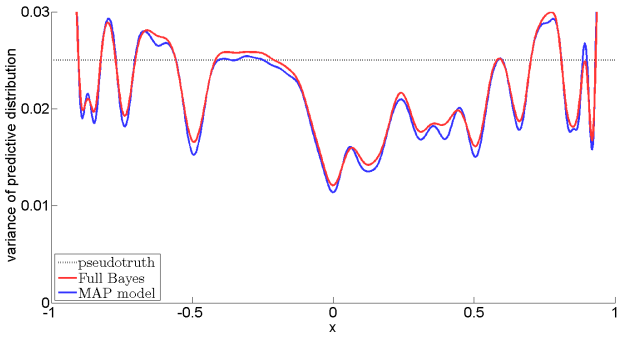


Figure 4.4: Variance of standard Bayes predictive distribution conditioned on a new input S as a function of S after 50 examples for the polynomial model-wrong experiment (Figure 3.1), shown both for the predictive distribution based on the full, model-averaging posterior and for the posterior conditioned on the MAP model $\mathcal{M}_{\check{p}_{\text{map}}}$. For both posteriors, the posterior mean of Y is incorrect for $x \neq 0$, yet $\bar{f}(Y \mid Z^{50}, X)$ still achieves small risk because of its small variance at $X = 0$.

How hypercompression arises in regression Figure 4.4 gives some clues as to how hypercompression is achieved: it shows the variance of the predictive distribution $\bar{f}(\cdot \mid Z^{50})$ as a function of $S \in [-1, 1]$ for the polynomial example of Figure 3.1 in the introduction, at sample size $n = 50$, where hypercompression takes place. Figure 3.1 gave the posterior mean (regression function) at $n = 100$; the function at $n = 50$ looks similar, correctly having mean 0 at $S = 0$ but, incorrectly, mean far from 0 at most other S. The predictive distribution conditioned on the MAP model $\mathcal{M}_{\check{p}_{\text{map}(Z^{50})}}$ is a t-distribution with approximately $\check{p}_{\text{map}(Z^{50})} \approx 50$ degrees of freedom, which means that it is approximately normal. Figure 4.4 shows that its variance is *much* smaller than the variance $\tilde{\sigma}^2$ at $S = 0$; as a result, its log-risk conditional on $U = 0$ is smaller than that of $\tilde{\theta} = (\tilde{p}, \tilde{\beta}, \tilde{\sigma}^2)$ by some large amount A. Conditioned at $S \neq 0$, its conditional mean is off by some amount, and its variance is, on average, slightly (but not much) smaller than $\tilde{\sigma}^2$, making its conditional log-risk given $U \neq 0$ larger than that of $\tilde{\theta}$ by an amount A' where, it turns out, A' is smaller than A. Both events $S = 0$ and $S \neq 0$ happen with probability $1/2$, so that the final, unconditional log-risk of $\bar{f}(\cdot \mid Z^{50})$ is smaller than that of $\tilde{\theta}$.

Summarizing, hypercompression occurs because the variance of the predictive distribution conditioned on past data and a new X is much smaller than $\tilde{\sigma}^2$ at $X = 0$. This suggests that, if instead of a prior on σ^2 we use models $\mathcal{M}_p$ with a fixed σ^2, we can only get hypercompression (and correspondingly bad square-risk behaviour) if $\sigma^2 \ll \tilde{\sigma}^2$, because the predictive variance based on linear models $\mathcal{M}_p$ with fixed variance σ^2 given $X = x$ is, for all x, lower bounded by σ^2. Our experiments in Section 5.1.1 confirm that this is indeed what happens.

4.1.4 Explanation III: The mixability gap & the Bayesian belief in concentration

As we indicated at the end of Section 4.1.2, bad misspecification can occur only if the standard ($\eta = 1$) posterior is *nonconcentrated*.[1] Intriguingly, by formalizing 'concentration' in the appropriate way, we will now show, under some conditions on the prior, that a *Bayesian a priori always believes that the posterior will concentrate very fast*. Thus, if we observe data Z^n, and for many $n' \leq n$, the posterior based on $Z^{n'}$ is not concentrated, then we can view this as an indication of bad misspecification. In the next section we will see that SafeBayes selects a $\hat{\eta} \ll 1$ iff we have such nonconcentration at $\eta = 1$. Thus, SafeBayes can partially be understood as a prior predictive check, i.e. a test whether the assumptions implied by the prior actually hold on the data (Box, 1980).

The mixability gap We express posterior nonconcentration in terms of the *mixability gap* (Grünwald, 2012; De Rooij et al., 2014). In this section we only consider the special case of $\eta = 1$ (standard Bayes), for which the mixability gap δ_i is defined as the difference between 1-R-log-loss (3.18) and standard log-loss as obtained by predicting with the posterior predictive, at sample size i:

$$\begin{aligned}\delta_i &:= \mathbf{E}_{\theta\sim\Pi|z^{i-1}}\left[-\log f(y_i \mid x_i, \theta)\right] - \left(-\log \mathbf{E}_{\theta\sim\Pi|z^{i-1}}[f(y_i \mid x_i, \theta)]\right) \\ &= \mathbf{E}_{\theta\sim\Pi|z^{i-1}}\left[-\log f_\theta(y_i \mid x_i)\right] - \left(-\log \bar{f}(y_i \mid x_i, z^{i-1})\right), \qquad (4.6)\end{aligned}$$

Straightforward application of Jensen's inequality as in (3.19) gives that $\delta_i \geq 0$. δ_i, which depends on $z_1, \ldots, z_i$, is a measure of the posterior's concentratedness at sample size i when used to predict y_i given x_i: it is small if $f_\theta(y_i \mid x_i)$ does not vary much among the θ that have substantial η-posterior mass; by strict convexity of $-\log$, it is 0 iff there exists a set Θ_0 with $\Pi(\Theta_0 \mid Z^{i-1}) = 1$ such that for all $\theta, \theta' \in \Theta_0$, $f_\theta(y_i \mid x_i) = f_{\theta'}(y_i \mid x_i)$.

We set the *cumulative mixability gap* to be $\Delta_n := \sum_{i=1}^n \delta_i$.

The Bayesian belief in posterior concentration As a theoretical contribution of this chapter, we now show that, under some conditions on model and prior, if the data are as expected by the model and prior, then the expected mixability gap goes to 0 as $O((\log n)/n)$, and hence a Bayesian automatically a priori believes that the posterior will concentrate fast. For simplicity we restrict ourselves to a model $\mathcal{M} = \{P_\theta \mid \theta \in \Theta\}$ where Θ is countable, and we let all $\theta \in \Theta$ represent a conditional distribution for Y given X, extended to n outcomes by independence. We let π be a probability mass on Θ, and define the joint Bayesian distribution Π on $\Theta \times \mathcal{Y}^n \mid \mathcal{X}^n$ in the usual way, so that for measurable $\mathcal{A} \subset \mathcal{Y}^n$, $\Pi((\theta^*, \mathcal{A}) \mid X^n = x^n) = \pi(\theta^*) \cdot P_{\theta^*}(\mathcal{A} \mid X^n = x^n)$. The random variable θ^* refers to the θ chosen according to density π. We will

[1] Things would simplify if we could say 'bad misspecification can occur if and only if there is hypercompression', but we do not know whether that is the case; see Section 4.3.3.

look at the Bayesian probability distribution of the θ^*-expected mixability gap, $\bar{\delta}_n := \mathbf{E}_{\theta^*}[\delta_n]$.

Theorem 4.1. *Consider a countable model with prior Π as above. Suppose that the density ratios in Θ are uniformly bounded, i.e. there is a $v > 1$ such that for all $x, y \in \mathcal{X} \times \mathcal{Y}$, all $\theta, \theta' \in \Theta$, $f_\theta(y \mid x)/f_{\theta'}(y \mid x) \leq v$. Suppose that for some $\eta < 1$ we have $\sum_\theta \pi(\theta)^\eta < \infty$. Then for every $a > 0$ there are constants C_0 and C_1 such that, for all n,*

$$\Pi\left(\bar{\delta}_n \geq C_0 \cdot \frac{\log n}{n} \right) \leq C_1 \cdot \frac{1}{n^a}. \tag{4.7}$$

Moreover, for any $0 < a' \leq 1$, there exist C_2 and C_3 such that

$$\Pi\left(\Delta_n \geq C_2 \cdot n^{a'} \right) \leq C_3 \cdot \frac{(\log n)^2}{n^{a'}}, \tag{4.8}$$

i.e. the Bayesian believes that the mixability gap will be small on average and that the cumulative mixability gap will be small with high probability.

Thus, even though Δ_n is the difference between two quantities that are typically linear in n, with high probability it grows only polylogarithmically. This means that observing a large value of Δ_n strongly indicates misspecification.

> We hasten to add that the regularity conditions for Theorem 4.1 do *not* hold in the regression problem of Chapter 3; the theorem is merely meant to show that Δ_n is believed to be small in idealized circumstances that have been simplified so as to make mathematical analysis easier. Note however, that the regularity conditions do not constrain Θ in the most important respect: by allowing countably infinite Θ, we can approximate nonparametric models arbitrarily well by suitable covers (Cover and Thomas, 1991). In particular we do allow sets Θ for which maximum likelihood methods would lead to disastrous overfitting at all sample sizes. Also the condition that $\sum \pi(\theta)^\eta < \infty$ is standard in proving Bayesian and MDL convergence theorems (Barron and Cover, 1991; Zhang, 2006a). In fact, since the constants C_0 and C_1 scale logarithmically in v, we expect that Theorem 4.1 can be extended to the regression setting we are dealing with here as long as all distributions in the model have exponentially small tails, using methods similar to those in Grünwald (2014).

Example 4.B. [Cumulative nonconcentration can (and will) go together with momentary concentration: Example 4.A, Bernoulli, cont.] Consider the first instance of the Bernoulli Example 4.A again, where we again look at what happens if both distributions are equally bad: $\mathcal{M} = \{P_{0.2}, P_{0.8}\}$, whereas $Y_1, Y_2, \ldots$ are i.i.d. $\sim P_{\theta^*}$ with $\theta^* = 1/2$. As we showed in that example, at any given n, with P_{θ^*}-probability at least 0.32, $\min_{\theta \in \{0.2, 0.8\}} \pi(\theta \mid Y^n) \approx 2^{-\sqrt{n}/2}$: the posterior puts almost all mass on one θ. Lemma 6 of Van Erven et al. (2011) shows that in such cases δ_n is small; in this case, $\delta_n \leq 2(e-2) \min_{\theta \in \{0.2, 0.8\}} \pi(\theta \mid Y^n)$ $\approx 1.42 \cdot 2^{-\sqrt{n}/2}$. Thus, the posterior *looks* exceedingly concentrated at time n, with nonnegligible probability (this unwarranted confidence is a simplified

version of what was called the *fair balance paradox* by Yang (2007b), who conjectured it is the underlying reason for the problem of 'overconfident posteriors' in Bayesian phylogenetic tree inference). However, SafeBayes detects misspecification by looking at *cumulative* concentration, i.e. the sum of the δ's: L as in (4.1) can be interpreted as a random walk on $\mathbf{Z}$ starting at the origin, with equal probabilities to move to the left and to the right. By the central limit theorem, the random walk crosses the origin at time n with probability about $1/\sqrt{n\pi/2} = \tilde{O}(n^{-1/2})$, so that we may conjecture that, with high probability, it crosses the origin $\tilde{O}(n \cdot n^{-1/2}) = \tilde{O}(n^{1/2})$ times. Each time it crosses the origin, the posterior is uniform and hence as nonconcentrated as it can be, and Δ_n is increased by at least a fixed constant. One would therefore expect (under the 'true' θ^*) that Δ_n is of order $\sqrt{n}$, which by Theorem 4.1 is much larger than a Bayesian a priori expect it to be — the model fails the 'prior predictive check'.[2]

4.2 How SafeBayes works

In its simplest form, the in-model fixed variance case, SafeBayes finds the $\hat{\eta}$ that minimizes cumulative square-loss on the sample and thus can simply be understood as a pragmatic attempt to find a $\hat{\eta}$ that achieves small risk. However, the other versions of SafeBayes do not have such an easy interpretation. To explain them further, we need to generalize the notion of mixability gap in terms of the η'-*flattened* η-generalized Bayesian predictive density. The latter is defined, for $\eta, \eta' \leq 1$, as:

$$\bar{f}(y_i \mid x_i, z^{i-1}, \langle \eta' \rangle; \eta) := \left(\mathbf{E}_{\theta \sim \Pi \mid z^{i-1}, \eta} \left[f_\theta^{\eta'}(y_i \mid x_i) \right] \right)^{1/\eta'}. \tag{4.9}$$

By Jensen's inequality, we have $\bar{f}(y_i \mid x_i, z^{i-1}, \langle \eta' \rangle; \eta) \leq \bar{f}(y_i \mid x_i, z^{i-1}, \eta)$ for any $\eta' \leq 1$ and any (x_i, y_i). Indeed, intentionally, $\bar{f}(\cdot \mid \langle \eta' \rangle; \eta)$ is a 'defective' density in the sense that $\int_{\mathbf{R}} \bar{f}(y \mid x_i, z^{i-1}, \langle \eta' \rangle; \eta) dy < 1$. The log-loss achieved by η-generalized, η'-flattened Bayesian prediction is called (η, η')-*mix-loss* from now on, following terminology from De Rooij et al. (2014). For $0 < \eta \leq \eta' \leq 1$, the *mixability gap* $\delta_{i,\eta,\eta'}$ is defined as the difference between the η-R-log-loss and the η'-mix-loss:

$$\delta_{i,\eta,\eta'} := \mathbf{E}_{\theta \sim \Pi \mid Z^{i-1}, \eta} \left[-\log f_\theta(Y_i \mid X_i) \right] - \left(-\log \bar{f}(Y_i \mid X_i, Z^{i-1}; \langle \eta' \rangle; \eta) \right). \tag{4.10}$$

We once again define a cumulative version $\Delta_{n,\eta,\eta'} = \sum_{i=1}^n \delta_{i,\eta,\eta'}$, and note that the definitions are compatible with the special cases $\delta_i := \delta_{i,1,1}$ and $\Delta_n := \Delta_{n,1,1}$ defined in the previous subsection. Now we can rewrite the cumulative R-log-

[2] This heuristic argument can actually be formalized: if data are i.i.d. Bernoulli$(1/2)$, then the expected regret for every absolute loss predictor is of order $\tilde{O}(n^{1/2})$ (Cesa-Bianchi and Lugosi, 2006), which implies, via the connections between regret and Δ_n given by De Rooij et al. (2014), that Δ_n must also be of order $n^{1/2}$; we omit further details.

loss achieved by Bayes with the η-generalized posterior as

$$\sum_{i=1}^{n} \mathbf{E}_{\theta\sim\Pi|z^{i-1},\eta}\left[-\log f_\theta(y_i \mid x_i)\right] = \Delta_{n,\eta,\eta'} + \mathrm{CML}_{n,\eta,\eta'}, \tag{4.11}$$

where

$$\mathrm{CML}_{n,\eta,\eta'} = \left(\sum_{i=1}^{n} -\log \bar{f}(y_i \mid x_i, z^{i-1}, \langle\eta'\rangle; \eta)\right)$$

is the cumulative (η,η')-mix-loss. (4.11) holds for all $0 < \eta \leq \eta' \leq 1$. Consider first $\eta' = 1$. As was seen, if $\Delta_{n,1,1}$ is large, then this indicates potential bad misspecification. But (4.11) still holds for smaller $\eta' < 1$; by Jensen's inequality, for any fixed η, decreasing η' will make $\Delta_{n,\eta,\eta'}$ smaller as well. Indeed, for any fixed P^*, defining

$$\bar{\delta}_{\eta'} := \sup_W \mathbf{E}_{X,Y\sim P^*}\left[\mathbf{E}_{\theta\sim W}[-\log f_\theta(Y \mid X)] - \left(-\frac{1}{\eta'}\log \mathbf{E}_{\theta\sim W}[f_\theta(Y \mid X)^{\eta'}]\right)\right],$$

where the supremum is over *all* distributions on Θ, we have

$$\lim_{\eta'\downarrow 0} \bar{\delta}_{\eta'} = 0,$$

so we have an upper bound on the expectation of $\Delta_{n,\eta,\eta'}$ independent of the actual data that, for small enough η', will become negligibly small. But the left-hand side of (4.11) does not depend on η', so if, by decreasing η', we decrease $\Delta_{n,\eta,\eta'}$, $\mathrm{CML}_{n,\eta,\eta'}$ must increase by the same amount — so as yet we have gained nothing. Indeed, not surprisingly, Barron's bound does not hold any more for $\mathrm{CML}_{n,\eta,\eta'}$ with $\eta = 1$ and $\eta' < 1$ (and in general, it does not hold for η,η' whenever $\eta' < \eta$). *But*, it turns out, a version of Barron's bound still holds for $\mathrm{CML}_{n,\eta',\eta'}$, for all $\eta' > 0$: the cumulative log-risk of η'-flattened, η'-generalized Bayes is still guaranteed to be within a small RED_n of the cumulative log-risk of $\tilde{\theta}$, although RED_n does monotonically increase as η' gets smaller — simply because the prior becomes more important relative to the data (standard results in learning theory show that $\mathrm{CML}_{n,\eta,\eta}$ is monotonically decreasing in η, and can be upper bounded as $O(1/\eta)$; see e.g. (De Rooij et al., 2014, Lemma 1). Thus, it makes sense to consider the special case $\eta' = \eta$, and think of SafeBayes as finding the η minimizing

$$\sum_{i=1}^{n} \mathbf{E}_{\theta\sim\Pi|z^{i-1},\eta}\left[-\log f_\theta(y_i \mid x_i)\right] = \Delta_{n,\eta,\eta} + \mathrm{CML}_{n,\eta,\eta}, \tag{4.12}$$

since we have clear interpretations of both terms: the second indicates, by Barron's bound, how much worse the η-generalized posterior predicts in terms of log-loss compared to the optimal $\tilde{\theta}$; the first indicates how much is additionally lost if one is forced to predict by distributions inside the model. The second term decreases in η, the first has an upper bound which increases in

η. SafeBayes can now be understood as trying to minimize both terms at the same time.

Now broadly speaking, the central convergence result of Grünwald (2012) states that $\Delta_{n,\eta,\eta}$ will be 'sufficiently small' for all $\eta < 1$, and under some further conditions even for $\eta = 1$, if the model is correct or convex; and it will also be 'sufficiently small' if the model is incorrect, as long as η is smaller than some 'critical' value η_{crit} (which may depend on n though). Here 'sufficiently small' means that it is not the dominating term in (4.12). Intuitively, we would like the $\hat{\eta}$ determined by SafeBayes to be the largest η that is smaller than η_{crit}. Grünwald (2012) shows that SafeBayes indeed finds such an η, and that prediction based on the generalized posterior with this η achieves good frequentist convergence rates.

Experimental illustration Consider the main wrong-model experiment of Section 3.5. Figure 4.5 shows, as a function of η, in red, the cumulative η-R-log-loss measured by SafeBayes, averaged over 30 runs of the wrong-model experiment of Figure 3.3. In each individual run, SafeBayes picks the $\hat{\eta}$ minimizing this quantity; we thus get that on most runs, $\hat{\eta}$ is close to 0.4. In contrast to η-R-log-loss, and as predicted by theory, the η-mix-loss (in purple) decreases monotonically and coincides with the standard Bayesian log-loss at $\eta = 1$ and with the η-R-log-loss as $\eta \downarrow 0$. We also see hypercompression again: near $\eta = 1$, both the Bayesian log-loss and the mix-loss are smaller than the log-loss achieved by the best $\tilde{\theta}$ in the model. At $\eta = 0.5$, there is a sudden sharp rise in $\Delta_{n,\eta,\eta}$ (the difference between the red and purple curves). We can think of SafeBayes as trying to identify this 'critical' η_{crit}.

> Theorem 4.1 shows that, if both model and prior are well-specified, then the Bayesian posterior cumulatively concentrates in a very strong sense. More generally, if the model is correct but also if there is 'benign' misspe-

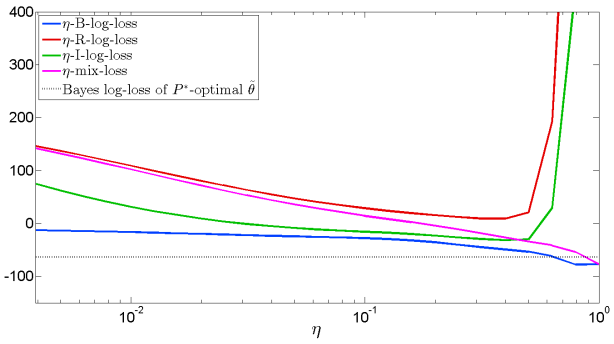


Figure 4.5: Cumulative losses up to sample 100 (where the posterior has not converged yet) as a function of η, averaged over 30 runs, for the experiment of Figure 3.3. η-B-log-loss is the cumulative log-loss achieved by standard Bayes with the η-generalized posterior.

cification, then, under some conditions on the prior, by the results of Grünwald (2012), the Bayesian posterior eventually cumulatively concentrates at $\eta = 1$. One might thus be tempted to interpret η_{crit} (the learning rate which SafeBayes tries to learn) as 'largest learning rate at which the posterior cumulatively concentrates'. However, this interpretation works only if $\eta_{\text{crit}} = 1$. If $\eta_{\text{crit}} < 1$, we can only show that, for every $\eta < \eta_{\text{crit}}$, $\Delta_{n,\eta,\eta}$ is small; true cumulative concentration would instead mean that $\Delta_{n,\eta,1}$ is small for such η (note we must have $\Delta_{n,\eta,\eta} \leq \Delta_{n,\eta,1}$ by Jensen). The figure shows that $\Delta_{n,\eta,1}$ (the difference between the red and blue curve) may indeed be large even at small η. A better interpretation is that, for every fixed η, with decreasing η', the geometry of the (η, η')-mix-loss changes, so that the loss difference between the mix loss and the R-log-loss obtained by randomization gets smaller. By then further using the generalized posterior for the same η', we guarantee that a version of Barron's bound holds for the (η', η')-mix-loss.

Replacing R- by I-loss Although the proofs of Grünwald (2012) are optimized for R-SafeBayes, the same story as above can be told for any fixed transformation from the posterior to a possibly randomized prediction, i.e. anything of the form (3.21); in particular for the most extreme transformation where we replace the posterior predictive by the distribution indexed by the posterior mean parameters so that instead of R-SafeBayes we end up with I-SafeBayes. In fact, the importance of the distinction between 'in-model' and 'out-model' prediction under model misspecification has been emphasized before (Grünwald, 2007; Barron and Hengartner, 1998; Kotłowski et al., 2010). In general, although we do not know how to exploit this intuition to strengthen the convergence proofs of Grünwald (2012), it seems more natural to replace the randomized predictions by deterministic, in-model predictions.

4.3 Discussion, open problems and conclusion

"If a subjective distribution Π attaches probability zero to a non-ignorable event, and if this event happens, then Π must be treated with suspicion, and *modified* or replaced" (emphasis added)
— A.P. Dawid (1982).

"Some models are obviously wrong, yet evidently useful"
— (very freely paraphrasing Box, 1979).

We already discussed the theoretical significance of the inconsistency result in the introduction. Extensive further discussion on Bayesian inference under misspecification is given by Walker (2013) and Grünwald and Langford (2007). For us, it remains to discuss the place of both the inconsistency result and our solution in Bayesian methodology.

Following the well-known Bayesian statisticians Box (1980), Good (1983), Dawid (1982, 2004) and Gelman (2004) (see also Gelman and Shalizi, 2012), we take the stance that model checking is a crucial part of successful Bayesian

practice. When there is a large discrepancy between a model's predictions and actual observations, it is not merely sufficient to keep gathering data and update one's posterior: something more radical is needed. In many such cases, the right thing to do is to go back to the drawing board and try to devise a more realistic model. However, we think this story is incomplete: in machine learning and pattern recognition, one often encounters situations in which the model employed is *obviously* wrong in some respects, yet there is a model instantiation (parameter vector) that is *pretty adequate* for the specific prediction task one is interested in. Examples of such obviously-wrong-yet-pretty-adequate models are, like in Chapter 3, assuming homoskedasticity in linear regression when the goal is to approximate the true regression function and the true noise is heteroskedastic,[3] but also the use of N-grams in language modelling (is the probability of a word given the previous three words really independent of everything that was said earlier?), logistic regression in e.g. spam filtering, and every single successful data compression method that we know of (see *Bayes and Gzip* (Grünwald, 2007, Chapter 17, page 537)). The difference with the more standard statistical (be it Bayesian or frequentist) mode of reasoning is eloquently described in Breiman's (2001) *the two cultures*.[4] Bayesian inference is among the most successful methods currently used in the obviously-wrong-yet-pretty-adequate-situation (to witness, state-of-the-art data compression methods such as Context-Tree-Weighting (Willems et al., 1995) have a Bayesian interpretation). Yet our results show that there is a danger: even *if* the employed model is pretty adequate (in the sense of containing a pretty good predictor), the Bayesian machinery might not be able to find it. The SafeBayesian algorithm can thus be viewed as an attempt to provide an alternative for the *data-analysis cycle* (Gelman and Shalizi, 2012) to this, in some sense, less ambitious setting: just like in the standard cycle, we do a model check, albeit a very specific one: we check whether there is 'cumulative concentration of the posterior' (see Section 4.1.4). If there is not, we know that we may not be learning to predict as well as the best predictor in our model, so we *modify* our posterior. Not in the strong sense of 'going back to the drawing board', but in the much weaker sense of making the learning rate smaller — we cannot hope that our model of reality has improved, because we still employ the same model — but we can now guarantee that we are doing the best we can with our given model, something which may be enough for the task at hand and which, as our experiments show, cannot always be achieved with standard Bayes.

[3] As long as, as in Chapter 3, the tails of the conditional distribution of Y given $X = x$ are sub-Gaussian, for each x; if they are not, there may be real outliers and then one cannot say that the model is 'pretty adequate' any more.

[4] The 'two cultures' does *not* refer to the Bayesian-frequentist divide, but to the modelling vs. prediction-divide. We certainly do not take the extreme view that statisticians should *only* be interested in prediction tasks such as classification and square-error prediction rather than density estimation and testing; our point is merely that in some cases, the goal of inference is clearly defined (it could be classification, but it could also be determination whether some random variables are (conditionally) (in)dependent), whereas part of our model is unavoidably misspecified; and in such cases, one may want to use a generalized form of Bayesian inference.

Benign vs. bad misspecification One might argue that the example of Chapter 3 is rather extreme, and that in practical situations, choosing a learning rate different from 1 may never be a useful thing to do. A crucial point here is that one can have 'benign' and 'bad' misspecification (Section 4.1.2). Under benign misspecification, standard Bayes with $\eta = 1$ will behave nicely under weak assumptions on the prior. While in our particular example, after 'eyeballing' the data one would probably have chosen a different, less misspecified model, it may be the case that 'bad' misspecification (as in Figure 4.1) also occurs, at least to some extent, in general, real-world data and is then not so easily spotted. Since we simply do not know whether such situations occur in practice, to be on the safe side, it seems desirable to have a theory about when we can get away with using standard Bayesian inference for a given prediction task even if the model is wrong, and how we can still use it with little modification if there is bad misspecification. Our work (esp. (Grünwald, 2014), the theoretical counterpart to Chapters 3–5) is a first step in this direction.

Towards a theory of Bayesian inference under misspecification What we have in mind is a theory of Bayesian inference under misspecification, in which the *goal* of learning plays a crucial role. The standard Bayesian approach is very ambitious: it can be used to solve every conceivable type of prediction or inference task. Every such task can be encoded as a loss or utility function, and, given the data and the prior, one merely has to calculate the posterior, and then makes an optimal decision by taking the act that minimizes expected loss or maximizes expected utility according to the posterior. Crucially, one uses the same posterior, independently of the utility function at hand, implying that one believes that one's own beliefs are correct *in every possible respect*. We envision a more modest approach, in which one acknowledges that one's beliefs are only adequate in some respects, not in others; how one proceeds then depends on how one's model and loss function interact. For example, if one is interested in data compression then, this problem being essentially equivalent to cumulative log-loss prediction, by Barron's (1998) bound one can simply use the standard $(\eta = 1)$ Bayesian predictive distribution — even under misspecification, this will guarantee that one predicts (at least!) as well as one could with the best element of one's model. If, on the other hand, one is interested in any of the KL-associated inference tasks (for linear models, these are square-loss and reliability, Section 3.2.3), then using $\eta = 1$ is not sufficient any more, and one may have to learn η from the data, e.g. in the SafeBayesian manner. Finally, if we are interested in an inference task that is not KL-associated under our model (i.e., a model instance can be good in the KL sense but bad in the task of interest), then a more radical step is needed: either go back to the drawing board and design a new model after all; or perhaps, the model can be changed in a more pragmatic way so that, for the right η, η-generalized Bayes once again will find the best predictor for the task at hand. Let us outline such a procedure for the case that the inference talk is simply prediction under some loss function $\ell : \mathcal{Y} \times \hat{\mathcal{Y}} \to \mathbf{R}$. In this case, if the ℓ-risk is not KL-associated this simply means that, for some data, the log likelihood is not a monotonic

function of the loss ℓ. To get the desired association, we may associate each conditional distribution $P_\theta(Y \mid X)$ in the model with its associated Bayes act δ_θ: $\delta_\theta(x)$ is defined as the act $\hat{y} \in \hat{\mathcal{Y}}$ which minimizes $P_\theta \mid X = x$-expected loss $\mathbf{E}_{Y \sim P_\theta | X=x}[\ell(y, \hat{y})]$. We can then define a new set of densities

$$f^{\text{NEW}}_{\theta,\gamma}(y \mid x) = \frac{1}{Z(\gamma)} e^{-\gamma \ell(y, \delta_\theta(x))}, \tag{4.13}$$

and perform (generalized) Bayesian inference based on these. Note that this effectively replaces, for each θ, the full likelihood by a 'likelihood' in which some information has been lost, and is thus reminiscent of what is done in *pseudo-likelihood* (Besag, 1975), *substitution likelihood* (Jeffreys, 1961; Dunson and Taylor, 2005), or *rank-based likelihood* (Gu and Ghosal, 2009) approaches (as a Bayesian, one may not want to lose information, but whether this still applies in nonparametric problems (Robins and Wasserman, 2000) let alone under misspecification (Grünwald and Halpern, 2004) is up to debate).

(4.13) can be made precise in two ways: either one just sets γ and $Z(\gamma)$ to 1, and allows the f^{NEW}_θ to be pseudo-densities, not necessarily integrating to 1 for each x. This is a standard approach in learning theory (Zhang, 2006b; Catoni, 2007). One could then learn η by, e.g., the basic SafeBayes algorithm with $\ell_\theta(x, y) := \ell(y, \delta_\theta(x))$ instead of log-loss. Or, one could define $Z(\gamma)$ so that the densities normalize (how to achieve this if $\int_y e^{-\gamma \ell(y, \delta_\theta(x))} dy$ depends on x is explained by Grünwald (2008)) and put a prior on γ as well (for linear models, this is akin to putting a prior on the variance). This will make the loss ℓ KL-associated and the KL-optimal $\tilde{\theta}$ will also have the reliability property, see again (Grünwald, 2008) for details. In this case we will get, with $z_i = (x_i, y_i)$, $\ell_\theta(z_i) := \ell(y_i, \delta_\theta(x_i))$, and using a prior on Θ and the scaling parameter γ, that the η-generalized posterior becomes

$$\pi(\theta, \gamma \mid z^n, \eta) \propto \frac{1}{Z(\gamma)^{\eta n}} e^{-\eta \gamma \sum_{i=1}^n \ell_\theta(z_i)} \cdot \pi(\theta, \gamma). \tag{4.14}$$

This idea was, in essence, already suggested by (Grünwald, 1998, Example 5.4) (see also Grünwald (1999)) under the name of *entropification* (however, Grünwald's papers wrongly suggest that, by introducing the scale parameter γ, it would be sufficient to only consider $\eta = 1$); see also (Lacoste-Julien et al., 2011; Quadrianto and Ghahramani, 2014).

Now both 'pure' subjective Bayesians and 'pure' frequentists might dismiss this programme as severe ad-hockery: the strict Bayesian would claim that nothing is needed on top of the Bayesian machinery; the strict frequentist would argue that Bayesian inference was never designed to 'work' under misspecification, so in misspecified situations it might be better to avoid Bayesian methods altogether rather than trying to 'repair' them. We strongly disagree with both types of purism, the reason being the ever-increasing number of successful applications of Bayesian methods in machine learning in situations in which models are obviously wrong. We would like to challenge the pure subjective Bayesian to explain this success, given that the statistician is using

a priori distributions that reflect beliefs which she knows to be false, and are thus not really her beliefs. We would like to challenge the pure frequentist to come up with better, non-Bayesian methods instead. In summary, we would urge both purists not to throw away the Bayesian baby with the misspecified bath water!

Moreover, from a prequential (Dawid, 1984), learning theory (citations see below) and Minimum Description Length (MDL (Barron et al., 1998)) perspective, the extension from Bayes to SafeBayes is *perfectly natural*. From the prequential perspective, SafeBayes seeks to find the largest η at which the generalized Bayesian predictions have a predictive interpretation in terms of the loss of interest rather than the log-loss. The learning theory and MDL perspectives are further explained in the next section.

4.3.1 Related work I: Learning theory and MDL

Learning theory From the learning theory perspective, generalized Bayesian updating as in (4.14) with $Z(\gamma)$ set to 1 can be seen as the result of a simple regularized loss minimization procedure (this was probably first noted by Williams (1980); see in particular (Zhang, 2006b)), which means that it continues to make sense if $\exp(-\gamma\ell_\theta)$ as in (4.13) does not have a direct probabilistic interpretation. Variations of such generalized Bayesian updating are known as "aggregating algorithm", "Hedge" or "exponential weights", and often have good worst-case optimality properties in nonstochastic settings (Vovk, 1990; Cesa-Bianchi and Lugosi, 2006) — but to get these the learning rate must often be set as small as $O(1/\sqrt{n})$. Similarly, PAC-Bayesian inference (Audibert, 2004; Zhang, 2006b; Catoni, 2007) (for a variation, see (Freund et al., 2004)) is also based on a posterior of form (4.13) and can achieve minimax optimal rates in e.g. classification problems by choosing an appropriate η, usually also very small. From this perspective, SafeBayes can be understood as trying to find a *larger* η than the worst-case optimal one, if the data indicate that the situation is not worst-case and faster learning is possible. Finally, Bissiri et al. (2013) give a motivation for (4.14) (with $Z(\gamma) \equiv 1$) based on coherence arguments that are more Bayesian in flavour.

MDL Of particular interest is the interpretation of the SafeBayesian method in terms of the MDL principle for model selection, which views learning as data compression. When several models for the same data are available, MDL picks the model that extracts the most 'regularity' from the data, as measured by the minimum number of bits needed to code the data *with the help of the model*. This is an interpretation that remains valid even if a model is completely misspecified (Grünwald, 2007). The resulting procedure (based on so-called *normalized maximum likelihood* codelengths) is operationally almost identical to Bayes factor model selection. Thus, it provides a potential answer to the question 'what does a high posterior belief in a model really mean, since one knows all models under consideration to be incorrect in any case?' (asked by, e.g., Gelman and Shalizi (2012)): even if all models are wrong, the information-

theoretic MDL interpretation stands. However, our work implies that there is a serious issue with these NML codes: note that any distribution P in a model $\mathcal{M}$ can be mapped to a code (the *Shannon-Fano code*) that would be optimal in expectation if data were sampled from P. Now, our work shows that if the data are sampled from some $P^* \notin \mathcal{M}$, then the codes based on Bayesian predictive distributions can sometimes compress substantially *better* in expectation than can be done based on any $P \in \mathcal{M}$ — this is the hypercompression phenomenon of Section 4.1.3. The same thing then holds for the NML codes, which assign almost the same codelengths as the Bayesian ones. Our work thus invalidates the interpretation of NML codelengths as 'compression with the help of (and only of!) the model', and suggests that, similarly to in-model SafeBayes one should design and use 'in-model' versions of the NML codes instead — codes that are guaranteed not to outperform, at least in expectation, the code based on the best distribution in the model.

4.3.2 Related work II: Analysis of Bayesian behaviour under misspecification

Consistency theorems The study of consistency and rate of convergence under misspecification for likelihood-based and specifically Bayesian methods go back at least to Berk (1966). For recent state-of-the-art work on likelihood-based, non-Bayesian methods see e.g. Dümbgen et al. (2011) and the very general Spokoiny (2012). Recent work on Bayesian methods includes Kleijn and Van der Vaart (2006), De Blasi and Walker (2013) and Ramamoorthi et al. (2013) who obtained results in quite general, i.i.d. nonparametric settings, non-i.i.d. settings (Shalizi, 2009), and more specific settings (Sriram et al., 2013); see also Grünwald (2014). Yet, as explicitly remarked by De Blasi and Walker (2013), the conditions on model and prior needed for consistency under misspecification are generally stronger than those needed when the model is correct. Essentially, if the data are i.i.d. both according to the model and the sampling distribution P^*, then Theorem 1 (in particular its Corollary 1) of De Blasi and Walker (2013) implies the following: if, for all $\epsilon > 0$, the model can be covered by a finite number of ϵ-Hellinger balls, then the Bayesian posterior eventually concentrates: for all $\delta, \gamma > 0$, the posterior mass on distributions within Hellinger distance δ of the $P_{\tilde{\theta}}$ that is closest to P^* in KL divergence will become larger than $1 - \gamma$ for all n larger than some n_γ. This implies that both in the ridge regression (finite p) and in the model averaging experiments (finite $p_{\max}$), Bayes eventually 'recovers' — as we indeed see in our experimental results. However, if $p_{\max} = \infty$, then the model has no finite Hellinger cover any more for small enough ϵ and indeed the conditions for Theorem 1 of De Blasi and Walker (2013) do not apply any more. Our results show that in such a case we can indeed have inconsistency if the model is incorrect. On the other hand, even if $p_{\max} = \infty$, we do have consistency in the setup of our correct-model experiment for the standard Bayesian posterior, as follows from the results by Zhang (2006a).

The limiting $\eta = 1$ Like several earlier results (Barron and Cover, 1991; Walker and Hjort, 2002), Zhang's consistency results for correct models hold under very weak conditions for generalized Bayes with any $\eta < 1$, and only under much stronger conditions for $\eta = 1$. Zhang provides an example of inconsistency-like behaviour in the well-specified case with $\eta = 1$ that automatically disappears as soon as one picks $\eta < 1$, leading Zhang (2006a) to claim that in general, generalized Bayesian methods $(\eta < 1)$ are more stable than standard Bayesian ones. Zhang's example, and the example of Bayesian model selection inconsistency in a well-specified model by Csiszár and Shields (2000) are closely related to ours, in that the Bayes predictive distribution for $\eta = 1$ becomes significantly different from any distribution in the model (see Figure 4.1). In their examples, the problem is resolved by taking any $\eta < 1$; in our misspecification case, η should even be taken much smaller.

Anomalous behaviour and modifications of Bayesian posterior under misspecification Anomalous behaviour of Bayesian inference under misspecification was, of course, observed before, e.g. (less dramatically than here) by Yang (2007b); Müller (2013) and (as dramatically, but involving a very artificial model) Grünwald and Langford (2007). Presumably also related is the 'brittleness' of Bayesian inference that has been observed by Owhadi and Scovel (2013). Not surprisingly then, we are not the first to suggest modification of likelihood-based estimators (see e.g. White, 1982; Royall and Tsou, 2003; Kotłowski et al., 2010) and posteriors (Royall and Tsou, 2003; Hoff and Wakefield, 2012; Doucet and Shephard, 2012; Müller, 2013). The latter three approaches (that extend the first) employ the so-called *sandwich posterior*, in which the covariance matrix of the posterior is changed based on a 'sandwich formula' involving the empirical variance; Müller (2013) provides extensive explanation and experimentation. Compared to the sandwich approach, our proposal, besides being applicable in fully nonparametric contexts, seems substantially more radical. This can be seen from the regression applications in Müller (2013), which involve a noninformative Jeffreys' prior on the regression coefficient vector β. With such a prior (as well as any normal prior scaled by variance σ^2), the posterior *mean* of β, and thus also the frequentist square-risk (which only depends on the posterior mean) remains unaffected by the sandwich modification, so for square-risk the method would perform like standard Bayes in our model-wrong experiments. Thus Müller (2013, Section 2.4) demonstrates its usefulness on other loss functions. Nevertheless, both the sandwich and the SafeBayesian methods can be thought of as methods for measuring the spread of a posterior, and it would be useful to compare the two in detail, both in theory and practice.

4.3.3 Future work and open problems

The results of these chapters raise several issues and prompt the following research agenda:

1. The misspecification in our example would presumably be easily spotted in practice. This raises the question whether 'bad' misspecification also arises for data sets that occur in practice and for which it would not be easily spotted. Currently, we know only of one experiment in this direction: Jansen (2013) applied the Bayesian Lasso (Park and Casella, 2008) to several real-world data sets, where the λ (i.e. $1/\eta$) is taken that minimizes the cumulative *square-loss* whereas at the same time σ^2 is a free parameter. Thus it is a hybrid of I-square-SafeBayes and I-log-SafeBayes, but equal to neither; the method was (somewhat) outperformed by standard Bayes on most data sets tried. However, we also tried this hybrid method in the model-wrong experiment of Chapter 3 and found that it is not competitive with either of the two 'true' in-model SafeBayes methods either; so the experiment does not 'really' test SafeBayes; more precise experiments are needed.

2. Our method has one major disadvantage: even if the data do not have a natural ordering, the $\hat{\eta}$ selected by SafeBayes will, in general, be order-dependent. Grünwald (2011) suggested a very different (and in fact, the first) method to learn $\hat{\eta}$, that does not have this problem. However, it is only applicable to countable models, and has no obvious computationally efficient implementation, so we do not know whether it has a future. Another method that is clearly related to I-square-SafeBayes is to determine η using leave-one-out cross-validation based on the squared error. This method is also order-independent and behaves comparably to I-square-SafeBayes (Section 5.1.1), but it is not clear how to extend it to general misspecified models. While we show in the same section that cross-validation based on log-loss of the Bayes predictive distribution fails dramatically, it may be that cross-validation based on log-loss of the Bayes posterior *mean* would generally work fine, and this method can be applied to general misspecified models, not just linear ones. Compared to I-log-SafeBayes this *in-model log-loss cross-validation* would have the advantage that it is order independent, and the disadvantage that it cannot (at least not straightforwardly) be used in an online setting and/or for non-i.i.d. models. Also, we suspect that if the number of models is exponential in the covariates (as in variable selection), cross-validation may be prone to overfitting whereas SafeBayes would not be, but this is just extrapolation from the well-specified case: it would be useful to investigate "in-model cross-validation" further.

3. What exactly are relations between the sandwich posterior (see above) and our approach? It would be good to test SafeBayes on the data sets used by Müller (2013).

4. It would be useful to establish exactly what properties of Bayesian updating remain valid for generalized Bayesian updating, and what properties do not hold any more. For example, *telescoping* (Cesa-Bianchi and Lugosi, 2006) holds for the standard posterior, for the η-flattened, η-generalized

posterior, but not for the (nonflattened) η-generalized posterior.

5. As discussed at the end of Section 4.2, the final term in (3.23) is lacking in the in-model versions of SafeBayes, and this does suggest that they should work better than the randomization versions — the corresponding $\Delta_{\eta,\eta}$ is always smaller. Yet we have no theoretical results to this end, and our empirical results confirm this to some extent (R-square-SafeBayes is not competitive), but not fully (R-log-SafeBayes is competitive), so more research is needed here.

6. As we indicated in Section 4.1.3, hypercompression implies nonconcentration, but we do not know whether the reverse implication holds as well, so we may perhaps have bad misspecification yet no hypercompression. It would give significant insight if we knew whether this indeed could happen.

7. In light of the discussion underneath (4.13), one would like to formulate a general theory of substitution likelihoods so that likelihoods can be determined based on the inference task of interest, so that this task becomes KL-associated, for *arbitrary* prediction tasks. Ideally, (4.13) and approaches such as pseudo-likelihood and rank-based likelihood would all become a special case. If this can be done, we would have a truly generalized Bayesian method.

Appendix 4.A More on mix loss

4.A.1 Implementing SafeBayes

To implement the SafeBayesian algorithm (page 52), generalized posteriors must be computed for different values of η, and the randomized loss (3.18) must be computed for each sample size. For linear models with conjugate priors as considered in our experiments, all required quantities can be computed analytically. We have already seen how to do this for models $\mathcal{M}_p$ with fixed dimension p. For unions of such models, it turns out that the mix-loss is a helpful tool.

Role of mix loss in generalized posterior over models The generalized posterior *across* a discrete set of models is given by (3.7), which, writing $\tau = (\beta, \sigma^2)$, is, via (3.10) and (3.9), equivalent to

$$\begin{aligned}\pi(p \mid z^n, \eta) &= \int_{\Theta_p} \pi(p, \tau \mid z^n, \eta)\, d\tau \\ &\propto \int (f(y^n \mid x^n, \tau, p))^\eta \pi(\tau \mid p)\, d\tau\, \pi(p). \end{aligned} \tag{4.15}$$

Here $\propto$ means 'proportional to' when p is varied and z^n and η are fixed. In practice we prefer to calculate this quantity incrementally: the posterior for z^{n+1} with prior Π is equal to the posterior for a single data point z_{n+1} when the posterior for z^n is used as prior (in this sense the generalized posterior behaves like the standard posterior): using this to further rewrite the second line of (4.15) gives

$$\begin{aligned}&\pi(p \mid z^n, \eta) \\ &\quad\propto \int (f(y^n \mid x^n, \tau, p))^\eta \pi(\tau \mid p)\, d\tau\, \pi(p) \\ &\quad= \int (f(y_n \mid x_n, \tau, p))^\eta \cdot (f(y^{n-1} \mid x^{n-1}, \tau, p))^\eta \pi(\tau \mid p)\, d\tau\, \pi(p) \\ &\quad= \int (f(y_n \mid x_n, \tau, p))^\eta \\ &\qquad\cdot \left(\pi(\tau \mid z^{n-1}, p, \eta) \cdot \int (f(y^{n-1} \mid x^{n-1}, \tau')^\eta \pi(\tau' \mid p) d\tau' \right) d\tau\, \pi(p) \\ &\quad\propto \int (f(y_n \mid x_n, \tau, p))^\eta \cdot \pi(\tau \mid z^{n-1}, p, \eta)\, d\tau \cdot \pi(p \mid z^{n-1}, \eta),\end{aligned}$$

where in the third inequality we used the definition of the generalized posterior and in the last we used (4.15).

The integral appearing in both the cumulative and the step-wise expression equals the expectation in (4.9) from the η-flattened η-generalized Bayesian predictive density for n and 1 outcome respectively; $-\log[(\cdot)^{1/\eta}]$ of this quantity is the mix loss of model p. We will now derive formulas for this quantity.

Model with fixed variance Use the notation of Section 3.3.1. Write $\sigma^2_{\text{mix}} = \sigma^2(1/\eta + x_{n+1}\Sigma_n x_{n+1}^\top)$. Then the mix loss for predicting one new data point y_{n+1} is

$$-\log \bar{f}(y_{n+1} \mid x_{n+1}, z^n, \langle\eta\rangle; \eta) = \frac{1}{\eta}\left[\frac{1}{2}(\eta-1)\log(2\pi\sigma^2) + \frac{1}{2}\log\eta + \frac{1}{2}\log(2\pi\sigma^2_{\text{mix}}) + \frac{1}{2\sigma^2_{\text{mix}}}(y_{n+1} - x_{n+1}\beta_n)^2\right].$$

Model with conjugate prior on variance Using the notation of Section 3.3.1, the mix loss is given by

$$-\log \bar{f}(y_{n+1} \mid x_{n+1}, z^n, \langle\eta\rangle; \eta) = \frac{1}{\eta}\left[\frac{1}{2}\eta\log\pi + \frac{1}{2}\log(1+\eta x_{n+1}\Sigma_n x_{n+1}^\top) + a_{n+1}\log(2b_n + \frac{(y_{n+1}-x_{n+1}\beta_n)^2}{1/\eta + x_{n+1}\Sigma_n x_{n+1}^\top}) - a_n \log 2b_n - \log\frac{\Gamma(a_{n+1})}{\Gamma(a_n)}\right].$$

4.A.2 Belief in concentration (proof of Theorem 4.1)

For simplicity, we only give the proof for the unconditional case, in which the θ represent distributions P_θ on $z \in \mathcal{Z}$; extension to the conditional case is straightforward. For $0 < \eta < 1$, let $d_\eta(\theta^* \| \theta)$ denote the Rènyi divergence of order $1-\eta$ (Van Erven and Harremoës, 2014), i.e. $d_\eta(\theta^*\|\theta) = -\frac{1}{\eta}\log \mathbf{E}_{Z\sim\theta^*}\left(\frac{f_\theta(Z)}{f_{\theta^*}(Z)}\right)^\eta$. We first state a lemma, proved further below. In the lemma, as in the remainder of the proof, (θ^*, Z^n) is the random variable distributed according to the Bayesian distribution Π.

Lemma 4.2. *Let Θ, Π and π be as in the statement of Theorem 4.1. For every $1/2 \le \eta < 1$, $\epsilon > 0$, let $\bar{\Theta}_{\eta,\epsilon} := \{\theta \in \Theta \mid d_\eta(\theta^*\|\theta) > \epsilon\}$. For every $b > 0$ and every sample size n and setting $\epsilon := (b\log n)/(n\eta)$ and $c_\eta = (1-\eta)/(1+\eta(1-\eta))$, we have:*

$$\Pi\left(\Pi(\bar{\Theta}_{\eta,\epsilon} \mid Z^n) \ge n^{-bc_\eta}\right) \le 2\left(\sum_{\theta\in\Theta}\pi(\theta)^\eta\right)\cdot n^{-bc_\eta}.$$

In particular, if π is summable for some $\eta < 1$, then using $b = 1/c_\eta$, we get that the Bayesian probability that the posterior probability of the set of θ farther than $b(\log n)/n$ from θ^* exceeds $1/n$, is $O(1/n)$.

We proceed to prove Theorem 4.1 using this lemma. By the information inequality (Cover and Thomas, 1991), we have for every probability density $f \neq f_{\theta^*}$ that

$$\begin{aligned} D(\theta^*\|\theta) &= \mathbf{E}_{Z_n\sim P_{\theta^*}}[-\log f_\theta(Z_n) + \log f_{\theta^*}(Z_n)] \\ &\ge \mathbf{E}_{Z_n\sim P_{\theta^*}}[-\log f_\theta(Z_n) + \log f(Z)]. \end{aligned}$$

In particular this holds with $f = \bar{f} \mid Z^n$, the Bayes predictive distribution based on the sample seen so far. It then follows from (4.6) that

$$\bar{\delta}_n \leq \mathbf{E}_{\theta \sim \Pi | Z^n}[D(\theta^* \| \theta)] \tag{4.16}$$

Since π^η is decreasing in η, we may without loss of generality assume that the η mentioned in the theorem statement is at least $1/2$. Now note (Van Erven and Harremoës, 2014, Theorem 16) that for every $1/2 < \eta < 1$, $d_{1/2}(\theta^* \| \theta) \leq (\eta/(1-\eta)) \cdot d_\eta(\theta^* \| \theta)$. We also know from (Yang and Barron, 1999, Lemma 4) that the KL divergence $D(\theta^* \| \theta)$ satisfies $D(\theta^* \| \theta) \leq (2 + \log v) d_{1/2}(\theta^* \| \theta)$. Since trivially $d_\eta(\theta^* \| \theta) \leq \log v$, we have, with $C = \frac{\eta}{1-\eta} \cdot (2 + 2 \log v)$, for every $\epsilon > 0$, using (4.16),

$$\begin{aligned}
\bar{\delta}_n &\leq C \cdot \mathbf{E}_{\theta \sim \Pi | Z^n}[d_\eta(\theta^* \| \theta)] \\
&\leq C\Pi\left(d_\eta > \epsilon \mid Z^n\right) \log v + C\left(1 - \Pi\left(d_\eta > \epsilon \mid Z^n\right)\right) \epsilon \\
&\leq C\left(\Pi\left(d_\eta > \epsilon \mid Z^n\right) \log v + \epsilon\right),
\end{aligned}$$

so that $\Pi\left(d_\eta > \epsilon \mid Z^n\right) \geq (C^{-1}\bar{\delta}_n - \epsilon)/(\log v)$ and by Lemma 4.2, we have for $\epsilon = b(\log n)/(n\eta)$ as in the lemma, that

$$\Pi\left(\frac{C^{-1}\bar{\delta}_n - \epsilon}{\log v} \geq n^{-bc_\eta}\right) \leq 2\left(\sum_{\theta \in \Theta} \pi(\theta)^\eta\right) \cdot n^{-bc_\eta}.$$

Rewriting this expression, plugging in the value of ϵ and using $\eta \geq 1/2$, gives

$$\Pi\left(\bar{\delta}_n \geq C\left((\log v) n^{-bc_\eta} + \frac{2b(\log n)}{n}\right)\right) \leq 2\left(\sum_{\theta \in \Theta} \pi(\theta)^\eta\right) \cdot n^{-bc_\eta}. \tag{4.17}$$

The first part of the result follows by setting $b = a/c_\eta$. For the second result, note that the first result implies (take $a = 2$), by the union bound over sample sizes $1, \ldots, n$, that the Bayesian probability that $\mathbf{E}_{Z^n \sim \theta^*}[\Delta_n]$ exceeds $C_0 \sum_{i=1}^n (\log i)/i \asymp (\log n)^2$ is $O(1/n)$. Thus there exists C', C_0' such that the Bayesian probability that $\mathbf{E}_{Z^n \sim \theta^*}[\Delta_n]$ exceeds $C_0'(\log n)^2$ is bounded by C'/n. Thus for the probability in (4.8) we have

$$\begin{aligned}
\Pi\left(\Delta_n \geq C_2 \cdot n^{a'}\right) &= \Pi\left(\Delta_n \geq C_2 \cdot n^{a'}, \mathbf{E}_{Z^n \sim \theta^*}[\Delta_n] \geq C_0'(\log n)^2\right) \\
&\quad + \Pi\left(\Delta_n \geq C_2 \cdot n^{a'}, \mathbf{E}_{Z^n \sim \theta^*}[\Delta_n] < C_0'(\log n)^2\right) \\
&\leq \Pi\left(\mathbf{E}_{Z^n \sim \theta^*}[\Delta_n] \geq C_0'(\log n)^2\right) \\
&\quad + \Pi\left(\Delta_n \geq C_2 \cdot n^{a'}, \mathbf{E}_{Z^n \sim \theta^*}[\Delta_n] < C_0'(\log n)^2\right) \\
&\leq \frac{C'}{n} + \frac{C_0'(\log n)^2}{C_2 n^{a'}},
\end{aligned}$$

where in the final step we used Markov's inequality. The second result follows.

Proof of Lemma 4.2 Fix $A > 0$ and $\gamma > 0$. We have

$$\begin{aligned}
\Pi\left(\Pi(\bar{\Theta}_{\eta,\epsilon} \mid Z^n) \geq A\right) &= \Pi\left(\frac{\sum_{\theta\in\bar{\Theta}_{\eta,\epsilon}} \pi(\theta)\cdot f_\theta(Z^n)}{\sum_{\theta\in\Theta} \pi(\theta)\cdot f_\theta(Z^n)} \geq A\right) \\
&= \Pi\left(\frac{\sum_{\theta\in\bar{\Theta}_{\eta,\epsilon}} \pi(\theta)\cdot f_\theta(Z^n)}{f_{\theta^*}(Z^n)} \cdot \frac{f_{\theta^*}(Z^n)}{\sum_{\theta\in\Theta} \pi(\theta)\cdot f_\theta(Z^n)} \geq A\right) \\
&\leq \Pi\left(\frac{\sum_{\theta\in\bar{\Theta}_{\eta,\epsilon}} \pi(\theta)\cdot f_\theta(Z^n)}{f_{\theta^*}(Z^n)} \geq A^{1+\gamma}\right) + \Pi\left(\frac{f_{\theta^*}(Z^n)}{\sum_{\theta\in\Theta} \pi(\theta)\cdot f_\theta(Z^n)} \geq A^{-\gamma}\right),
\end{aligned} \tag{4.18}$$

where we used the union bound. The first term is equal to, and can be further bounded as

$$\begin{aligned}
&= \Pi\left(\frac{\left(\sum_{\theta\in\bar{\Theta}_{\eta,\epsilon}} \pi(\theta)\cdot f_\theta(Z^n)\right)^\eta}{(f_{\theta^*}(Z^n))^\eta} \geq A^{\eta(1+\gamma)}\right) \\
&\leq \Pi\left(\frac{\sum_{\theta\in\bar{\Theta}_{\eta,\epsilon}} \pi(\theta)^\eta\cdot (f_\theta(Z^n))^\eta}{(f_{\theta^*}(Z^n))^\eta} \geq A^{\eta(1+\gamma)}\right) \\
&= \sum_{\theta^*} \pi(\theta^*) P_{\theta^*}\left(\frac{\sum_{\theta\in\bar{\Theta}_{\eta,\epsilon}} \pi(\theta)^\eta\cdot (f_\theta(Z^n))^\eta}{(f_{\theta^*}(Z^n))^\eta} \geq A^{\eta(1+\gamma)}\right) \\
&\leq \sum_{\theta^*\in\Theta} \pi(\theta^*)\, \mathbf{E}_{Z^n\sim P_{\theta^*}}\left[\frac{\sum_{\theta\in\bar{\Theta}_{\eta,\epsilon}} \pi(\theta)^\eta\cdot (f_\theta(Z^n))^\eta}{(f_{\theta^*}(Z^n))^\eta}\right]\cdot A^{-\eta(1+\gamma)} \\
&= \sum_{\theta^*\in\Theta} \pi(\theta^*) \sum_{\theta\in\bar{\Theta}_{\eta,\epsilon}} \pi(\theta)^\eta \cdot \left(\mathbf{E}_{Z\sim P_{\theta^*}}\left[\frac{(f_\theta(Z))^\eta}{(f_{\theta^*}(Z))^\eta}\right]\right)^n \cdot A^{-\eta(1+\gamma)} \\
&\leq \left(\sum_{\theta\in\bar{\Theta}_{\eta,\epsilon}} \pi(\theta)^\eta\right) e^{-n\eta\epsilon}\cdot A^{-\eta(1+\gamma)}.
\end{aligned}$$

where the first inequality follows by differentiation to η (or equivalently, by monotonicity of ℓ^p-norms), the second is Markov's, and the third is the definition of Rènyi divergence.

The second term in (4.18) can be bounded as

$$\begin{aligned}
&\leq \Pi\left(\frac{f_{\theta^*}(Z^n)}{\pi(\theta^*)\cdot f_{\theta^*}(Z^n)} \geq A^{-\gamma}\right) = \Pi(\pi(\theta^*)^{-1+\eta} \geq A^{-(1-\eta)\gamma}) \\
&\leq \mathbf{E}_{\theta^*\sim\Pi}[\pi(\theta^*)^{-1+\eta}] A^{\gamma(1-\eta)} = \sum_{\theta^*} \pi(\theta^*)^\eta A^{\gamma(1-\eta)}.
\end{aligned}$$

Combining the upper bounds on the two terms on the right in (4.18), we get:

$$\Pi\left(\Pi(\bar{\Theta}_{\eta,\epsilon} \mid Z^n) \geq A\right) \leq \left(\sum_{\theta\in\bar{\Theta}_{\eta,\epsilon}} \pi(\theta)^\eta\right)\left(e^{-n\eta\epsilon}\cdot A^{-\eta(1+\gamma)} + A^{\gamma(1-\eta)}\right).$$

Now we plug in the chosen value of $\epsilon = (b \log n)/(n\eta)$ and we set $A = n^{-b/(\gamma+\eta)}$. With these values the second factor on the right becomes

$$e^{-n\eta\epsilon} \cdot A^{-\eta(1+\gamma)} + A^{\gamma(1-\eta)} \\ = n^{-b} n^{b(\eta(1+\gamma))/(\gamma+\eta)} + n^{-b\gamma(1-\eta)/(\gamma+\eta)} = 2n^{-b\cdot\gamma\cdot\frac{1-\eta}{\gamma+\eta}}.$$

Since this holds for all $\gamma > 0$, it also holds for $\gamma = 1/(1-\eta)$, and the result follows.

Chapter 5

Bayesian Inconsistency: More Experiments

This chapter provides additional linear regression experiments to test whether the results of Chapter 3 also hold with different priors, models, methods, and data-generating distributions. We find that three versions of SafeBayes consistently perform well, while other methods, including Bayes and AIC, perform badly.

5.1 Experiments on variations of the prior and the model

Apart from the priors on parameters given the models we used in our main experiments, we tried several alternative prior distributions, described in the subsections below. The first subsection describes experiments with fixed (i.e., a degenerate prior on) σ^2.

5.1.1 Experiments with fixed σ^2

When models with fixed σ^2 are used, our two SafeBayes methods become *R*-square- and *I*-square-SafeBayes, as defined in Section 3.4.2. These also have a direct interpretation as trying to find the best η for predicting with a square-loss function, as was explained in that section. In this context, the value $\eta = 1$ has no special status, so we now also tried values $\eta > 1$ (we did experiment with varying η in the previous varying σ^2 experiments as well, but there it did not make any substantial difference in the results). Specifically, we set $\mathcal{S}_n$ in the SafeBayesian algorithm to $\{2^{\kappa_{\max}}, 2^{\kappa_{\max}-\kappa_{\text{STEP}}}, 2^{\kappa_{\max}-2\kappa_{\text{STEP}}}, \ldots, 2^{-\kappa_{\max}}\}$, with $\kappa_{\text{STEP}} = 1/2$ and $\kappa_{\max} = 6$. All priors on the regression coefficients β remain as described in Section 3.5.1.

5.1.1.1 Model averaging experiment, fixed σ^2

The model-correct experiment showed no surprises (all methods performed well), so we only show results for the model-wrong experiment, as described in Section 3.5.1, testing each of Bayes, R-square- and I-square-SafeBayes twice: once based on a model with variance σ^2 overly large (3 times $\tilde{\sigma}^2$), and once with σ^2 overly small (1/3 times $\tilde{\sigma}^2$) variance. To allow precise comparison with the results in the main text, we also show behaviour of R-log-SafeBayes with varying variance (defined precisely as in Figure 3.3) in Figure 5.1.

5.1.1.2 Ridge regression experiments, fixed σ^2

Again we only show results for the model-wrong experiment.

Note that here standard Bayes — as can be seen from plugging $\eta = 1$ into (3.12) — does not depend on σ^2 and thus coincides in terms of square-risk behaviour with standard Bayes in the variable σ^2 case as in Figure 3.7. Also (see below (3.12)) I-square-SafeBayes for fixed σ^2 does not itself depend on σ^2 and simply minimizes the cumulative sum of squared errors.

Just as for ridge regression with variable σ^2, one may equivalently interpret the η-generalized-posterior means $\bar{\beta}_{i,\eta}$ as the standard, nongeneralized Bayesian posterior means that one would get with a modified prior on β, proportional to the original prior raised to the power η^{-1} (see above (3.31), Section 3.5.4). It may then once again seem reasonable to learn η itself in a Bayesian or likelihood-based way such as empirical Bayes.[1] Indeed, this was suggested implicitly as early as 1999 by one of us (Grünwald, 1999). The procedure described in Section 3.4.3 ('hierarchical loss') of Bissiri et al. (2013) also arrives, via a different derivation, at a similar prescription for finding η (we immediately add that the authors describe many ways for determining η, of which this is just one). Unfortunately, just as for the empirical Bayes learning of η with varying σ^2, the figures below indicate that it does not perform well at all.

Conclusion Standard Bayes again performs comparably badly in both experiments (note the difference in scale in the first graphs of Figures 5.1 and 5.2). I-square-SafeBayes behaves excellently in both experiments. But now in the ridge experiment R-square-SafeBayes becomes a highly problematic method for small samples, worse even than standard Bayes. The reason is its dependence on the specified σ^2 as can be clearly seen from (3.23). If σ^2 was set to be much larger than the actual average prediction error on the sample, then the third term in (3.23) dominates. This term decreases with η and thus automatically pushes $\hat{\eta}$ 'upward' by an arbitrary amount. The term also decreases with n, so that the problem disappears at a large enough sample size. The problem did not occur in the model averaging experiment; we suspect that this is because in this experiment, there is substantial prior mass on a small model

[1] In the present setting, learning η by empirical Bayes has a second interpretation: if one fixes the variance σ^2 appearing in the prior on β, uses the linear model with a different variance σ'^2, and then learns σ'^2 by empirical Bayes, the result is identical to fixing $\sigma'^2 = \sigma^2$ and learning η by empirical Bayes.

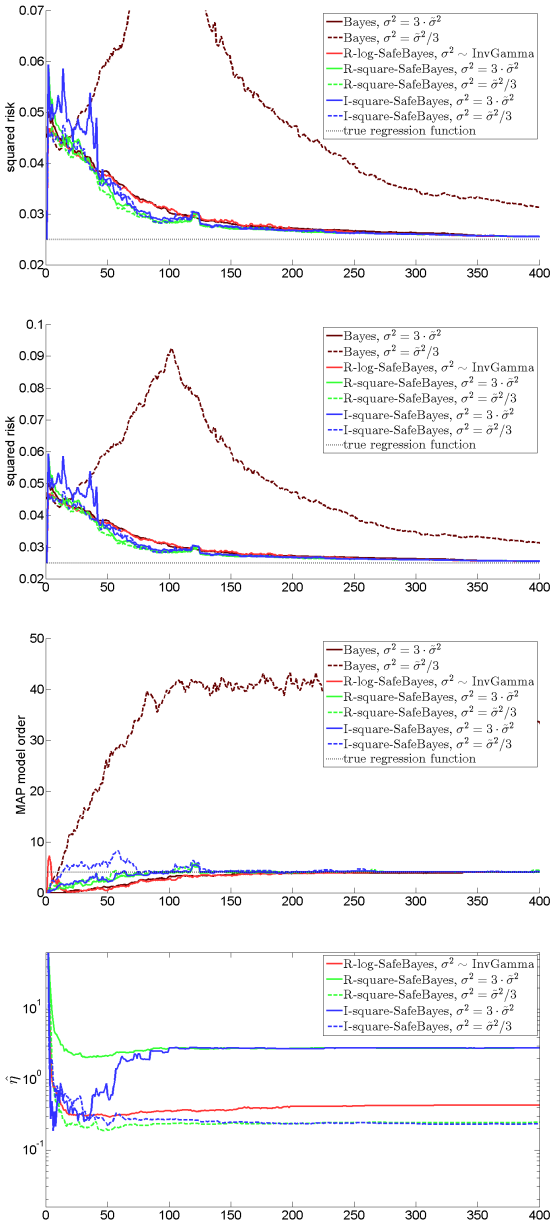


Figure 5.1: Bayesian model averaging, fixed σ^2, for the model-wrong experiment of Figure 3.3 with $p_{\max} = 50$. The second graph is a scaled version of the first. Since fixed σ^2 implies fixed self-confidence ratio, the self-confidence graph is not shown. For clarity in the η-graph we do not show standard deviations of the η's.

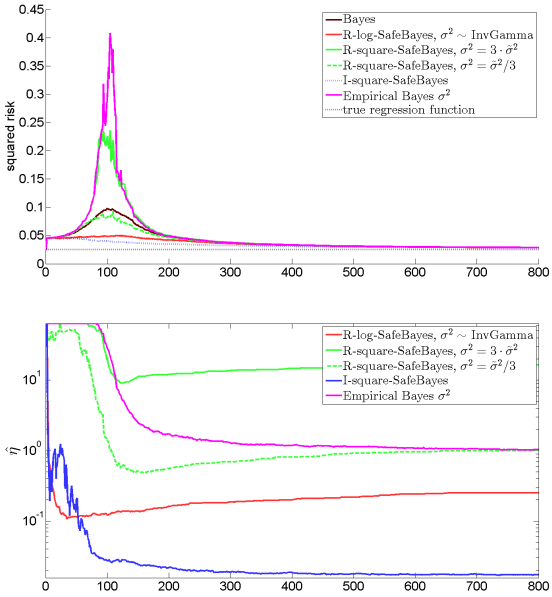


Figure 5.2: Bayesian ridge regression: Same graphs as in Figure 3.7, for fixed σ^2 and the model-wrong experiment conditioned on $p := p_{\max} = 50$. Note the difference in scale for the risk in this figure and Figure 5.1.

($p = 4$) containing the pseudo-truth, and for this submodel, the final term in (3.23) (which is approximately linear in p) is much smaller than for $p = 50$ and does have not such a strong influence.

5.1.2 Slightly informative prior

Again we only consider model-wrong experiments. Within each model, we now use the following prior parameters: $\bar{\beta}_0 = \mathbf{0}$ and $\Sigma_0 = 10^3\mathbf{I}$ for the multivariate normal distribution on β; and $a_0 = 1$ and $b_0 = \sigma^{*2}a_0$ (as before) for the inverse gamma distribution on σ^2 (where σ^{*2} is the true marginal variance of noise in our data, as defined in Section 3.5.1.2). We repeated the model-wrong experiment of Section 3.5.3 with $p_{\max} = 50$ with this slightly informative prior and obtained similar results to those obtained using our original informative prior with $\Sigma_0 = \mathbf{I}$: Bayes performs badly roughly between samples 90 and 130 and has some risk spikes before that so that its overall performance is comparable to before, while R-log-SafeBayes and I-log-SafeBayes both obtain good risks. We omit the pictures as they show no surprises.

We also repeated the model-wrong experiment for ridge regression (Section 3.5.4). Here the effect of the new prior on Bayes' performance is similar: the square-risk now peaks at a larger value, but in a smaller range of sample sizes. However, the effect of changing the learning rate is different in this ex-

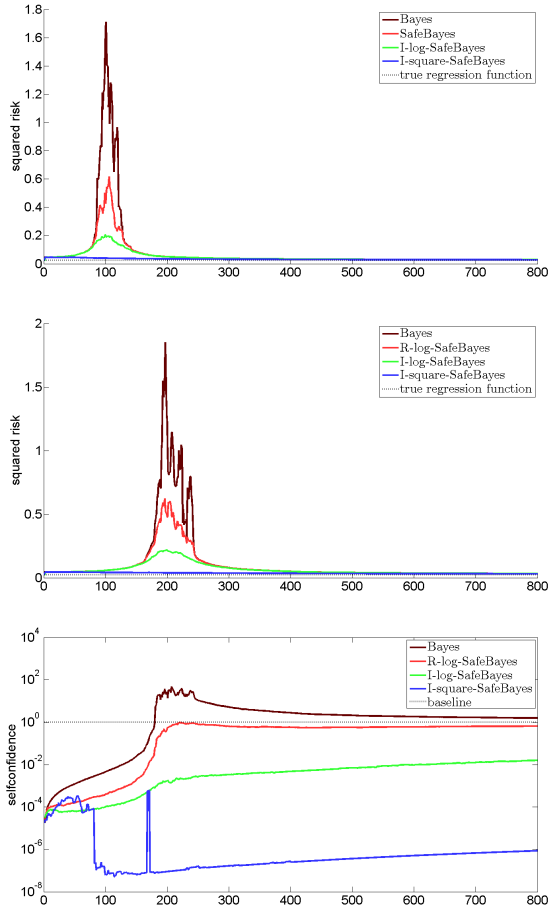


Figure 5.3: Top two graphs: square-risk for two different ridge experiments. In both experiments the slightly informative prior of Section 5.1.2 is used. In the first experiment $p = 50$, in the second $p = 100$; otherwise the experiments are just as the 'wrong model experiment' of Section 3.5.4, Figure 3.7, but we also included performance of I-square-SafeBayes. Final graph shows self-confidence for the $p = 100$ case for Bayes and SafeBayes, on a logarithmic scale because of the range of values involved.

periment than what we have seen before: here one can take η *very* small and still get good results. So in a sense, the problematic behaviour of Bayes has a trivial solution here: just pick a very small but fixed η. R-log-SafeBayes was too conservative in this, I-log-SafeBayes did fine. R-log-SafeBayes became competitive again however, if we used the discounting version described in Section 5.2.1 below.

In Figure 5.3 we repeat the pictures for ridge regression (Section 3.5.4) with

this slightly informative prior, because they give additional insight. Note that the phenomenon is now much more 'temporary'. In the beginning, it seems that there is a sort of cancellation between the influence of the irrelevant variables and standard Bayes behaves fine. However, if we increase the number of irrelevant variables, the problem (while starting at a later sample) takes longer to recover from.

5.1.3 Prior as advised by Raftery et al.

In Raftery et al. (1997), some guidelines for choosing priors in regression models are given. Letting $\bar{\beta}_0$ denote the prior mean, one of their recommendations is that the prior densities for $\beta = \bar{\beta}_0$ and $\beta = \bar{\beta}_0 + \mathbf{1}$ should differ by a factor of at most $\sqrt{10}$. The prior density on β marginalized over σ^2 follows a multivariate t-distribution, and the factor in question varies with the dimensionality of β, so that models of larger order are given less informative priors. In our case, we find that the resulting prior is always less informative than our original prior, and for model $\mathcal{M}_{10}$ and above (i.e. β of dimension 11 or larger), it becomes even less informative than the prior introduced in the previous section.

For the prior on σ^2, Raftery et al. advise that the density should vary by no more than a factor 10 in a region of σ^2 from some small value to the sample variance of y. For our choice of hyperparameters $a_0 = 1$, $b_0 = 1/40$, the mode of $\pi(\sigma^2)$ is at $b_0/(a_0 + 1) = 1/80$, and the density is within a factor 10 of this maximum in the approximate region $(0.0037, 0.0941)$. For the correct model experiments, the actual variance of Y is 0.065; for the wrong model experiments, it is 0.045 (with a larger variance for 'good' points and zero variance for 'easy' points). For both experiments, the factor-10 condition holds between $\mathrm{Var}(Y)/12$ and $\mathrm{Var}(Y)$. We conclude that this prior satisfies the guidelines in Raftery et al. quite well.

We will refer to the prior described above as Raftery's prior (even though it is really a different prior for each model order). Using this prior, we found the following experimental results.

In the model-wrong setting of Section 3.5.3 (model selection/averaging), with our original prior replaced by Raftery's prior, Bayes performs somewhat *better* than R-log-SafeBayes (except on very small sample sizes). However, I-log-SafeBayes performs as well as Bayes, and so does the R-log-SafeBayes variant that discounts half of the initial sample when choosing the learning rate (see Section 5.2.1).

This might suggest that Raftery's prior could be used to accomplish the same kind of safety against wrong models as SafeBayes provides, at least in a model selection context. To test this, another experiment was performed where the fraction of 'easy' points was increased to 75%. In this experiment, the misbehaviour of Bayes seen in Section 3.5.3 returned worse than before, with risks a factor 20 larger than before, whereas the SafeBayes methods continued to work fine. This suggests that Raftery's prior can not be relied on if the severeness of misspecification is unknown.

If Raftery's prior is used for model selection with a correct model, Bayes and the SafeBayes variants perform well, and very similarly to each other.

For ridge regression, the results with Raftery's prior for both the correct and the incorrect model experiment are very similar to those with the slightly informative prior, except that the peak in the risks is higher for all methods.

5.1.4 The g-prior

Another prior we experimented with was the g-prior, which is a popular choice in model selection contexts (Zellner, 1986; Liang et al., 2008). For all definitions we refer to the latter paper. In contrast to all other priors we considered, the g-prior depends on the design matrix $\mathbf{X}_n$, and hence can only be used in settings where this matrix, and hence the eventual sample size of interest n, is given once and for all. For this reason, we decided to depict in Figure 5.4, for each value of n, the risk obtained when predicting the $(n+1)$-th data point with the posterior calculated from the g-prior corresponding to the first n covariates $(x_1, \ldots, x_n)$ and observed data y^n. This is subtly different from our previous graphs (e.g. Figures 3.3–3.6) that show how the risk evolves as n increases in a *single* run of the experiment, averaged over 30 runs.

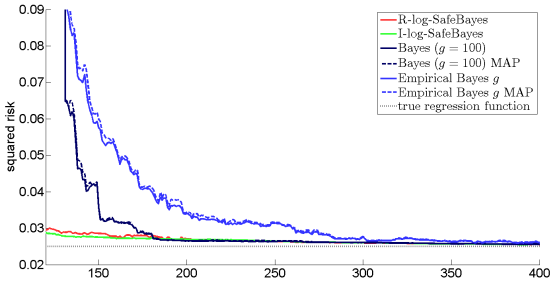


Figure 5.4: Risk as a function of sample size (starting at the first sample size at which the g-prior is defined) for model averaging and selection based on the g-prior in the model-wrong experiment of Figure 3.3 both with $g = 100$ and with g chosen by Empirical Bayes at each sample size

The graph is not shown starting at $n = 0$, because of another difference between the g-prior and the priors we used in other experiments:

Because of the same design dependence, with the g-prior, the posterior on β remains a degenerate distribution on an initial segment of outcomes. For example, with $\mathcal{M}_p$ for $p = 50$, the matrix $\mathbf{X}_n^\top \mathbf{X}_n$ is singular until at least 50 *different* design vectors have been observed. For our model-wrong experiment, this means that on average, about a 100 observations are required before the posterior becomes nondegenerate; this explains why Figure 5.4 starts at n a little over a 100.

The experimental results clearly indicate that the g-prior does not deal with our data in a satisfactory way, regardless of the value of g. Of the values of g

we tried (up to 10^4), $g \approx 100$ (shown in the graph) yielded the smallest square-risk around sample size $n = 200$; for larger sample sizes, larger values of g were better, but only slightly. Furthermore (as in fact we expected by analogy to learning η with Empirical Bayes), the value of g found by Empirical Bayes is not optimal for dealing with our data and only makes things worse: larger values of g (which put more weight on the data) would yield smaller risks.

5.2 Experiments on variations on the method

Below we look at a number of other more or less promising alternative approaches to modifying standard Bayes.

5.2.1 An idea to be explored further: Discounting initial observations

Just like standard Bayes, all our SafeBayesian methods are, at heart, *prequential* (Dawid, 1984). All prequential methods suffer to a greater or lesser extent from the *start-up problem* (Van Erven et al., 2007; Wong and Clarke, 2004): sequential predictions based on a model $\mathcal{M}_p$ may perform very badly for the first few samples. While they quickly recover when the sample size gets large, the behaviour on the first few samples may dominate their cumulative prediction error for a while, leading to suboptimal choices for moderate n. We can address this issue in several ways. A very simple method to 'discount' initial observations, apparently first used (implicitly) to modify standard Bayes factors by Lempers (1971, Chapter 6), is to only look at the cumulative sequential prediction error on the second half of the sample, so that the first half of the sample merely functions as a 'warming-up' sample (Catoni, 2012). Without claiming that this is the 'right' method to discount initial observations, we experimented with it to see whether it can further improve the performance of SafeBayes; for simplicity, we concentrated on R-log-SafeBayes.

We found that in most experiments, this new method for determining η performed very similarly to the standard method based on the whole sample, sometimes slightly better and sometimes slightly worse, making it hard to say whether the new method is an improvement or not. Still, there are two experiments in which the new method performed substantially better, namely the experiments with less informative priors of Section 5.1.2 and 5.1.3. Thus we cannot just dismiss the idea of fitting η based on only part of the data or more generally, discounting initial observations, and it would be interesting to explore this further in future work: of course taking half of the data is rather arbitrary, and better choices may be possible. In particular, we may try a variation of *switching* between η's analogously to the switch distribution (Van Erven et al., 2007) to counter the start-up problem.

5.2.2 Other methods for model selection: AIC, BIC, (generalized) cross-validation

We tested the performance of several classic model selection methods on the same data and models as in our main model selection/averaging experiment, Section 3.5.3. We associated with each model $\mathcal{M}_p$ its standard (i.e. $\eta = 1$) Bayes predictive distribution under the prior described in Section 3.5.1 (these generally perform better than the maximum likelihood distributions based on $\mathcal{M}_p$ whose use is more standard here). We then ran leave-one-out cross-validation, 10-fold cross-validation and GCV based on the predictions (posterior means/MAPs $\bar{\beta}_{i,\eta}$) made by these predictive distributions. We also compared the models via AIC and BIC, where for AIC we used the small-sample correction of Hurvich and Tsai (1989).

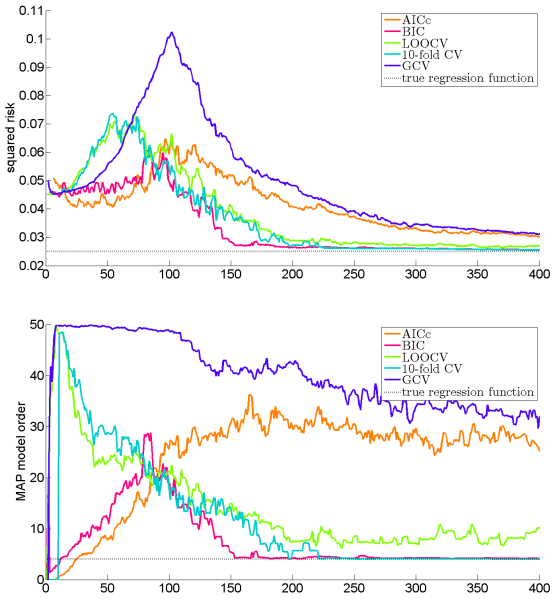


Figure 5.5: Square-risk and selected model order for five different model selection methods. The risks in this graph are risks of single models selected by each method (similar to the MAP risks shown for Bayes and SafeBayes).

We see in Figure 5.5 that AIC and generalized cross-validation have risks and selected model orders similar to those of standard Bayes, though they do not recover as well as Bayes when the sample size increases. Of the other three methods, BIC and 10-fold cross-validation find the optimal model and have smaller risks towards the end than leave-one-out cross-validation, which continues to select larger-than-optimal models with substantial probability. Note that none of the methods can compete with SafeBayes on sample sizes below 150: SafeBayes's risk goes down immediately after the start of the experiment

while for all the other methods it goes up first. Also, SafeBayes finds the optimal model quickly without first trying much larger models.

5.2.3 Other methods for learning η: Cross-validation on log-loss and on squared loss

As indicated in the introduction and Section 3.4.2, finding $\hat{\eta}$ by I-square-SafeBayes is somewhat similar to finding $\hat{\eta}$ by leave-one-out cross-validation with the squared-error loss, the difference being that I-square-SafeBayes finds the optimal η for predicting each point based on past data data points rather than the optimal η for predicting each point based on all other data points. Since the leave-one-out method is often employed in ridge regression, it seemed of interest to try out here as well. Figure 5.6 shows that LOO-cross validation indeed performs very similarly in terms of square-risk to R-log-SafeBayes (and to I-log- and I-square-SafeBayes, which are not depicted here but are similar to R-log-SafeBayes). However, LOO-cross validation is consistently a bit worse in terms of self-confidence; we do not have a clear explanation for this phenomenon.

Perhaps more interestingly, in Figure 5.7 we show what happens if we use LOO-cross validation based on the log-loss of the Bayes predictive distribution, which may seem a reasonable procedure from a 'likelihoodist' perspective. Here we see dismal behaviour, the reason being the hypercompression phenomenon of Section 4.1.3: cross-validation will select a model that, at the given sample size, has small log-risk, but because of hypercompression this model can sometimes perform very badly in terms of all the associated prediction tasks such as square-risk and reliability.

5.3 Experiments on variations of the truth

Other distributions of covariates In all experiments described in Section 3.5 and the earlier sections of this chapter, the covariates $(X_{i1}, X_{i2}, \ldots)$ were sampled independently from a 0-mean multivariate Gaussian. We repeated most of our experiments with $X_{i1}, X_{i2}, \ldots$ that were sampled independently uniformly from $[-1, 1]$, and, as already indicated in the introduction, with polynomials, $X_{ij} = P_j(S_i)$ for P_j the Legendre polynomial of degree j and $S_i \in [-1, 1]$ uniform. This did not change the results in any substantial way, so we do not report on it further.

Fewer easy and 'less-easy' points If the fraction of 'easy' points is reduced, one would expect the performance of standard Bayes to improve. This is confirmed by an experiment where each data point had a probability of only $1/4$ to be $(0, 0)$. Here Bayes still has some trouble finding the optimal model, but the square-risk, MAP model order, and time taken to recover are all much reduced compared to the original experiment in Section 3.5.3 where half the data points

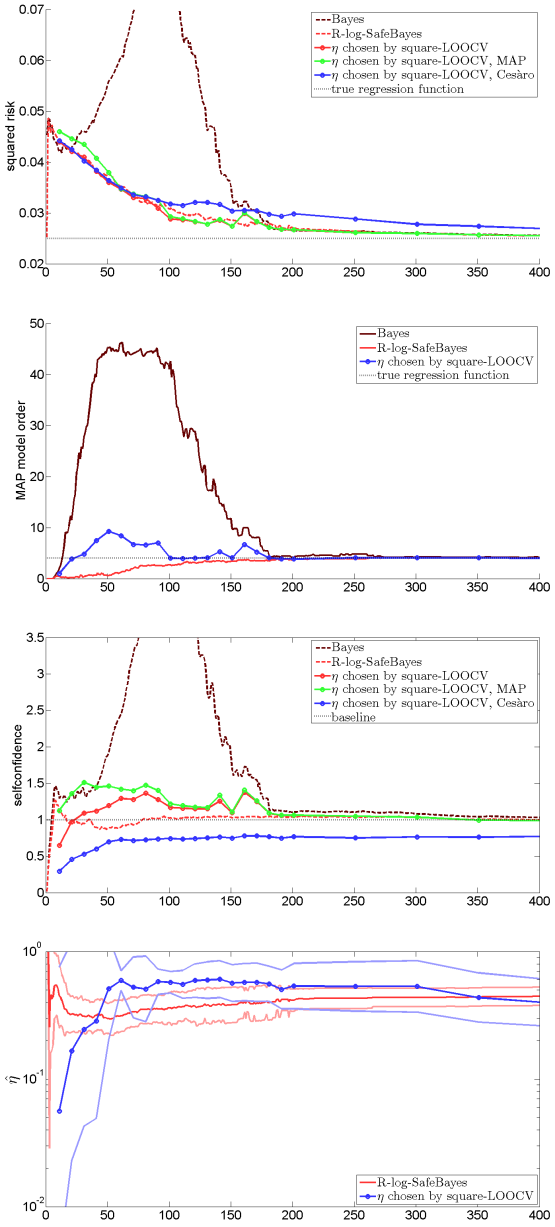


Figure 5.6: Analogue of Figure 3.3 for determining η by leave-one-out cross-validation with square-loss with the wrong-model experiment, $p_{\max} = 50$.

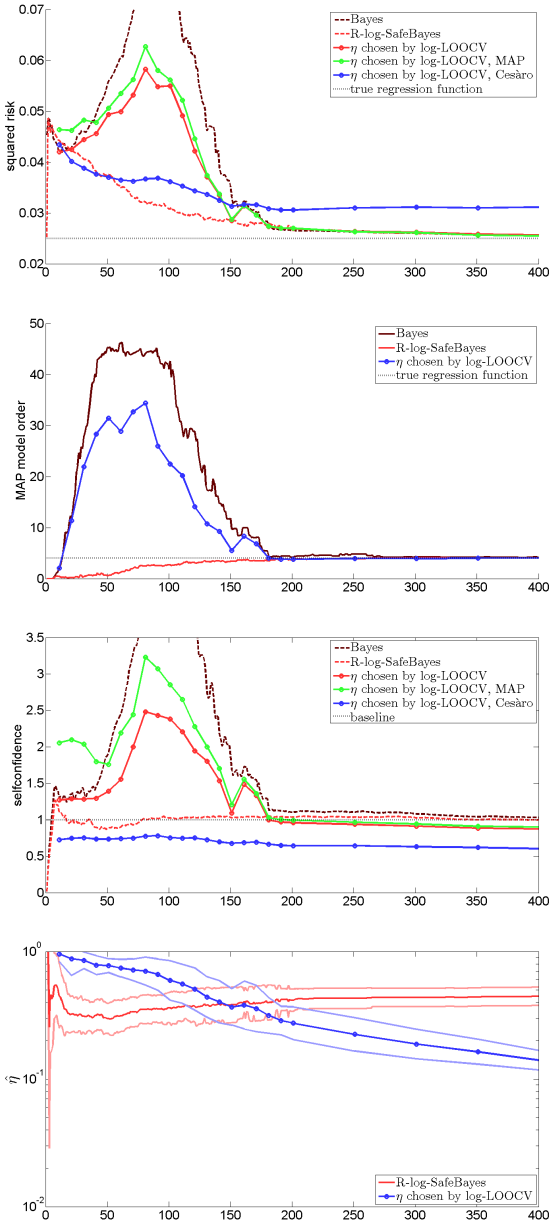


Figure 5.7: Analogue of Figure 3.3 for determining η by leave-one-out cross-validation with log-loss.

were 'easy'. SafeBayes on the other hand showed the same good performance as before.

Two points that might be raised against the use of 'easy' points in our simulations are that they are unlikely to occur in practice, and that if they were to occur, they would be easily detected and dealt with another way. To address this line of argument to some extent, another experiment was performed with a smaller contrast between 'easy' and 'hard' points. Rather than being identically $(0, 0)$, the 'easy' points were random but with smaller variance than the 'hard' points. To be precise, the covariates and noise were both a factor 5 smaller (so that their variances were 25 times smaller). In this experiment, the same phenomena as in Section 3.5.3 occurred, albeit again on a smaller scale (though larger than in the previous, $1/4$-easy experiment).

Different optimal regression functions We experimented with a number of variations of the wrong-model experiment of Section 3.5.3, by changing the underlying 'true' distribution P^*. In each variation, we still tossed, at each i, an independent biased coin to determine whether i would be 'easy' (still probability $1/2$) or 'regular' (probability $1/2$), but in each case we changed the definition of either the 'easy' or the 'regular' instances or both. In all experiments, for the 'regular' instances, only $P^*(Y_i \mid X_i)$ was changed; the marginal distribution of the X_i was still multivariate normal as before. Here is a list of things we tried:

1. For regular instances, set $P^*(Y_i \mid X_i)$ so that $Y_i = 0 + \epsilon_i$ instead of (3.27), with ϵ_i i.i.d. normal as before; easy instances were still set to $(\mathbf{0}, 0)$.

2. For regular instances, (3.27) was replaced by $Y_i = X_{i1} + X_{i2} + X_{i3} + X_{i4} + \epsilon_i$, so the optimal coefficients $\tilde{\beta}_1 \ldots \tilde{\beta}_4$ are ten times as large as in the original experiment; easy instances were still set to $(\mathbf{0}, 0)$.

3. For regular instances, (3.27) was replaced by $Y_i = .1 \cdot (X_{i1} + \ldots + X_{i4}) - .04 + \epsilon_i$ (so the intercept is not 0), and the easy instances were set to $(X_i, Y_i) = (.\mathbf{2}, .04)$, where $.\mathbf{2}$ represents the K-dimensional vector $(.2, \ldots, .2)$. Note that the easy points are on the optimal regression function.

4. For regular instances, (3.27) was replaced by $Y_i = .1 \cdot (X_{i1} + \ldots + X_{i4}) + .5 + \epsilon_i$ so the intercept was again not 0; the easy instances were set to $(\mathbf{0}, .5)$.

We explain each in turn. For the first experiment, all the results were comparable to the results of Experiment 1 in Section 3.5, so we do not list them. For the second experiment, the risks obtained by standard Bayes and SafeBayes were similar to each other. The model order behaviours were similar to what they were before (with standard Bayes selecting large model orders initially), but all methods recovered much more quickly, converging on the optimal model shortly after $n = 50$; presumably this could happen because now the optimal coefficients were substantially larger than the standard deviation in the data.

The third experiment was included to see whether there would be an effect if the 'easy' points would be placed at an arbitrary point rather than the special,

fully symmetric $(\mathbf{0}, 0)$. We added the intercept -0.04 so as to make sure that, for the data we actually observe, $\mathbf{E}_{X,Y\sim P^*}[Y_i] = (1/2) \cdot 0.04 - (1/2) \cdot 0.04 = 0$; thus the Y-values will appear centred around 0, which is standard both in frequentist and Bayesian approaches to regression (for example, both Raftery et al. (1997) and Hastie et al. (2001) preprocess the data so that $\sum_{i=1}^{n} Y_i = 0$). Again, we discerned no difference in the results so did not include any further details.

Finally, the fourth experiment was included just to see what happens if, contrary to standard methodology, we apply the method to Y_i that are *not* (even approximately) centred. In this experiment, standard Bayes did not converge to the optimal model until after $n = 150$ as in the experiment of Section 3.5.3, but its risk and selected model orders were both smaller. The versions of Safe-Bayes worked well as before.

Chapter 6

Worst-Case Optimal Probability Updating

The final three chapters of this dissertation discuss worst-case optimal probability updating, an alternative to conditioning that may be used when the distribution is not fully specified. In Chapter 6, we introduce the problem, and find how optimal solutions may be recognized for different loss functions; our main tool is convex analysis. We find that for logarithmic loss, optimality is characterized by the elegant *RCAR (reverse coarsening at random)* condition.

In Chapter 7, we analyse the combinatorial aspect of the probability updating problem, and present some theoretical tools that may help us find worst-case optimal solutions to a probability updating problem, as opposed to merely recognizing such solutions. Further, we see that the applicability of the RCAR condition is not restricted to the cases discovered in Chapter 6, and explore the consequences.

In Chapter 8, we give algorithms that automate the task of finding worst-case optimal solutions, for restricted classes of probability updating problems.

A more detailed overview of this first chapter will be provided in Section 6.1.2.

6.1 Introduction

There are many situations in which a decision maker receives incomplete data and still has to reach conclusions about these data. One type of incomplete data is *coarse* data: instead of the real outcome of a random event, the decision maker observes a subset of the possible outcomes, and knows only that the actual outcome is an element of this subset. An example frequently occurs in questionnaires, where people may be asked if their date of birth lies between 1950 and 1960 or between 1960 and 1970 et cetera. Their exact year of birth is unknown to us, but at least we now know for sure in which decade they

are born. We introduce another instance of coarse data with the following example.

Example 6.A (Fair die)**.** Suppose I throw a fair die. I get to see the result of the throw, but you do not. Now I tell you that the result lies in the set $\{1,2,3,4\}$. This is an example of coarse data. You know that I used a fair die and that what I tell you is true. Now you are asked to give the probability that I rolled a 3. Likely, you would say that the probability of each of the remaining possible results is $1/4$. This is the knee-jerk reaction of someone who studied probability theory, since this is standard *conditioning*. But is this always correct?

Suppose that there is only one alternative set of results I could give you after rolling the die, namely the set $\{3,4,5,6\}$. I can now follow a *coarsening mechanism*: a procedure that tells me which subset to reveal given a particular result of the die roll. If the outcome is 1, 2, 5, or 6, there is nothing for me to choose. Suppose that if the outcome is 3 or 4, the coarsening mechanism I use selects set $\{1,2,3,4\}$ or set $\{3,4,5,6\}$ at random, each with probability $1/2$. If I throw the die 6000 times, I expect to see the outcome 3 a thousand times. Therefore I expect to report the set $\{1,2,3,4\}$ five hundred times after I see the outcome 3. It is clear that I expect to report the set $\{1,2,3,4\}$ 3000 times in total. So for die rolls that I told you resulted in an outcome in $\{1,2,3,4\}$, the probability of the true outcome being 3 is actually $1/6$ with this coarsening mechanism. We see that the prediction in the first paragraph was not correct, in the sense that the probabilities computed there do not correspond to the long-run relative frequencies. We conclude that the knee-jerk reaction is not always correct.

In Example 6.A we have seen that standard conditioning does not always give the correct answers. Heitjan and Rubin (1991) answer the question under what circumstances standard conditioning of coarse data is correct. They discovered a necessary and sufficient condition of the coarsening mechanism, called *coarsening at random (CAR)*. A coarsening mechanism satisfies the CAR condition if, for each subset y of the outcomes, the probability of choosing to report y is the same no matter which outcome $x \in y$ is the true outcome. In many situations, however, this condition cannot hold, as proved by Grünwald and Halpern (2003). It depends on the arrangement of possible revealed subsets whether a coarsening mechanism exists that satisfies CAR.

We continue with an example generating much debate among both the general public and professional probabilists: the Monty Hall puzzle, posed by Selvin (1975) and popularized years later when it appeared in Ask Marilyn, a weekly column in Parade Magazine by Marilyn vos Savant, in September 1990 (Gill, 2011). In this example, no coarsening mechanism satisfies the CAR condition.

Example 6.B (Monty Hall)**.** Suppose you are on a game show and you may choose one of three doors. Behind one of the doors a car can be found, but the other two only hide a goat. Initially the car is equally likely to be behind each of the doors. After you have picked one of the doors, the host Monty Hall, who

knows the location of the prize, will open one of the other doors, revealing a goat. Now you are asked if you would like to switch from the door you chose to the other unopened door. Is it a good idea to switch?

At this moment we will not answer this question, but we show that the problem of choosing whether to switch doors is an example of the coarse data problem. The unknown random value we are interested in is the location of the car: one of the three doors. When the host opens a door different from the one you picked, revealing a goat, this is equivalent to reporting a subset. The subset he reports is the set of the two doors that are still closed. For example, if he opens door 2, this tells us that the true value, the location of the car, is in the subset $\{1, 3\}$. Note that if you have by chance picked the correct door, there are two possible doors Monty Hall can open, so also two subsets he can report. This implies that Monty has a choice in reporting a subset. How does Monty's coarsening mechanism influence your prediction of the true location of the car?

The CAR condition can only be satisfied for very particular distributions of where the prize is: the probability that the prize is hidden behind the initially chosen door must be either 0 or 1, otherwise no CAR coarsening mechanism exists (Grünwald and Halpern, 2003, Example 3.3)[1]. If the prize is hidden in any other way, for example uniformly at random as we assume, then CAR cannot hold, and standard conditioning will result in an incorrect conclusion for at least one of the two subsets.

Examples 6.A and 6.B are just two instances of a more general problem: the number of outcomes may be different; the initial distribution of the true outcome may be any distribution; and the subsets of outcomes that may be reported to the decision maker may be any family of sets. Our goal in this chapter is to define general procedures that tell us how to update the probabilities on the outcomes after making a coarse observation, in situations where standard conditioning is not adequate.

These probabilities may be either frequentist or Bayesian probabilities. In Example 6.A, they were frequentist probabilities: the original distributions were known, and the updated probabilities we found were again frequencies over many repetitions of the same experiment. The original distribution of the outcomes could also express our Bayesian prior belief of how likely each outcome is. This is the more powerful interpretation in Example 6.B, where the uniform distribution of the location of the car requires an assumption on the frequentist's part, while it is a reasonable choice of prior for a Bayesian (Gill, 2011). In this case, the updated probabilities after a coarse observation take the role of the Bayesian posterior distribution. In either case, we will refer to the initial probability of an outcome, regardless of observations, as the *marginal* probability.

Without any assumptions on the quizmaster's strategy (i.e. the coarsening mechanism), the conditional distributions of outcomes given observations will

[1] This uses the *weak* version of CAR in the terminology of Jaeger (2005a), in which outcomes with probability 0 are exempt from the equality constraint. A *strong* CAR coarsening mechanism does not exist regardless of the probabilities with which the prize is hidden.

be unknown, and this uncertainty cannot be fully expressed by a single probability distribution over the outcomes. So to get a single answer to the question how to update our probabilities, we need to make some assumption about how the quizmaster chooses his strategy. Assuming that the coarsening mechanism satisfies CAR is one such approach, but as we saw in the two examples, there are scenarios where this assumption cannot hold. We instead take a *worst-case approach*, treating the coarsening of the observation and the subsequent probability update as a game between two players: the *quizmaster* and the *contestant* (named for their roles in the Monty Hall scenario). The subset of outcomes communicated by the quizmaster to the contestant will be called the *message*.

In this fictional game, the quizmaster's goal is the opposite of the contestant's, namely to make predicting the true outcome as hard as possible for the contestant. Such situations are rare in practice: The sender of a message might be motivated by interests other than informing us (for example, a newspaper may be trying to optimize its sales figures, or a company may want to present its performance in the best light), but rarely by trying to be as uninformative as possible. (Though see Section 7.4.3 in the next chapter, where we consider the case that the players' goals are not diametrically opposed.) In other situations, the 'sender' might not be a rational being at all, but just some unknown process. Yet this game is a useful way to look at the problem of updating our probabilities even if we do not believe that the coarsening mechanism is chosen adversarially: if we simply do not know how 'nature' chooses which message to give us and do not want to make any assumptions about this, then choosing the worst-case (or *minimax*) optimal probability update as defined here guarantees that we get at most some fixed expected loss, while any other probability update may lead to a larger expected loss depending on the unknown coarsening mechanism.

We will need a loss function to measure how well the quizmaster and the contestant are doing at this game. Our results apply to a wide variety of loss functions. For an analysis of the Monty Hall game, 0-1 loss would be appropriate, as the contestant must choose a single door; this is the approach used in Gill (2011); Gnedin (2011). Other loss functions, such as logarithmic loss and Brier loss, also allow the contestant to formulate their prediction of where the prize is hidden as an arbitrary probability distribution over the outcomes. For the Monty Hall game with one of these two loss functions, the worst-case optimal answer for the contestant is to put probability $1/3$ on his initially chosen door and $2/3$ on the other door. (These probabilities agree with the literature on the Monty Hall game.) Surprisingly, we will see (in Example 6.D on page 129) that for similar games, logarithmic and Brier loss may lead to two different answers!

We will find in this chapter that for finite outcome spaces, both players in our game have worst-case optimal strategies for many loss functions: the quizmaster has a strategy that makes the contestant's prediction task as hard as possible, and the contestant has a strategy that is guaranteed to give good predictions no matter what the quizmaster does. We give characterizations that allow us to recognize such strategies, for different conditions on the loss

functions. Interestingly, Theorem 6.10 shows that the worst-case optimal prediction under logarithmic loss satisfies a property that has a striking similarity with the CAR condition, but switches the roles of outcome and message. By Lemma 6.14, if a betting game is played repeatedly and the contestant is allowed to distribute investments over different outcomes and to reinvest all capital gained so far in each round, then the same strategy is optimal, *regardless of the pay-offs!*

Example 6.A (continued). For logarithmic loss, the worst-case optimal prediction of the die roll conditional on the revealed subset is found with the help of Theorem 6.10. The worst-case optimal prediction given that you observe the set $\{1,2,3,4\}$ is: predict outcomes 1 and 2 each with probability $1/3$, and predict 3 and 4 each with probability $1/6$. Given that you observe the set $\{3,4,5,6\}$, the worst-case optimal prediction is: 3 and 4 with probability $1/6$, and 5 and 6 with probability $1/3$.

These probabilities correspond with the coarsening mechanism given earlier. However, it is a good prediction even if you do not know what coarsening mechanism I am using. An intuitive argument for this is the following: If I wanted, I could use a very extreme coarsening mechanism, always choosing to reveal the set $\{1,2,3,4\}$ when the die comes up 3 or 4. But this is balanced by the possibility that I might be using the opposite coarsening mechanism, which always reveals $\{3,4,5,6\}$ if the result is 3 or 4. The worst-case optimal prediction given above hedges against both possibilities.

6.1.1 Caveats on the use of the word 'conditioning'

Since this chapter is concerned with making a worst-case optimal prediction conditional on a set of outcomes, we want to highlight the use of the word *conditioning*. Above, we used the word *standard* conditioning for using the conditional information in the standard way: with random variables X the true outcome and Y the coarse observation (a set of outcomes), computing $P(X = x \mid Y = y)$ by $P(X = x \mid X \in y) = P(X = x)/P(X \in y)$.

In the two examples, we saw that this does not always give the correct probabilities given a coarse observation. For instance in Example 6.A, $P(X \mid X \in y)$ is the uniform distribution, while $P(X \mid Y = y)$ cannot simultaneously be uniform for both $y = \{1,2,3,4\}$ and $y = \{3,4,5,6\}$. In such situations, we call this computation *naive conditioning*. The formula for conditional probabilities does give the correct result if instead of on the outcome space, we work in a larger space: the space of all combinations of an outcome and a set. The problem we face in this chapter is that we do not know the probabilities of all these combinations, as they depend on the unknown coarsening mechanism.

In the case of Bayesian probabilities, the question is how to extend our prior on the outcomes to a prior on the larger space. The worst-case optimal coarsening mechanism we characterize in our theorems can be seen as a recommendation for such a prior.

6.1.2 Contents

In Section 6.2, we will give a precise definition of the 'conditioning game' we described. In Section 6.3, we find general conditions on the loss function under which worst-case optimal strategies for the quizmaster exist, and we characterize such strategies. Section 6.4 does the same for worst-case optimal strategies for the contestant. (See Figure 6.4 for a visual illustration of the concepts used in these sections.) If stronger conditions hold, worst-case optimal strategies for both players may be easier to recognize. This is explored for two classes of loss functions in Section 6.5; in particular, we find in Section 6.5.2 that for a class of loss functions including logarithmic loss, worst-case optimal strategies for the quizmaster are characterized by a simple condition on their probabilities: the *RCAR (reverse CAR)* condition. An overview of the theorems and the conditions under which they apply is given in Table 6.1 on page 125. Many examples are included to illustrate (the limits of) the theoretical results. Section 6.6 gives some concluding remarks.

All proofs are given in Appendix 6.A at the end of this chapter.

This work is an extension of Feenstra (2012) to loss functions other than logarithmic loss, and to the case where the worst-case optimal strategy for the quizmaster assigns probability 0 to some combinations of outcomes x and messages y with $x \in y$. It can also be seen as a concrete application of the ideas in Grünwald and Dawid (2004) about minimax optimal decision making and its relation to entropy.

6.2 Definitions and problem formulation

A *(probability updating) game* $\mathcal{G}$ is defined as a quadruple $(\mathcal{X}, \mathcal{Y}, p, L)$, where $\mathcal{X}$ is a finite set, $\mathcal{Y}$ is a family of distinct subsets of $\mathcal{X}$ with $\bigcup_{y \in \mathcal{Y}} y = \mathcal{X}$, p is a nowhere-zero probability mass function on $\mathcal{X}$, and L is a function $L : \mathcal{X} \times \Delta_{\mathcal{X}} \to [0, \infty]$, where $\Delta_{\mathcal{X}}$ is the set of all probability mass functions on $\mathcal{X}$. We call $\mathcal{X}$ the *outcome space*, $\mathcal{Y}$ the *message structure*, p the *marginal distribution*, and L the *loss function*. We shall discuss choices we made in this definition in Section 6.2.3, and first work out the definition in detail.

Example 6.B (continued). We assume the car is hidden uniformly at random behind one of the three doors. With this assumption, we can abstract away the initial choice of a door by the contestant: by symmetry, we can assume without loss of generality that he always picks door 2. Then the probability updating game starts with the quizmaster opening door 1 or 3, thereby giving the message "the car is behind door 2 or 3" or "the car is behind door 1 or 2", respectively. This can be expressed as follows in our formalization:

- outcome space $\mathcal{X} = \{x_1, x_2, x_3\}$;
- message space $\mathcal{Y} = \{y_1, y_2\}$ with $y_1 = \{x_1, x_2\}$ and $y_2 = \{x_2, x_3\}$;
- marginal distribution p uniform on $\mathcal{X}$.

If a loss function L is also given, this fully specifies a game. One example is randomized 0-1 loss, which is given by $L(x, Q) = 1 - Q(x)$. Here x is the true outcome, and Q is the contestant's prediction of the true outcome in the form of a probability distribution. Thus the prediction is awarded a smaller loss if it assigned a larger probability $Q(x)$ to the outcome that actually obtained. We will see other examples of loss functions in Section 6.2.2.

A function from some finite set S to $\mathbf{R}$ corresponds to an $|S|$-dimensional vector when we fix an order on the elements of S. We write $\mathbf{R}^S$ for the set of such functions/vectors. Even if no order on S is specified, this allows us to apply concepts from linear algebra to $\mathbf{R}^S$ without ambiguity. For example, we may say that some set is an affine subspace of $\mathbf{R}^S$. (This identification and the resulting notation are also used by Schrijver (2003a).)

Using this correspondence, we identify the elements of $\Delta_{\mathcal{X}}$ with the $|\mathcal{X}|$-dimensional vectors in the unit simplex, though we use ordinary function notation $P(x)$ for its elements. The probability mass function p that is part of a game's definition is also a vector in $\Delta_{\mathcal{X}}$. Vector notation p_x will be used to refer to its elements to set p apart from P, which will denote distributions chosen by the quizmaster rather than fixed by the game.

For any message $y \subseteq \mathcal{X}$, we define $\Delta_y = \{P \in \Delta_{\mathcal{X}} \mid P(x) = 0 \text{ for } x \notin y\}$. Note that these are vectors of the same length as those in $\Delta_{\mathcal{X}}$, though contained within a lower-dimensional affine subspace.

A loss function L is called *proper* if $\arg\min_{Q \in \Delta_{\mathcal{X}}} \mathbf{E}_{X \sim P} L(X, Q) = P$ for all $P \in \Delta_{\mathcal{X}}$, and *strictly proper* if this minimizer is unique (this is standard terminology; see for instance Gneiting and Raftery (2007)). Thus if a predicting agent believes the true distribution of an outcome to be given by some P, such a loss function will encourage him to report $Q = P$ as his prediction.

6.2.1 Strategies

Strategies for the players are specified by conditional distributions: a strategy P for the quizmaster consists of distributions on $\mathcal{Y}$, one for each possible $x \in \mathcal{X}$, and a strategy Q for the contestant consists of distributions on $\mathcal{X}$, one for each possible $y \in \mathcal{Y}$. These strategies define how the two players act in any situation: the quizmaster's strategy defines how he chooses a message containing the true outcome (the coarsening mechanism), and the contestant's strategy defines his prediction for each message he might receive.

We write $P(\cdot \mid x)$ for the distribution on $\mathcal{Y}$ the quizmaster plays when the true outcome is $x \in \mathcal{X}$. Because $p_x > 0$, this conditional distribution can be recovered from the joint $P(x, y) := P(y \mid x)p_x$; we will use this joint distribution to specify a strategy for the quizmaster. If $P(y) := \sum_{x \in y} P(x, y) > 0$, we may also write $P(\cdot \mid y)$ for the vector in Δ_y given by $P(x \mid y) := P(x, y)/P(y)$. No such rewrites can be made for Q, as no marginal $Q(y)$ is specified by the game or by the strategy Q. To shorten notation and to emphasize that Q is not a joint distribution, we write $Q_{|y}$ rather than $Q(\cdot \mid y)$ for the distribution that the contestant plays in response to message y.

We restrict the quizmaster to conditional distributions P for which $P(y \mid x) = 0$ if $x \notin y$; that is, he may not 'lie' to the contestant. We make no similar requirement on the contestant's choice of Q, though for proper loss functions, and in fact all other loss functions we will consider in our examples, the contestant can gain nothing from using a strategy Q for which $Q_{|y}(x) > 0$ where $x \notin y$.

Example 6.B (continued). The table below specifies all aspects of a game except for its loss function: its outcome space (here, for the Monty Hall game, $\mathcal{X} = \{x_1, x_2, x_3\}$), message space ($\mathcal{Y} = \{y_1, y_2\}$ with $y_1 = \{x_1, x_2\}$ and $y_2 = \{x_2, x_3\}$) and marginal distribution (p uniform on $\mathcal{X}$).

P	x_1	x_2	x_3
y_1	$1/3$	$1/6$	$-$
y_2	$-$	$1/6$	$1/3$
p_x	$1/3$	$1/3$	$1/3$

(6.1)

In this table we have filled in a strategy P for the quizmaster in the form of a joint distribution on pairs of x and y. The cells in the table where $x \notin y$ are marked with a dash to indicate that P may not assign positive probability there. The probabilities in each column sum to the marginal probabilities at the bottom, so this joint distribution P has the correct marginal distribution on the outcomes. For this particular strategy, if the true outcome is x_2, the quizmaster will give message y_1 or y_2 to the contestant with equal probability.

More formally, write $\mathcal{R}(\mathcal{X}, \mathcal{Y})$ as an abbreviation for the set of pairs $\{(x, y) \mid x \in y \in \mathcal{Y}\}$. In the case of the Monty Hall game, there are four such pairs: $\mathcal{R}(\mathcal{X}, \mathcal{Y}) = \{(x_1, y_1), (x_2, y_1), (x_2, y_2), (x_3, y_2)\}$. The notation $\mathbf{R}_{\geq 0}^{\mathcal{R}(\mathcal{X},\mathcal{Y})}$ represents the set of all functions from $\mathcal{R}(\mathcal{X}, \mathcal{Y})$ to $\mathbf{R}_{\geq 0}$. If P is an element of this set and $(x, y) \in \mathcal{R}(\mathcal{X}, \mathcal{Y})$, the value of P at (x, y) is denoted by $P(x, y)$. For (x, y) with $x \notin y$, the notation $P(x, y)$ does not correspond to a value of the function, but is taken to be 0.

We again identify the elements of $\mathbf{R}_{\geq 0}^{\mathcal{R}(\mathcal{X},\mathcal{Y})}$ with vectors. Thus the mass function P shown in (6.1) is identified with a four-element vector $(1/3, 1/6, 1/6, 1/3)$. (We could have chosen a different ordering instead.)

We define the set $\mathcal{P}$ of strategies for the quizmaster as $\{P \in \mathbf{R}_{\geq 0}^{\{x \in y\}} \mid \sum_{y \ni x} P(x, y) = p_x \text{ for all } x\}$; this is a convex set. The set of strategies for the contestant is $\mathcal{Q} \coloneqq \Delta_{\mathcal{X}}^{\mathcal{Y}} = \{(Q_{|y})_{y \in \mathcal{Y}} \mid Q_{|y} \in \Delta_{\mathcal{X}} \text{ for each } y \in \mathcal{Y}\}$.

For given strategies P and Q, the expected loss the contestant incurs is

$$\sum_{x \in \mathcal{X}} p_x \sum_{\substack{y: \\ x \in y \in \mathcal{Y}}} P(y \mid x) L(x, Q_{|y}) = \mathbf{E}_{X \sim p} \, \mathbf{E}_{Y \sim P(\cdot|X)} \, L(X, Q_{|Y}) = \mathbf{E}_{(X,Y) \sim P} \, L(X, Q_{|Y}). \quad (6.2)$$

We allowed L to take the value ∞; if this value occurs with positive probability, then the contestant's expected loss is infinite. However, for terms where the

probability is zero, we define $0 \cdot \infty = 0$, as is consistent with measure-theoretic probability.

We approach this problem as a zero-sum game between two players: the quizmaster chooses $P \in \mathcal{P}$ to maximize (6.2), while simultaneously (that is, without knowing P) the contestant chooses $Q \in \mathcal{Q}$ to minimize that quantity. The game $(\mathcal{X}, \mathcal{Y}, p, L)$ is common knowledge for the two players.

If the contestant knew the quizmaster's strategy, he would pick a strategy Q that for each y minimizes the expected loss of predicting x given y. When the contestant receives a message and knows the distribution $P \in \Delta_{\mathcal{X}}$ over the outcomes given that message, this expected loss is written as

$$H_L(P) := \inf_{Q \in \Delta_{\mathcal{X}}} \sum_x P(x) L(x, Q) = \inf_{Q \in \Delta_{\mathcal{X}}} \mathbf{E}_{X \sim P} L(X, Q). \tag{6.3}$$

This is the *generalized entropy* of P for loss function L (Grünwald and Dawid, 2004). (Note that in the preceding display, P and Q are not strategies but simply distributions over $\mathcal{X}$.) If the contestant picks his strategy Q this way, (6.2) becomes the *expected generalized entropy* of the quizmaster's strategy $P \in \mathcal{P}$:

$$\sum_{y \in \mathcal{Y}} P(y) H_L(P(\cdot \mid y)), \tag{6.4}$$

where we again define terms with $P(y) = 0$ as 0. We say a strategy P is *worst-case optimal for the quizmaster* if it maximizes this expected generalized entropy over all $P \in \mathcal{P}$. We call the version of the game where the quizmaster has to play first the *maximin* game, where the order of the words 'max' and 'min' reflects the order in which they appear in the expression for the value of this game as well as the order in which the maximizing and minimizing players take their turns.

Similarly, if the contestant were to play first (the *minimax* game), his goal might be to find a strategy Q that minimizes his worst-case expected loss

$$\max_{P \in \mathcal{P}} \sum_{x \in y} P(x, y) L(x, Q_{|y}) = \max_{P \in \mathcal{P}} \mathbf{E}_{(X,Y) \sim P} L(X, Q_{|Y}). \tag{6.5}$$

(In this case, the maximum is always achieved so we can write max rather than sup: for each x, the quizmaster can choose P that puts all mass on a $y \ni x$ with the maximum loss.) We call a strategy *worst-case optimal for the contestant* if it achieves the minimum of (6.5).

It is an elementary result from game theory that if worst-case optimal strategies P^* and Q^* exist for the two players, their expected losses are related by

$$\sum_{y \in \mathcal{Y}} P^*(y) H_L(P^*(\cdot \mid y)) \leq \max_{P \in \mathcal{P}} \sum_{x \in y} P(x, y) L(x, Q^*_{|y}) \tag{6.6}$$

(Rockafellar, 1970, Lemma 36.1: "maximin $\leq$ minimax"). The inequality expresses that in a sequential game where one of the players knows the other's strategy before choosing his own, the player to move second may have an advantage.

In the next section, we will see that in many probability updating games, worst-case optimal strategies for both players exist (but may not be unique), and the maximum expected generalized entropy *equals* the minimum worst-case expected loss:

$$\sum_{y\in\mathcal{Y}} P^*(y)H_L(P^*(\cdot \mid y)) = \max_{P\in\mathcal{P}} \sum_{x\in y} P(x,y)L(x,Q^*_{|y}). \tag{6.7}$$

When this is the case, we say that the minimax theorem holds (von Neumann, 1928; Ferguson, 1967). We remark here that our setting, while a zero-sum game, differs from the usual setting of zero-sum games in some respects: We consider possibly infinite loss and (in general) infinite sets of strategies available to the players, but do not allow the players to randomize over these strategies. Randomizing over $\mathcal{P}$ would not give the quizmaster an advantage, as $\mathcal{P}$ is convex and he could just play the corresponding convex combination directly; because (6.2) is linear in P, this results in the same expected loss. (Another way to view this is that, essentially, the quizmaster *is* randomizing, over a finite set of strategies.) For the contestant, $\mathcal{Q}$ is also convex, but in general (depending on L), playing a convex combination of strategies does not correspond to randomizing over those strategies. The two do correspond in the case of randomized 0-1 loss, where L is linear. If L is convex, then playing the convex combination is at least as good for him as randomizing (and if L is strictly convex, better), so allowing randomization would again not give an advantage.

When (6.7) holds, any pair of worst-case optimal strategies (P^*, Q^*) forms a *(pure strategy) Nash equilibrium*, a concept introduced by Nash (1951): neither player can benefit from deviating from their worst-case optimal strategy if the other player leaves his strategy unchanged. This means that the definitions of worst-case optimality given above are also meaningful in the game we are actually interested in, where the players move simultaneously in the sense that neither knows the other's strategy when choosing his own.

6.2.2 Three standard loss functions

Three commonly used loss functions are logarithmic loss, Brier loss, and randomized 0-1 loss. These are defined as follows (Grünwald and Dawid, 2004):

Logarithmic loss is a strictly proper loss function, given by

$$L(x,Q) = -\log Q(x).$$

Its entropy is the Shannon entropy $H_L(P) = \sum_x -P(x)\log P(x)$. The functions L and H_L are displayed in Figure 6.1 for the case of a binary prediction (i.e. a prediction between two possible outcomes). The (three-dimensional) graph of H_L for the case of three outcomes will appear in Figure 6.4 on page 128.

Brier loss is another strictly proper loss function, corresponding to squared Euclidean distance:

$$L(x,Q) = \sum_{x'\in\mathcal{X}} \left(\mathbf{1}_{x'=x} - Q(x')\right)^2 = (1-Q(x))^2 + \sum_{x'\in\mathcal{X},x'\neq x} Q(x')^2.$$

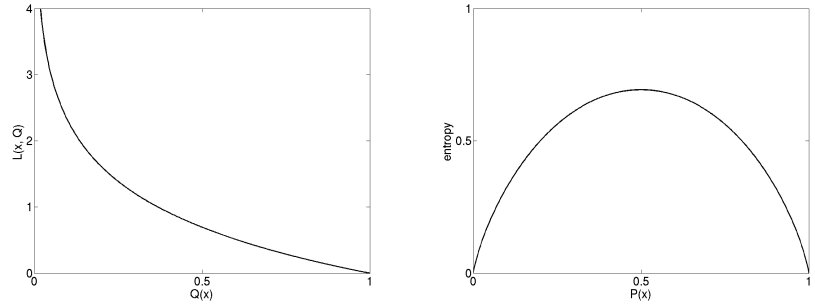


Figure 6.1: Logarithmic loss and entropy (natural base) on a binary prediction

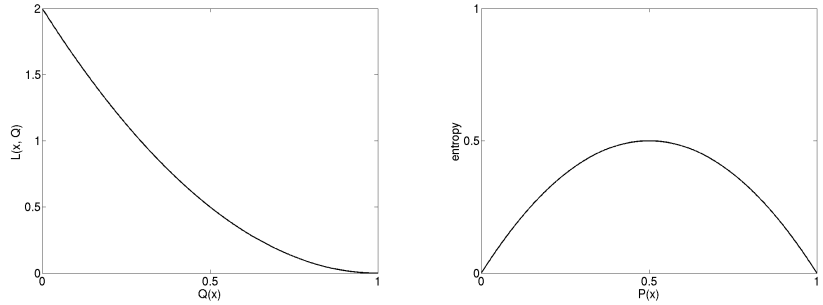


Figure 6.2: Brier loss and entropy on a binary prediction

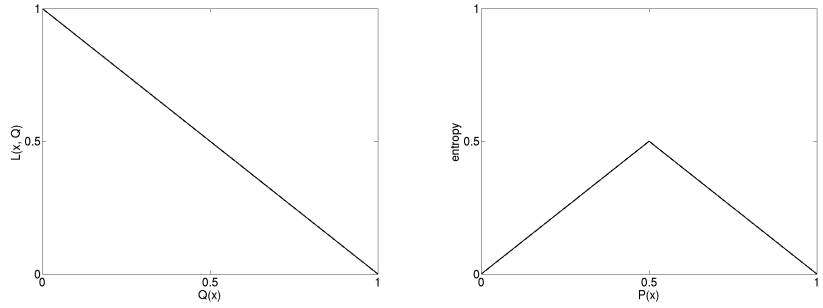


Figure 6.3: Randomized 0-1 loss and entropy on a binary prediction

Its entropy function is $H_L(P) = 1 - \sum_{x \in \mathcal{X}} P(x)^2$; L and H_L are displayed in Figure 6.2 for a binary prediction. Because L is a function of the entire distribution Q and not only of $Q(x)$, the graph for L does not fully capture the behaviour of Brier loss on non-binary predictions.

The third loss function we will often refer to is *randomized 0-1 loss*, given by

$$L(x, Q) = 1 - Q(x).$$

It is improper: an optimal response Q to some distribution P puts all mass on outcome(s) with maximum $P(x)$. Its entropy function is $H_L(P) = 1 - \max_{x \in \mathcal{X}} P(x)$ (see Figure 6.3). It is related to *hard 0-1 loss*, which requires the contestant to pick a single outcome x' and gives loss 0 if $x' = x$ and 1 otherwise. Randomized 0-1 loss essentially allows the contestant to randomize his prediction: $L(x, Q)$ equals the expected value of hard 0-1 loss when x' is distributed according to Q. An important difference between games with hard and randomized 0-1 loss will be shown later in Example 6.F.

6.2.3 Notes on our definition

Our definition of a game rules out duplicate messages in $\mathcal{Y}$, which would not meaningfully change the options of either player as the two messages represent the same move for the quizmaster; this will be made precise in Lemma 6.2. The definition does allow duplicate outcomes: pairs of outcomes $x_1, x_2 \in \mathcal{X}$ such that $x_1 \in y$ if and only if $x_2 \in y$ for all $y \in \mathcal{Y}$. We will see later (in Example 6.D) that games with such outcomes cannot generally be solved in terms of games without, and thus we must analyse them in their own right.

We also ruled out the existence of outcomes with marginal probability zero. A game with such a marginal is equivalent to a game with the corresponding outcomes removed from the outcome space and from all messages; this modified game does have positive marginal probability on all outcomes.

6.3 Worst-case optimal strategies for the quizmaster

In this section, we formulate the problem of finding a worst-case optimal strategy for the quizmaster as a convex optimization problem. Worst-case optimal strategies for the contestant will be discussed in Section 6.4. In order to be applicable to a wide range of loss functions, these two sections are rather technical, and the characterizations of worst-case optimal strategies we find here are not always easy to use (though the abstract results in these sections are illustrated by concrete examples in Sections 6.3.1 and 6.4.3). We will find simpler characterizations for smaller classes of loss functions in Section 6.5. An overview of these results is given in Table 6.1.

For the convex optimization problem we will discuss in this section, it will prove convenient to allow P to range over $\mathbf{R}_{\geq 0}^{\mathcal{R}(\mathcal{X},\mathcal{Y})}$. That is, we allow arbitrary vectors of nonnegative reals that may disobey the marginal constraint,

Table 6.1: Results on worst-case optimal strategies for different loss functions

Conditions on L	Results	Example
H_L finite and continuous	P^* exists and is characterized by Theorem 6.3	hard 0-1 loss
H_L finite and continuous; all minimal supporting hyperplanes realizable	Q^* exists, a Nash equilibrium exists (Theorem 6.5); Q^* characterized by Theorem 6.7	randomized 0-1 loss
L proper and continuous; H_L finite and continuous	all the above simplified by Theorem 6.9	Brier loss
L local and proper; H_L finite and continuous	characterization of P^* simplified further by Theorem 6.10 (RCAR condition)	logarithmic loss

and might not even sum to one where we would normally expect a probability distribution (though we do still have $P(x,y) = 0$ for $x \notin y$). However, when $P(y) > 0$ for some y, $P(\cdot \mid y)$ defined by $P(x \mid y) := P(x,y)/P(y)$ as in Section 6.2.1 still defines a probability distribution, because the scale factor cancels out. The constraints in the optimization problem will then enforce that our solution P lies in $\mathcal{P}$; such a P is called a *feasible* solution.

The following function extends the quizmaster's objective function (6.4) (the expected generalized entropy of $P \in \mathcal{P}$) to the domain $\mathbf{R}_{\geq 0}^{\mathcal{R}(\mathcal{X},\mathcal{Y})}$:

$$f_0(P) := \inf_{Q \in \mathcal{Q}} \sum_{y \in \mathcal{Y}, x \in y} P(x,y) L(x, Q_{|y}). \tag{6.8}$$

Note that while the quizmaster is given more freedom in that we allow $P \notin \mathcal{P}$ as argument to f_0, the minimization (representing the contestant's choice) is still restricted to the set of strategies $\mathcal{Q}$ defined in the previous section. Because the conditional distributions that make up Q can be chosen independently for each y, we can rewrite f_0 in terms of ordinary generalized entropies as follows:

$$f_0(P) = \sum_{\substack{y \in \mathcal{Y}: \\ P(y) > 0}} \inf_{Q_{|y} \in \Delta\mathcal{X}} \sum_{x \in y} P(x,y) L(x, Q_{|y}) = \sum_{\substack{y \in \mathcal{Y}: \\ P(y) > 0}} P(y) H_L(P(\cdot \mid y)).$$

We will need the following properties of H_L throughout our theory:

Lemma 6.1. *For all loss functions L, if H_L is finite, then it is also concave and lower semi-continuous. If L is finite everywhere, then H_L is finite, concave, and continuous.*

(When we talk about (semi-)continuity, this is always with respect to the extended real line topology of losses, as in Rockafellar (1970, Section 7).)

Using just the concavity of the objective f_0 (which is a linear combination of concave generalized entropies), we can prove the following intuitive result.

Lemma 6.2 (Message subsumption)**.** *Suppose that for $P \in \mathcal{P}$ there are two messages $y_1, y_2 \in \mathcal{Y}$ such that any outcome $x \in y_2$ with $P(x, y_2) > 0$ is also in y_1. Then if H_L is concave, the quizmaster can do at least as well without using y_2. More precisely, P' given by*

$$P'(x,y) = \begin{cases} P(x,y_1) + P(x,y_2) & \text{for } y = y_1; \\ 0 & \text{for } y = y_2; \\ P(x,y) & \text{otherwise.} \end{cases}$$

is also in $\mathcal{P}$ and its expected generalized entropy is at least as large as that of P. In particular, if P is worst-case optimal, then so is P'.

In particular, if $y_1 \supset y_2$, any strategy P can be replaced by a strategy P' with $P'(y_2) = 0$ without making things worse for the quizmaster. Thus the quizmaster, who wants to maximize the contestant's expected loss, never needs to use a message that is contained in another.

A *dominating hyperplane* to a function f from $D \subseteq \mathbf{R}^{\mathcal{X}}$ to $\mathbf{R}$ is a hyperplane in $\mathbf{R}^{\mathcal{X}} \times \mathbf{R}$ that is nowhere below f. A *supporting hyperplane* to f (at P) is a dominating hyperplane that touches f at some point P.[2] A concave function has at least one supporting hyperplane at every point (Rockafellar, 1970, Theorem 11.6), but it may be vertical. A nonvertical hyperplane can be described by a linear function $\ell : \mathbf{R}^{\mathcal{X}} \to \mathbf{R}$: $\ell(P) = \alpha + \sum_x P(x)\lambda_x$, where $\alpha \in \mathbf{R}$ and $\lambda \in \mathbf{R}^{\mathcal{X}}$.

While H_L is defined as a function of $\Delta_{\mathcal{X}}$, we will often need to talk about supporting hyperplanes to the function H_L restricted to Δ_y for some message $y \in \mathcal{Y}$. We use the notation $H_L \restriction \Delta_y$ for the restriction of H_L to the domain Δ_y. (Recall that we defined Δ_y as a subset of $\Delta_{\mathcal{X}}$.) A supporting hyperplane to $H_L \restriction \Delta_y$ is not a supporting hyperplane to H_L itself if it goes below H_L at some $P \in \Delta_{\mathcal{X}} \setminus \Delta_y$.

A *supergradient* is a generalization of the gradient: a supergradient of a concave function at a point is the gradient of a supporting hyperplane. If $H_L \restriction \Delta_y$ is finite and continuous (and thus concave by Lemma 6.1), then for any vector $\lambda \in \mathbf{R}^{\mathcal{X}}$, a unique supporting hyperplane to $H_L \restriction \Delta_y$ can be found having that vector as its gradient, by choosing α appropriately in $\ell(P) = \alpha + \sum_x P(x)\lambda_x$ (Rockafellar, 1970, Theorem 27.3). It will often be convenient in our discussion to talk about supporting hyperplanes rather than supergradients because they fix this choice of α.

Theorem 6.3 (Existence and characterization of P^*)**.** *For H_L finite and upper semi-continuous (thus continuous), a worst-case optimal strategy for the quizmaster (that is, a $P \in \mathcal{P}$ maximizing (6.4)) exists, and P^* is such a strategy if and only if there exists a $\lambda^* \in \mathbf{R}^{\mathcal{X}}$ such that*

$$H_L(P') \leq \sum_{x \in y} P'(x)\lambda^*_x \quad \text{for all } y \in \mathcal{Y} \text{ and } P' \in \Delta_y,$$

[2] We deviate slightly from standard terminology here: what we call a supporting hyperplane to a concave function f is usually called a supporting hyperplane to $\{(u,v) \in \mathbf{R}^{\mathcal{X}} \times \mathbf{R} \mid v \leq f(u)\}$, the hypograph of f.

with equality if $P^*(y) > 0$ and $P' = P^*(\cdot \mid y)$. That is, for y with $P^*(y) > 0$, the linear function $\sum_{x \in y} P(x)\lambda^*_x$ defines a supporting hyperplane to $H_L \restriction \Delta_y$ at $P^*(\cdot \mid y)$, and a dominating hyperplane for other y.

A vector $\lambda^* \in \mathbf{R}^{\mathcal{X}}$ that satisfies the above for some worst-case optimal P^* satisfies it for all worst-case optimal P^* and is called a Kuhn-Tucker vector *(or* KT-vector*).*

Several examples illustrating the application of Theorem 6.3 are given in Section 6.3.1; a graphical illustration of the theorem is also included there (Figure 6.4). We will see in Section 6.4 that KT-vectors form the bridge between worst-case optimal strategies for the quizmaster and for the contestant.

Note that at any P with $P(y) = 0$ for some $y \in \mathcal{Y}$, the objective function f_0 defined in (6.8) is not differentiable for most L, so there may be multiple supporting hyperplanes at P. (This is why to formulate the preceding theorem we needed the theory of Rockafellar (1970), where no differentiability is assumed.)

6.3.1 Application to standard loss functions

The generalized entropy for logarithmic loss has only vertical supporting hyperplanes at the boundary of Δ_y for any $y \in \mathcal{Y}$. These hyperplanes do not correspond to any KT-vector $\lambda^* \in \mathbf{R}^{\mathcal{X}}$, from which it follows that for any y with $P^*(y) > 0$, the worst-case optimal strategy will not have $P^*(\cdot \mid y)$ at the boundary of Δ_y. The same is not generally true: we will see below how for randomized 0-1 loss (in Example 6.B on page 127, and Example 6.D) and Brier loss (in Example 6.E), games may have a worst-case optimal strategy for the quizmaster that has $P^*(y) > 0$, yet $P^*(x \mid y) = 0$ for some $x \in y \in \mathcal{Y}$.

Of the three loss functions we saw earlier, Brier loss and 0-1 loss are finite, so by Lemma 6.1, all conditions of Theorem 6.3 are satisfied for them. Logarithmic loss is infinite when the obtained outcome was predicted to have probability zero. The generalized entropy is still finite, because for any true distribution P, there exist predictions Q that give finite expected loss (in particular, $Q = P$ does this). The entropy is also continuous: $-P(x) \log P(x)$ is continuous as a function of $P(x)$ with our convention that $0 \cdot \infty = 0$, and H_L is the sum of such continuous functions. Thus we can apply Theorem 6.3 to analyse the Monty Hall problem for each of these three loss functions.

Example 6.B (continued). For Monty Hall, the strategy P^* of choosing a message uniformly when the true outcome is x_2 is worst-case optimal for the quizmaster, for all three loss functions. It is easy to verify that the theorem is satisfied by this strategy combined with the appropriate KT-vector:

$$\text{for logarithmic loss: } \lambda^* = \left(-\log\frac{2}{3}, -\log\frac{1}{3}, -\log\frac{2}{3}\right);$$

$$\text{for Brier loss: } \lambda^* = \left(\frac{2}{9}, \frac{8}{9}, \frac{2}{9}\right);$$

$$\text{for randomized 0-1 loss: } \lambda^* = (0, 1, 0).$$

The situation for logarithmic loss is illustrated in Figure 6.4.

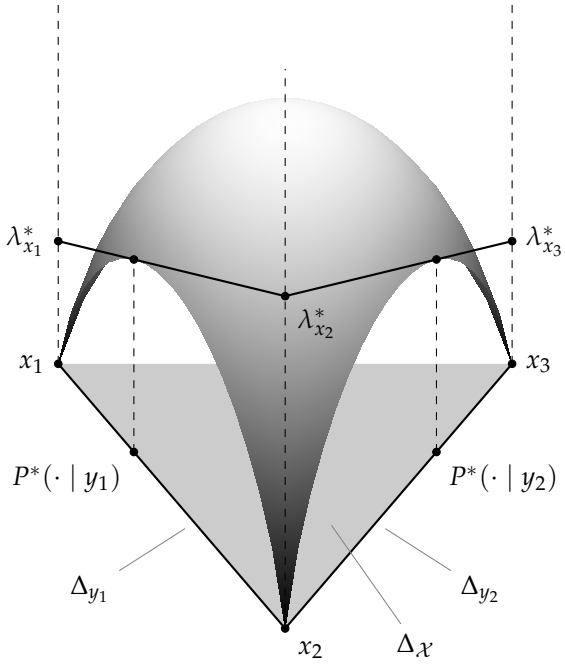


Figure 6.4: The worst-case optimal strategy for the quizmaster in the Monty Hall game with logarithmic loss, as characterized by Theorem 6.3. The triangular base is the full simplex $\Delta_{\mathcal{X}}$, on which the entropy function H_L is defined (this is the grey dome); the points labelled x_1, x_2 and x_3 are the elements of this simplex putting all mass on that single outcome; and the line segments Δ_{y_1} and Δ_{y_2} are the subsets of $\Delta_{\mathcal{X}}$ consisting of all distributions on y_1 and y_2 respectively. Restricted to the domain Δ_{y_1}, the vector λ^* defines a linear function (having height λ^*_x at each $x \in \mathcal{X}$) that is a supporting hyperplane to H_L at $P^*(\cdot \mid y_1)$ (and similar for y_2). Note that when the linear function defined by λ^* is extended to all of $\Delta_{\mathcal{X}}$, it may go below H_L there, but not in Δ_{y_1} or Δ_{y_2}.

We also find that for logarithmic loss and Brier loss, P^* is the unique worst-case optimal strategy, as the hyperplanes specified by λ^* touch the generalized entropy functions at only one point each. For randomized 0-1 loss, on the other hand, all quizmaster strategies are worst-case optimal, as the hyperplane specified by λ^* touches $H_L \restriction \Delta_{y_1}$ at all $P(\cdot \mid y_1)$ with $P(x_1 \mid y_1) \geq 1/2$.

Example 6.C (The quizmaster discards a message)**.** Consider a different game, with $\mathcal{X} = \{x_1, x_2, x_3, x_4\}$, $\mathcal{Y} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_3, x_4\}\}$, p given by $p_{x_4} = 2/5$ and $p_x = 1/5$ elsewhere, and L logarithmic loss. In the terminology of the Monty Hall puzzle, there is no initial choice by the contestant that determines what moves are available to the quizmaster, but the quizmaster will again leave two doors closed: the one hiding the car, and another adjacent to it. Then one strategy for the quizmaster is to never give message y_2 to the contestant; i.e. to pick the following strategy $P \in \mathcal{P}$ with $P(y_2) = 0$:

P	x_1	x_2	x_3	x_4
y_1	$1/5$	$1/5$	$-$	$-$
y_2	$-$	0	0	$-$
y_3	$-$	$-$	$1/5$	$2/5$
p_x	$1/5$	$1/5$	$1/5$	$2/5$

The depicted strategy P is worst-case optimal: When applying the theorem, we see that the KT-vector $\lambda^* = (\log 2, \log 2, \log 3, -\log(2/3))$ gives supporting hyperplanes to $H_L \restriction \Delta_{y_1}$ and $H_L \restriction \Delta_{y_3}$, but a non-supporting dominating hyperplane to $H_L \restriction \Delta_{y_2}$. This strategy can be seen to be intuitively reasonable because when the contestant receives message $y_3 = \{x_3, x_4\}$, he knows that the probability of the true outcome being x_4 is at least twice as large as the probability of it being x_3. By always giving message y_3 when the true outcome is x_3, the quizmaster can keep this difference from becoming larger.

P is also the unique worst-case optimal strategy for Brier loss (as shown by the same analysis) and for randomized 0-1 loss (where the KT-vector is not unique: $(a, 1-a, 1, 0)$ for any $a \in [0, 1]$ is a KT-vector).

In the previous examples, the worst-case optimal strategies P coincided for logarithmic and Brier loss. The following example shows that this is not always the case.

Example 6.D (Dependence on loss function)**.** Consider the family of games with $\mathcal{X} = \{x_1, x_2, x_3, x_4\}$, $\mathcal{Y} = \{\{x_1, x_2\}, \{x_2, x_3, x_4\}\}$, $p_{x_1} = p_{x_2} = 1/3$, and $p_{x_3} = p_{x_4} = 1/6$:

P	x_1	x_2	x_3	x_4
y_1	$1/3$	$1/6$	$-$	$-$
y_2	$-$	$1/6$	$1/6$	$1/6$
p_x	$1/3$	$1/3$	$1/6$	$1/6$

This game is also similar to Monty Hall, but now one door has been 'split in two': the quizmaster will either open door 1, or doors 3 and 4. The strategy P shown above is worst-case optimal for logarithmic loss, but not for Brier loss:

for both loss functions, there is a unique supporting hyperplane for both y that touches $H_L \restriction \Delta_y$ at $P(\cdot \mid y)$ for P as shown in the table, but for Brier loss, these two hyperplanes do not have the same height at the common outcome x_2. (Using Theorem 6.9 from page 136, we can find the worst-case optimal strategy for the quizmaster under Brier loss by solving a quadratic equation with one unknown; this strategy has $P(x_2, y_1) = 11/3 - 2\sqrt{3} \approx 0.20$ and $P(x_2, y_2) = 2\sqrt{3} - 10/3 \approx 0.13$.)

For randomized loss, neither the worst-case optimal strategy nor the KT-vector are unique: the KT-vectors are $(0, 1, a, 1-a)$ for any $a \in [0, 1]$; the worst-case optimal strategies are the P given above, the strategy that always gives message y_1 when the true outcome is x_2, and all convex combinations of these.

Example 6.E (The quizmaster discards a message-outcome pair)**.** Again consider the game from the previous example, but now with a different marginal:

P	x_1	x_2	x_3	x_4
y_1	0.45	0.05	–	–
y_2	–	0	0.25	0.25
p_x	0.45	0.05	0.25	0.25

The strategy P is worst-case optimal for Brier loss, with KT-vector

$$\lambda^* = (0.02, 1.62, 0.5, 0.5).$$

P displays another curious property (that we also saw for randomized 0-1 loss in the previous example): while the quizmaster uses message y_2 for some outcomes, he does not use it in combination with outcome x_2. In the theorem, the hyperplane on Δ_{y_2} is supporting at $P(\cdot \mid y_2)$, but is not a tangent plane: compared to the tangent plane, it has been 'lifted up' at the opposite vertex of the simplex Δ_{y_2} (x_2) to the same height as the supporting hyperplane on Δ_{y_1}.

This behaviour cannot occur in games with logarithmic loss: as we observed at the beginning of Section 6.3.1, if a worst-case optimal strategy P^* has $P^*(y) > 0$ for some $y \in \mathcal{Y}$, then it must assign positive probability to $P^*(x \mid y)$ for all $x \in y$.

6.4 Worst-case optimal strategies for the contestant

We now turn our attention to worst-case optimal strategies for the contestant. To this end, we look at the relation between the KT-vectors that appeared in Theorem 6.3 and the set of strategies $\mathcal{Q}$ the contestant can choose from.

6.4.1 Realizable hyperplanes

For any $y \in \mathcal{Y}$, Δ_y is defined in Section 6.2 as a $(|y| - 1)$-dimensional subset of $\mathbf{R}^{\mathcal{X}}_{\geq 0}$. Thus a linear function $\ell : \Delta_y \to \mathbf{R}$ can be extended to a linear function $\bar{\ell}$ on the domain $\mathbf{R}^{\mathcal{X}}_{\geq 0}$ in different ways. Hence many different vectors $\lambda \in \mathbf{R}^{\mathcal{X}}$

representing supporting hyperplanes will correspond to what we can view as a single supergradient, because the hyperplanes agree on Δ_y. We can make the extension unique by requiring $\bar{\ell}$ to be zero at the origin and at the vertices of the simplex $\Delta_{\mathcal{X}\setminus y}$. Because such a normalized function $\bar{\ell} : \mathbf{R}^{\mathcal{X}}_{\geq 0} \to \mathbf{R}$ obeys $\bar{\ell}(0) = 0$, it can be written as $\bar{\ell}(P) = P^\top \lambda$ for some λ. These functions are thus uniquely identified by their gradients λ, allowing us to refer to them using 'the (supporting) hyperplane λ'. Let Λ_y be the set of all gradients of such normalized functions that represent dominating hyperplanes to $H_L \restriction \Delta_y$; in formula, let

$$\Lambda_y = \{\lambda \in \mathbf{R}^{\mathcal{X}} \mid \lambda_x = 0 \text{ for } x \notin y, \text{ and } \forall P \in \Delta_y : P^\top \lambda \geq H_L(P)\}.$$

For each nonvertical supporting hyperplane of $H_L \restriction \Delta_y$, clearly the gradient is in Λ_y; that is, all finite supergradients of this restricted function have a normalized representative in Λ_y. The set also includes vectors λ for which $P^\top \lambda > H_L(P)$ for all $P \in \Delta_y$, which do not correspond to supporting hyperplanes.

Not all vectors $\lambda \in \Lambda_y$ may be available to the contestant as responses to a play of $y \in \mathcal{Y}$ by the quizmaster. As a trivial example, consider logarithmic loss and a vector λ with $\sum_{x \in y} e^{-\lambda_x} < 1$ and $\lambda_x = 0$ for $x \notin y$. Then $\lambda \in \Lambda_y$ because the hyperplane defined by λ is dominating to $H_L \restriction \Delta_y$ (thus the expected loss from λ is *larger* than what the contestant could achieve), but clearly there is no distribution $Q \in \Delta_{\mathcal{X}}$ that results in these losses on $x \in y$. We say that a vector $\lambda \in \Lambda_y$ is *realizable on* y if there exists a $Q \in \Delta_{\mathcal{X}}$ such that $L(x, Q) = \lambda_x$ for all $x \in y$, and then we say that such a Q *realizes* λ.

A partial order on vectors $\lambda, \lambda' \in \mathbf{R}^{\mathcal{X}}$ is given by: $\lambda \leq \lambda'$ if and only if $\lambda_x \leq \lambda'_x$ for all $x \in \mathcal{X}$. We write $\lambda < \lambda'$ when $\lambda \leq \lambda'$ and $\lambda \neq \lambda'$. For all $y \in \mathcal{Y}$, this partial order has the following property: For $\lambda, \lambda' \in \Lambda_y$, we have $\lambda \leq \lambda'$ if and only if for all $P \in \Delta_y$, $P^\top \lambda \leq P^\top \lambda'$. Therefore if $Q, Q' \in \Delta_{\mathcal{X}}$ realize $\lambda, \lambda' \in \Lambda_y$ respectively and $\lambda \leq \lambda'$, the contestant is never hurt by using Q instead of Q' as a prediction given the message y.

Any minimal element with respect to this partial order defines a supporting hyperplane to $H_L \restriction \Delta_y$. For P in the relative interior of Δ_y, the converse also holds: all supporting hyperplanes at P are minimal elements. This is not the case for P at the relative boundary of Δ_y, where some supporting hyperplanes (the ones that 'tip over' the boundary) are not minimal.

Lemma 6.4. *If H_L is finite and continuous on Δ_y, then the following hold:*

1. *If $\lambda \in \Lambda_y$ is not a supporting hyperplane to $H_L \restriction \Delta_y$, then there exists a supporting hyperplane $\lambda' \in \Lambda_y$ with $\lambda' < \lambda$;*
2. *If $\lambda \in \Lambda_y$ is a supporting hyperplane to $H_L \restriction \Delta_y$ at P but is not minimal in Λ_y, then there exists a minimal $\lambda' < \lambda$ in Λ_y;*
3. *If $\lambda \in \Lambda_y$ is a supporting hyperplane to $H_L \restriction \Delta_y$ at P, then any $\lambda' \leq \lambda$ in Λ_y is a supporting hyperplane at P and obeys $\lambda'_x = \lambda_x$ for all $x \in y$ with $P(x) > 0$.*

Thus the contestant never needs to play a $Q_{|y}$ realizing a non-minimal element of Λ_y.

6.4.2 Existence

With the help of Lemma 6.4, we can formulate sufficient conditions for the existence of a worst-case optimal strategy Q^* for the contestant that, together with P^* for the quizmaster, forms a Nash equilibrium.

Theorem 6.5 (Existence of Q^*)**.** *Suppose that the conditions of Theorem 6.3 hold, and that for all $y \in \mathcal{Y}$, all minimal supporting hyperplanes $\lambda \in \Lambda_y$ to $H_L \restriction \Delta_y$ are realizable on y. Then there exists a worst-case optimal strategy $Q^* \in \mathcal{Q}$ for the contestant that achieves the same expected loss in the minimax game as P^* achieves in the maximin game: (P^*, Q^*) is a Nash equilibrium.*

We will see in Theorem 6.9 that a (or rather, at least one) Nash equilibrium exists for logarithmic loss and Brier loss. The existence of a Nash equilibrium in games with randomized 0-1 loss is shown by the following consequence of Theorem 6.5.

Proposition 6.6. *In games with randomized 0-1 loss, a Nash equilibrium exists.*

The following example shows what may go wrong if some supporting hyperplanes are not realizable.

Example 6.F (Hard 0-1 loss)**.**

P^*	x_1	x_2	x_3
y_1	$1/6$	$1/6$	$-$
y_2	$-$	$1/6$	$1/6$
y_3	$1/6$	$-$	$1/6$
p_x	$1/3$	$1/3$	$1/3$

Consider the game with $\mathcal{X}$, $\mathcal{Y}$ and p as shown in the table, and with hard 0-1 loss (so that the contestant is not allowed to randomize):

$$L(x, Q) = \begin{cases} 0 & \text{if } Q(x) = 1; \\ 1 & \text{otherwise.} \end{cases}$$

This loss function has the same entropy function as randomized 0-1 loss, so the two loss functions are the same from the quizmaster's perspective. The table shows the unique worst-case optimal strategy for the quizmaster, with KT-vector $\lambda^* = (1/2, 1/2, 1/2)$ and expected loss $1/2$. For randomized 0-1 loss, the (as we will see below: unique) worst-case optimal strategy for the contestant would be to respond to any message y with the uniform distribution on y. However, for all $y \in \mathcal{Y}$, λ given by $\lambda_x = \mathbf{1}_{x \in y} \lambda^*_x$ is not realizable on y under hard 0-1 loss, so Theorem 6.5 does not apply. In fact, for any strategy Q the contestant might use, there exists a strategy P for the quizmaster that gives

expected loss 2/3 or larger (because for at least two outcomes x, there must be a $y \ni x$ such that $L(x, Q_{|y}) = 1$). Thus the inequality (6.6) is strict: there is no Nash equilibrium, and a worst-case optimal strategy for either player is optimal only in the minimax/maximin sense.

This example also shows that the condition on existence of supporting hyperplanes in Theorem 6.5 cannot be replaced by the weaker condition that the infimum appearing in the definition (6.3) of H_L is always attained.

Games without Nash equilibria We will now briefly go into the situation seen in the preceding example, where Theorem 6.5 does not apply.

While for some games with hard 0-1 loss, no Nash equilibrium may exist, worst-case optimal strategies for the contestant do exist, and can be characterized using stable sets of a graph. A *stable set* is a set of nodes no two of which are adjacent (Schrijver, 2003b, Chapter 64). Consider the graph with node set $\mathcal{X}$ and with an edge between two nodes if and only if they occur together in some message. A set $S \subseteq \mathcal{X}$ is stable in this graph if and only if there exists a strategy $Q \in \mathcal{Q}$ for the contestant such that for all $x \in S$, $\max_{y: x \in y \in \mathcal{Y}} L(x, Q_{|y}) = 0$, and equal to 1 otherwise. The worst-case loss obtained by this strategy is $1 - \sum_{x \in S} p_x$. Thus finding the worst-case optimal strategy Q for the contestant is equivalent to finding a stable set S with maximum weight. Algorithmically, this is an NP-hard problem in general, though polynomial-time algorithms exist for certain classes of graphs, including perfect (this includes bipartite) graphs and claw-free graphs (Schrijver, 2003b).

With the exception of two examples in Section 6.5.1 illustrating the limits of our theory, we will not look at games without Nash equilibria any more from now on.

6.4.3 Characterization and nonuniqueness

The concept of a KT-vector, which helped characterize worst-case optimal strategies for the quizmaster in Theorem 6.3, now returns for a similar role in the characterization of worst-case optimal strategies for the contestant.

Theorem 6.7 (Characterization of Q^*)**.** *Under the conditions of Theorems 6.3 and 6.5, a strategy $Q^* \in \mathcal{Q}$ is worst-case optimal for the contestant if and only if the vector given by $\lambda_x := \max_{y \ni x} L(x, Q^*_{|y})$ is a KT-vector.*

If the loss $L(x, Q^*_{|y})$ equals λ_x for all $x \in y$, then the worst-case optimal strategy Q^* is an equalizer strategy (Ferguson, 1967): the expected loss of Q^* does not depend on the quizmaster's strategy. Not all games have an equalizer strategy as worst-case optimal strategy, as Example 6.H below shows.

When constructing a worst-case optimal strategy for the contestant, there are three points where different options are available, so that a worst-case optimal Q is in general not unique. The following examples demonstrate these three points.

Example 6.G (λ^* not unique)**.**

	x_1	x_2	x_3	x_4
y_1	$*$	$*$	$-$	$-$
y_2	$-$	$*$	$*$	$-$
y_3	$-$	$-$	$*$	$*$
y_4	$*$	$-$	$-$	$*$
p_x	$1/4$	$1/4$	$1/4$	$1/4$

Consider the game with $\mathcal{X}$, $\mathcal{Y}$ and p as in the table above, and with randomized 0-1 loss. For the quizmaster, any P^* that is uniform given each y is worst-case optimal, and any $\lambda_a = (a, 1-a, a, 1-a)$ with $a \in [0,1]$ is a KT-vector. To each λ_a corresponds a unique worst-case optimal Q^*, namely the strategy that puts conditional probability $1-a$ on outcome x_1 or x_3 (whichever is in the given message), and probability a on x_2 or x_4.

Note that if we replace randomized 0-1 loss by a strictly proper loss function such as logarithmic or Brier loss, the KT-vector and the worst-case optimal strategy for the contestant become unique, while the same set of strategies as before continues to be worst-case optimal for the quizmaster. This shows that the freedom for the contestant we see here for randomized 0-1 loss is due to the nonuniqueness of the KT-vector, not due to the nonuniqueness of P^*.

Example 6.H (Minimal λ not unique)**.**

P^*	x_1	x_2	x_3
y_1	$1/5$	$3/10$	$-$
y_2	$-$	$3/10$	$1/5$
y_3	0	$-$	0
p_x	$1/5$	$3/5$	$1/5$

Consider the game as shown in the table with logarithmic loss; the strategy P^* shown in this table is the unique worst-case optimal strategy for the quizmaster. Because logarithmic loss is proper, we know that $Q^*_{|y_1} = P^*(\cdot \mid y_1)$ and $Q^*_{|y_2} = P^*(\cdot \mid y_2)$ are optimal responses for the contestant, but this does not tell us what $Q^*_{|y_3}$ should be in a worst-case optimal strategy for the contestant.

We see that P^* assigns probability zero to message y_3, and the KT-vector

$$\lambda^* = (-\log\frac{2}{5}, -\log\frac{3}{5}, -\log\frac{2}{5})$$

specifies a hyperplane that does not support H_L in Δ_{y_3}. Hence the construction of $Q^*_{|y_3}$ in the proof of Theorem 6.5 allows freedom in the choice of a minimal element $\lambda \in \Lambda_{y_3}$ less than $(-\log 2/5, 0, -\log 2/5)$: the valid choices are $(-\log q, 0, -\log(1-q))$ for any $q \in [2/5, 3/5]$; each of these is realized on y_3 by $Q_{|y_3} = (q, 0, 1-q)$. Using Theorem 6.7, we can verify that these choices of $Q^*_{|y_3}$ define worst-case optimal strategies: the vector λ defined there equals the KT-vector λ^* given above.

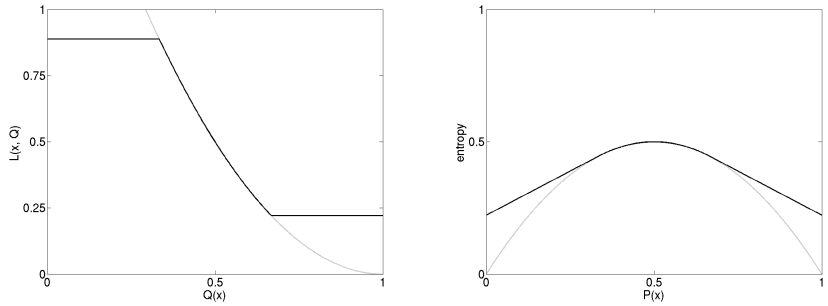


Figure 6.5: Loss and entropy on a binary prediction for the loss function in Example 6.I, with Brier loss and entropy shown in grey

This also shows that worst-case optimal strategies for the contestant cannot be characterized simply as 'optimal responses to P^*': in this example, $P^*(\cdot \mid y_3)$ is undefined, yet there is a nontrivial constraint on $Q_{|y_3}$ in the worst-case optimal strategy Q for the contestant.

Example 6.I (Q realizing λ not unique)**.** Take $\mathcal{X} = \{x_1, x_2, x_3\}$, $\mathcal{Y} = \{y_1 = \{x_1, x_2\}, y_2 = \{x_2, x_3\}\}$, and p uniform (as in Monty Hall), with loss function

$$L(x, Q) = \begin{cases} \frac{8}{9} & \text{for } Q(x) < \frac{1}{3}; \\ L_{\text{Brier}}(x, Q) & \text{for } Q(x) \in [\frac{1}{3}, \frac{2}{3}]; \\ \frac{2}{9} & \text{for } Q(x) > \frac{2}{3}; \end{cases}$$

(illustrated in Figure 6.5; L_{Brier} denotes the Brier loss function). This loss function is proper but not strictly proper: Because its generalized entropy is not strictly concave, a supporting hyperplane to $H_L \restriction \Delta_{y_1}$ exists that supports H_L at all $P \in \Delta_{y_1}$ with $P(x_1) \leq 1/3$. Any Q in the same interval realizes this hyperplane and thus minimizes the expected loss.

6.5 Results for well-behaved loss functions

In the preceding sections, we have established characterization results for the worst-case optimal strategies of both players. While these results are applicable to many loss functions, they have the disadvantage of being complicated, involving supporting hyperplanes. For some common loss functions, simpler characterizations can be given.

6.5.1 Proper continuous loss functions

Recall from page 119 that for a *proper* loss function, the contestant's expected loss for a given message is minimized if his predicted probabilities equal the

true probabilities. Such loss functions are natural to consider in our probability updating game, as our goal will often be to find these true probabilities. However, simplifying our theorems requires further restrictions on the class of loss functions. In this subsection, we consider loss functions that are both proper and continuous.

Lemma 6.8. *If the loss function $L(x, Q)$ is proper and continuous as a function of Q for all x and H_L is finite, then H_L is differentiable in the following sense: for all $y \in \mathcal{Y}$ and all $P \in \Delta_y$, there is at most one element of Λ_y that is a minimal supporting hyperplane to $H_L \restriction \Delta_y$ at P; if P is in the relative interior of Δ_y, there is exactly one. If it exists, the minimal supporting hyperplane at P is realized by $Q_{|y} = P$.*

The uniqueness of minimal supporting hyperplanes in Λ_y is equivalent to there being exactly one equivalence class of supergradients, where supergradients are taken to be equivalent if their corresponding supporting hyperplanes coincide on Δ_y. The property shown in the above lemma is then related to differentiability by Rockafellar (1970, Theorem 25.1), which says that for a finite, concave function such as H_L, uniqueness of the supergradient at P is equivalent to differentiability at P.

Theorem 6.9. *For L proper and continuous and H_L finite and continuous,*

1. *worst-case optimal strategies for both players exist and form a Nash equilibrium;*
2. *there is a unique KT-vector;*
3. *a strategy $P^* \in \mathcal{P}$ for the quizmaster is worst-case optimal if and only if there exists $\lambda^* \in \mathbf{R}^{\mathcal{X}}$ such that*

$$\begin{aligned} L(x, P^*(\cdot \mid y)) &= \lambda^*_x \quad \text{for all } x \in y \text{ with } P^*(x, y) > 0, \\ L(x, P^*(\cdot \mid y)) &\leq \lambda^*_x \quad \text{for all } x \in y \text{ with } P^*(x, y) = 0,\ P^*(y) > 0, \text{ and} \\ \exists Q^*_{|y} : L(x, Q^*_{|y}) &\leq \lambda^*_x \quad \text{for all } x \in y \text{ with } P^*(y) = 0; \end{aligned}$$

4. *a strategy Q^* for the contestant is worst-case optimal if and only if there exists a worst-case optimal P^* such that for all x,*

$$\max_{y \ni x} L(x, Q^*_{|y}) = \max_{\substack{y \ni x, \\ P^*(y) > 0}} L(x, P^*(\cdot \mid y)), \tag{6.9}$$

which holds if and only if (6.9) holds for all worst-case optimal P^*.

Using this theorem, many observations made about logarithmic loss and Brier loss in the examples we have seen so far can now be more easily verified. For instance, in the worst-case optimal strategy we saw in Example 6.E on page 130, we verify that $L(x_2, P^*(\cdot \mid y_2)) = 1.5 \leq 1.62 = \lambda^*_{x_2} = L(x_2, P^*(\cdot \mid y_1))$.

The preceding theory requires that L is both proper and continuous. If one of these conditions is removed, H_L may not be differentiable and the conclusions of Theorem 6.9 may fail to hold. This is illustrated by the following two examples, in which no Nash equilibria exist.

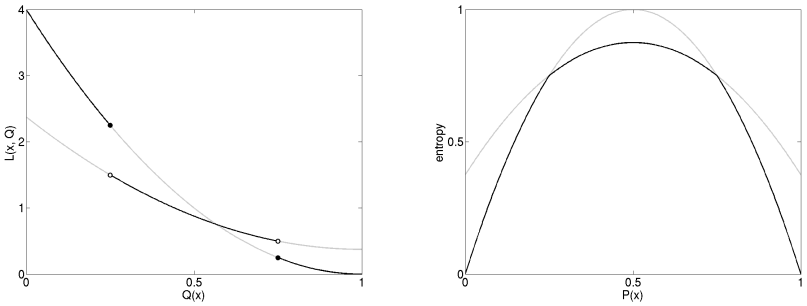


Figure 6.6: Loss and entropy on Δ_{y_1} for the loss function in Example 6.K. The functions are the same on Δ_{y_2} except for the value of L at the discontinuities. The grey lines show the two cases in the definition of L.

Example 6.J (Continuous but improper)**.** The following loss function is continuous, but not proper:

$$L(x, Q) = 1 - Q(x)^2.$$

This loss function is actually hard 0-1 loss (see Example 6.F) in disguise: both have the same entropy function as randomized 0-1 loss (which is not differentiable), and for both, the hyperplane realized by $Q_{|y}$ will touch H_L on Δ_y only if $Q_{|y}$ puts mass 1 on some $x \in y$, so that only a small number of supporting hyperplanes is realizable. Example 6.F shows how for these loss functions, a Nash equilibrium may fail to exist.

(Another example of a continuous but improper loss function is randomized 0-1 loss; while a Nash equilibrium does exist there, parts 2, 3 and 4 of Theorem 6.9 generally do not hold.)

Example 6.K (Strictly proper but not continuous)**.** Take $\mathcal{X} = \{x_1, x_2, x_3\}$ and $\mathcal{Y} = \{y_1 = \{x_1, x_2\}, y_2 = \{x_2, x_3\}\}$ as in Monty Hall, with loss function (we write L_{Brier} for the Brier loss function and H_{Brier} for its generalized entropy)

$$L(x, Q) = \begin{cases} L_{\text{Brier}}(x, Q) + \frac{3}{8} & \text{if } Q(x) \in (1/4, 3/4), \text{ or} \\ & \quad Q(x) \in [1/4, 3/4] \text{ and } Q(x_1) = 0; \\ 2L_{\text{Brier}}(x, Q) & \text{otherwise} \end{cases}$$

(illustrated in Figure 6.6). For each $P \in \Delta_y$ (for both $y \in \mathcal{Y}$), $Q = P$ realizes a supporting hyperplane to $H_L \restriction \Delta_y$ at P, so that L is proper. For both $y \in \mathcal{Y}$, the entropy H_L as a function of $P \in \Delta_y$ is the pointwise minimum of $2H_{\text{Brier}}$ and $H_{\text{Brier}} + 3/8$ (shown in grey in Figure 6.6). These two entropies coincide when $P(x_2) \in \{1/4, 3/4\}$, where $H_L \restriction \Delta_y$ is not differentiable but has multiple supporting hyperplanes. Only one of these hyperplanes is realizable. This means that L is strictly proper, but also that Theorem 6.5 does not apply.

Now consider the game with marginal $p = (1/8, 3/4, 1/8)$. The worst-case optimal strategy for the quizmaster has $P^*(x_2 \mid y_1) = P^*(x_2 \mid y_2) = 3/4$. Because L is strictly proper, the unique optimal responses $Q_{|y_1}$ and $Q_{|y_2}$ for the

contestant must coincide with these conditional distributions for (P^*, Q) to be a Nash equilibrium. However, $L(x_2, Q_{|y_1}) < L(x_2, Q_{|y_2})$, so that the quizmaster can increase the expected loss by choosing to give message y_2 more often when the true outcome is x_2. This shows that Q is not a worst-case optimal strategy for the contestant. The contestant can do better in the minimax game by choosing $Q_{|y_2}$ with $Q_{|y_2}(x_2) = 3/4 + \epsilon > 3/4$, so that $L(x_2, Q_{|y_2}) < L(x_2, Q_{|y_1})$. The expected loss of this strategy can be made arbitrarily close, but not equal to, the worst-case expected loss achieved by the quizmaster in the maximin game.

While uniqueness of λ^* was established by the theorem, we do not have uniqueness of P^* or Q^*. Multiple worst-case optimal strategies Q^* for the contestant may exist as soon as a message is unused, as in Example 6.H. Multiple worst-case optimal strategies P^* for the quizmaster are also possible. For instance, the loss function in Example 6.I is proper and continuous (though not strictly proper) and thus H_L is differentiable. But if the marginal is $p = (2/5, 1/5, 2/5)$, *all* strategies for the quizmaster are worst-case optimal.

6.5.2 Local loss functions

Logarithmic loss is an example of a *local* loss function: a loss function where the loss $L(x, Q)$ depends on the probability assigned by the prediction Q to the obtained outcome x, but not on the probabilities assigned to outcomes that did not occur. The following theorem shows how for such loss functions, worst-case optimality of the quizmaster's strategy can be characterized purely in terms of probabilities, without converting them to losses.

Among loss functions that are 'smooth' for all x, logarithmic loss is, up to some transformations, the only proper local loss function (Bernardo, 1979). We do not know what non-smooth local proper loss functions may exist. In particular, it is conceivable (yet unlikely) that a discontinuous L exists satisfying the conditions of Theorem 6.10, but not those of Theorem 6.9.

Theorem 6.10 (Characterization of P^* for local L)**.** *For L local and proper and H_L finite and continuous, $P^* \in \mathcal{P}$ is worst-case optimal if there exists a vector $q \in [0,1]^{\mathcal{X}}$ such that*

$$\begin{aligned} q_x &= P^*(x \mid y) \text{ for all } x \in y \in \mathcal{Y} \text{ with } P^*(y > 0), \text{ and} \\ \sum_{x \in y} q_x &\leq 1 \text{ for all } y \in \mathcal{Y}. \end{aligned} \tag{6.10}$$

If additionally $H_L \restriction \Delta_y$ is strictly concave for all $y \in \mathcal{Y}$, only such P^* are worst-case optimal for L.

If additionally L is continuous, then Theorem 6.9 applies, and it follows that $Q^* \in \mathcal{Q}$ is a worst-case optimal strategy for the contestant if $Q^*_{|y}(x) \geq q_x$ for all $x \in y \in \mathcal{Y}$. For strictly proper loss functions such as logarithmic loss, this fully characterizes the worst-case optimal strategies for the contestant.

Example 6.H (continued). Consider again the following game:

P^*	x_1	x_2	x_3
y_1	$1/5$	$3/10$	$-$
y_2	$-$	$3/10$	$1/5$
y_3	0	$-$	0
p_x	$1/5$	$3/5$	$1/5$

with logarithmic loss. The conditionals $P^*(x \mid y)$ agree with the vector $q = (2/5, 3/5, 2/5)$. For all $y \in \mathcal{Y}$ with $P^*(y) > 0$, this implies that $\sum_{x \in y} q_x = 1$; for y_3, we see that this sum equals $4/5 \leq 1$. Thus P^* is verified to be worst-case optimal.

The equality of conditionals $P^*(x \mid y)$ with the same x in the statement of Theorem 6.10 is oddly similar to the CAR condition we saw in Section 6.1, but reversing the roles of outcomes and messages. We may say that a strategy P^* satisfying (6.10) is *RCAR* (sometimes *with vector q*), for 'reverse CAR'. Note that whether a strategy is RCAR does not depend on the loss function.

A vector q is called an *RCAR vector* if a strategy $P^* \in \mathcal{P}$ exists such that P^* and q satisfy (6.10). This definition is also independent of the loss function. If q is an RCAR vector, then $q_x > 0$ for all $x \in \mathcal{X}$; otherwise we would get $P^*(x) = 0 < p_x$. Like the KT-vector λ^* in Theorem 6.9, the RCAR vector is unique:

Lemma 6.11. *Given $\mathcal{X}, \mathcal{Y}, p$, there exists a unique RCAR vector $q \in [0,1]^{\mathcal{X}}$.*

If each message in $\mathcal{Y}$ contains an outcome x not contained in any other message, then any strategy P^* must have $P^*(y) > 0$ for all $y \in \mathcal{Y}$. Then the first line of (6.10) implies that $\sum_{x \in y} q_x = 1$ for all y. Thus the second line is now satisfied automatically, allowing the theorem to be simplified for this case:

Corollary 6.12. *A strategy $P^* \in \mathcal{P}$ with $P^*(y) > 0$ for all $y \in \mathcal{Y}$ that satisfies*

$$P^*(x \mid y) = P^*(x \mid y') \text{ for all } y, y' \ni x \tag{6.11}$$

is worst-case optimal for the loss functions covered by Theorem 6.10

In this case, P^* is an equalizer strategy (Ferguson, 1967).

The symmetry between versions of CAR and RCAR is clearest in Corollary 6.12: the condition (6.11) is the mirror image of the definition of *strong CAR* in Jaeger (2005a). Thus we may call it *strong RCAR*. Ordinary RCAR (6.10) imposes an inequality on q for messages with probability 0, which has no analogue in the CAR literature that we know of: the definition of *weak CAR* in Jaeger puts no requirement at all on outcomes with probability 0.

Strict concavity of H_L occurred as a new condition in Theorem 6.10. The main loss function of interest here is logarithmic loss, and its entropy is strictly concave. For other loss functions, the following lemma relates strict concavity of H_L to conditions we have seen before.

Lemma 6.13. *If L is strictly proper and all minimal supporting hyperplanes $\lambda \in \Lambda_y$ to $H_L \restriction \Delta_y$ are realizable on y for all $y \in \mathcal{Y}$, then H_L is strictly concave on $H_L \restriction \Delta_y$ for all $y \in \mathcal{Y}$.*

Affine transformations of the loss function Above, we mentioned that logarithmic loss is the only local proper loss function up to some transformations. The transformations considered in Bernardo (1979) are affine transformations, of the form

$$L'(x, Q) = aL(x, Q) + b_x \tag{6.12}$$

for $a \in \mathbf{R}_{>0}$ and $b \in \mathbf{R}^{\mathcal{X}}$. (This transformation can result in a function L' that can take negative values, so that it does not satisfy our definition of a loss function. However, our results can easily be extended to loss functions bounded from below by an arbitrary real number, so we allow such transformations here.)

The following lemma shows that, for logarithmic loss as well as for other loss functions, the transformation (6.12) does not change how the players of the probability updating game should act.

Lemma 6.14. *Let L be a loss function for which H_L is finite and continuous, and let L' be an affine transformation of L as in (6.12). Then a strategy P^* is worst-case optimal for the quizmaster in the game $\mathcal{G}' := (\mathcal{X}, \mathcal{Y}, p, L')$ if and only if P^* is worst-case optimal in $\mathcal{G} := (\mathcal{X}, \mathcal{Y}, p, L)$. If $\mathcal{G}$ also satisfies the conditions of Theorem 6.5, then the same equivalence holds for worst-case optimal strategies Q^* for the contestant.*

Lemma 6.14 has highly important implications when applied to the logarithmic loss. While multiplying logarithmic loss by a constant $a \neq 1$ merely corresponds to changing the base of the logarithm, *adding* constants b_x allows the logarithmic loss to become the appropriate loss function for a very wide class of games. This means that the RCAR characterization of worst-case optimal strategies for logarithmic loss is also valid for all these games. We are referring to so-called *Kelly gambling* games, also known as *horse race games* (Cover and Thomas, 1991) in the literature. In such games (with terminology adapted to our setting), for any outcome x the contestant can buy a ticket which costs €1 and which pays off a positive amount €c_x if x actually obtains; if some $x' \neq x$ is realized, nothing is paid so the €1 is lost. The contestant is allowed to distribute his capital over several tickets (outcomes), and he is also allowed to buy a fractional nonnegative number of tickets. For example, if $\mathcal{X} = \{1, 2\}$ and $c_1 = c_2 = 2$, then the contestant is guaranteed to neither win nor lose any money if he splits his capital fifty-fifty over both outcomes.

Now consider a contestant with some initial capital (say, €1), who faces an i.i.d. sequence $(X_1, Y_1), (X_2, Y_2), \ldots \sim P$ of outcomes in $\mathcal{X} \times \mathcal{Y}$. At each point in time i he observes 'side information' $Y_i = y_i$ and he distributes his capital gained so far over all $x \in \mathcal{X}$, putting some fraction $Q_{|y_i}(x)$ of his capital on outcome x. Then he is paid out according to the x_i that was actually realized. Here each $Q_{|y}$ is a probability distribution over $\mathcal{X}$, i.e. for all $y \in \mathcal{Y}$, all $x \in \mathcal{X}$, $Q_{|y}(x) \geq 0$ and $\sum_{x \in \mathcal{X}} Q_{|y}(x) = 1$. So if his capital was U_i before the i-th round, it will be $U_i \cdot Q_{|y_i}(x_i) c_{x_i}$ after the i-th round. By the law of large numbers, his capital will grow (or shrink, depending on the odds on offer) almost surely exponentially fast, with exponent $\mathbf{E}_{X,Y \sim P}[\log Q_{|Y}(X) c_X] = \mathbf{E}_{X,Y \sim P}[\log Q_{|Y}(X) - b_X]$, where $b_x = -\log c_x$ (Cover and Thomas, 1991, Chapter 6). Thus, the contestant's capital will grow fastest, among all constant strategies and against

an adversarial distribution $P \in \mathcal{P}$, if he plays a worst-case optimal strategy for gains $\log Q(x) - b_x$, i.e. for loss function $L'(x, Q) = -\log Q(x) + b_x$. By Lemma 6.14 above, this worst-case optimal strategy Q^* is just the Q^* that is also worst-case optimal for logarithmic loss — *it does not depend on the pay-offs* ('odds' in the horse race interpretation) c_x. Clearly, if data are i.i.d. then this continues to hold even if the pay-offs are allowed to change over time, and even if the contestant is allowed to use different strategies at different time points: the worst-case optimal capital growth rate is always achieved by choosing Q^* at all time points.

The upshot is that whenever the probability updating game is played (a) repeatedly, and (b) the contestant is allowed to reinvest and redistribute his capital over all outcomes at each point in time, then his worst-case optimal strategy is equal to the worst-case optimal Q^* for logarithmic loss *irrespective of the pay-offs*. This makes the logarithmic loss, and hence the RCAR characterization, appropriate for a very wide class of settings.

6.6 Conclusion

We have seen many theorems in the last few subsections showing different properties of worst-case optimal strategies P^* and Q^* for the two players for different classes of loss functions. A summary of these theorems was given in Table 6.1 on page 125.

Worst-case optimal probability updating provides a robust new approach for dealing with underspecified distributions. There are many scenarios in which our results currently do not apply, but to which they might be extended. For example, the quizmaster's hard constraint $y \ni x$ could be replaced by some soft constraint, so that each message y still carries information about the true outcome, but no longer in the form of a subset of $\mathcal{X}$. One way to achieve this might be by affine transformations of the loss function as discussed at the end of Section 6.5.2, but allowing the constants to depend on both x and y. This could give a worst-case analogue to *Jeffrey conditioning* or *minimum relative entropy updating* (Grünwald and Halpern, 2003).

Another extension would be to infinite outcome and message spaces.

An online version of the probability updating game may also be considered (Cesa-Bianchi and Lugosi, 2006).

Other questions concern the comparison between different alternative approaches the contestant might use to update his probabilities. For example, can we bound the difference in expected loss between worst-case optimal and naive conditioning? What about ignoring the message and always predicting with the marginal, or ignoring the constraints imposed on the quizmaster by the marginal and predicting with the maximum entropy distribution on y? (Both these strategies are overly pessimistic.) Conversely, we might wonder how much the contestant loses by playing a worst-case optimal strategy when the quizmaster is not adversarial, but for instance chooses from the available messages uniformly at random.

Appendix 6.A Proofs

Proof of Lemma 6.1. For finite H_L, concavity of H_L is shown by Grünwald and Dawid (2004, Proposition 3.2), and lower semi-continuity by Rockafellar (1970, Theorem 10.2) (using that the domain of H_L is a simplex). If L is finite, then picking any $Q \in \Delta_{\mathcal{X}}$ gives an upper bound to H_L, so that H_L is in particular finite. Concavity now follows by the first claim, and continuity by Grünwald and Dawid (2004, Corollary 3.3; an important condition is in Corollary 3.2). □

Proof of Lemma 6.2. If $P(y_2) = 0$ then $P' = P$ and the result is trivial; if $P(y_2) > 0$ but $P(y_1) = 0$, then $P(\cdot \mid y_2) = P'(\cdot \mid y_1)$ so P and P' have the same expected generalized entropy. Otherwise $P(\cdot \mid y_1)$ and $P(\cdot \mid y_2)$ are well-defined, and $P'(\cdot \mid y_1) = (P(y_1)P(\cdot \mid y_1) + P(y_2)P(\cdot \mid y_2))/(P(y_1) + P(y_2))$ is a convex combination of them. By concavity of H_L, $\sum_y P'(y)H_L(P'(\cdot \mid y)) \geq \sum_y P(y)H_L(P(\cdot \mid y))$. □

Proof of Theorem 6.3. Rockafellar (1970, Theorem 27.3) provides conditions under which a convex minimization problem has a solution attaining the minimum. These are satisfied by $\mathcal{P}$ and $-H_L$: $\mathcal{P}$ is nonempty, closed, convex, and bounded (thus has no direction of recession), and $-H_L$ is convex (Lemma 6.1), finite for all $P \in \mathcal{P}$ (thus proper), and lower semi-continuous (thus closed).

By Rockafellar (1970, Corollary 28.2.2), a KT-vector λ^* exists, so that for the remaining claims of the theorem, it suffices to show that P^* is worst-case optimal and λ^* is a KT-vector if and only if the given conditions on (P^*, λ^*) hold.

We rewrite the maximin problem to

$$\begin{aligned} &\text{maximize} && f_0(P) \\ &\text{subject to} && \sum_{y \ni x} P(x,y) = p_x \quad \text{for all } x \in \mathcal{X}, \end{aligned}$$

with $P \in \mathbf{R}_{\geq 0}^{\mathcal{R}(\mathcal{X},\mathcal{Y})}$. By Rockafellar (1970, Theorem 28.3), $P^* \in \mathbf{R}_{\geq 0}^{\mathcal{R}(\mathcal{X},\mathcal{Y})}$ maximizes this and $\lambda^* \in \mathbf{R}^{\mathcal{X}}$ is a KT-vector if and only if $P^* \in \mathcal{P}$ and at P^*, the zero vector is a supergradient to

$$f_0(P^*) - \sum_{x \in \mathcal{X}} \lambda_x^* \left(\sum_{y \ni x} P^*(x,y) - p_x \right). \tag{6.13}$$

The term being subtracted is linear, with gradient $\bar{\lambda} \in \mathbf{R}^{\mathcal{R}(\mathcal{X},\mathcal{Y})}$ given by

$$\bar{\lambda}_{x,y} := \frac{\partial}{\partial P^*(x,y)} \sum_{x \in \mathcal{X}} \lambda_x^* \left(\sum_{y \ni x} P^*(x,y) - p_x \right) = \lambda_x^*. \tag{6.14}$$

By Rockafellar (1970, Theorem 23.8), 0 is a supergradient to (6.13) if and only if $\bar{\lambda}$ is a supergradient to f_0 at P^*.

For any P^* that is not everywhere zero, we have for all $c \geq 0$ that $f_0(cP^*) = cf_0(P^*)$, so that a supporting hyperplane to f_0 at a feasible P^* must go through

the origin. Then the supporting hyperplane with gradient $\bar{\lambda}$ has as defining equation the linear expression $\sum_{x,y} P(x,y)\bar{\lambda}_{x,y}$.

If $\sum_{x,y} P(x,y)\bar{\lambda}_{x,y}$ defines a supporting hyperplane to f_0 at P^*, then

1. at every $y \in \mathcal{Y}$ with $P^*(y) > 0$, it is a supporting hyperplane to $H_L \restriction \Delta_y$ at $P^*(\cdot \mid y)$, and
2. for every y with $P^*(y) = 0$, $H_L(P') \leq \sum_x P'(x)\bar{\lambda}_{x,y}$ for all $P' \in \Delta_y$.

The converse also holds: we have for all $y \in \mathcal{Y}$ and $P' \in \Delta_y$ that $H_L(P') \leq \sum_{x \in y} P'\bar{\lambda}_{x,y}$, with equality if $P^*(y) > 0$ and $P' = P^*(\cdot \mid y)$; taking the convex combination with coefficients $P^*(y)$ shows that the hyperplane defined by $\sum_{x,y} P(x,y)\bar{\lambda}_{x,y}$ is nowhere below f_0 and touches it at $P = P^*$.

For $\bar{\lambda}$ of the required form (6.14), this is in turn equivalent to the characterization given in the statement of the theorem. □

Proof of Lemma 6.4. The function $\sum_{x \in y} \lambda_x P(x) - H_L(P)$ attains its minimum d on Δ_y at some P (Rockafellar, 1970, Theorem 27.3). Let $\lambda' \in \Lambda_y$ be given by $\lambda'_x = \lambda_x - d$ for all $x \in y$: this defines a hyperplane to $H_L \restriction \Delta_y$ that is supporting at the minimizing P, proving the first part of the lemma.

For the third part, let λ, λ' and P be as described. $\lambda' \leq \lambda$ implies $P^\top\lambda' \leq P^\top\lambda$. Neither can be smaller than $H_L(P)$, and since the right hand side must equal $H_L(P)$ because λ is supporting at P, so must the left hand side, showing that λ' is also supporting at P. If $P(x) = 1$ for some $x \in y$, then $P^\top\lambda'' = \lambda''_x$ for any λ'', so in particular $\lambda'_x = \lambda_x = H_L(P)$. For other $P \in \Delta_y$, we use that two linear functions obeying an inequality on their domain $D := \Delta_{\{x \in y \mid P(x) > 0\}}$ and coinciding at a point in the relative interior of D must coincide everywhere on D, so that again $\lambda'_x = \lambda_x$ for $x \in y$ with $P(x) > 0$.

It remains to show that given such a λ, a minimal $\lambda' \leq \lambda$ exists in Λ_y. Consider the set $\Lambda' := \{\lambda' \in \Lambda_y \mid \lambda' \leq \lambda\}$. The set of supporting hyperplanes to $H_L \restriction \Delta_y$ at P in Λ_y is closed (Rockafellar, 1970, Section 23, definition subdifferential (page 215)); Λ' is a subset of this set (as we just saw), obtained by adding further non-strict linear constraints, so it too is closed. It also has the property that if $\lambda' \in \Lambda'$ is minimal in that set, it is also minimal in Λ_y. Now fix any P' in the relative interior of Δ_y, and pick some $\lambda' \in \Lambda'$ that minimizes ${P'}^\top\lambda'$ (this minimum must be attained because the expression is bounded below and Λ' is closed). Such a λ' is also minimal in the partial order, so it is the element we are looking for. □

Proof of Theorem 6.5. Take a worst-case optimal strategy P^* for the quizmaster and KT-vector λ^*. For each $y \in \mathcal{Y}$, define a vector

$$\lambda'_x = \begin{cases} \lambda^*_x & \text{for } x \in y \\ 0 & \text{for } x \notin y. \end{cases}$$

By the statement of Theorem 6.3, $\lambda' \in \Lambda_y$. Let λ be a minimal element of Λ_y with $\lambda \leq \lambda'$: such an element exists by parts 1 and 2 of Lemma 6.4. (If λ' is

itself minimal, $\lambda = \lambda'$). By assumption, λ is realizable on y. Let $Q^*_{|y}$ be given by this Q.

By playing this Q^*, the contestant will achieve expected loss (against any strategy $P \in \mathcal{P}$ for the quizmaster, for λ^* any KT-vector)

$$\sum_{x,y} P(x,y)L(x,Q^*_{|y}) \leq \sum_{x,y} P(x,y)\lambda^*_x = \sum_x p_x \lambda^*_x.$$

The right-hand side is the maximum loss the quizmaster can achieve in the maximin game. By (6.6), the reverse inequality also holds, so we find that the values of the minimax and maximin games must be equal. □

Proof of Proposition 6.6. We first introduce some additional terminology in order to apply a corollary from Rockafellar (1970).

A nonvertical hyperplane defined by $\lambda \in \mathbf{R}^{\mathcal{X}}$ is geometrically a subset of $\mathbf{R}^{\mathcal{X}} \times \mathbf{R}$, namely $\{(P', z') \in \mathbf{R}^{\mathcal{X}} \times \mathbf{R} \mid z' = {P'}^\top \lambda\}$. This set is the boundary of the half-space $H_\lambda = \{(P', z') \in \mathbf{R}^{\mathcal{X}} \times \mathbf{R} \mid z' \leq {P'}^\top \lambda\}$. A hyperplane λ is supporting to a concave function $f : \mathbf{R}^{\mathcal{X}} \to \mathbf{R}$ at the point $P \in \mathbf{R}^{\mathcal{X}}$ with $f(P) = P^\top \lambda$ if and only if the hypograph of f is a subset of H_λ.

A column vector $(-\alpha\lambda, \alpha)$ is called *normal* to a convex set C at a point $(P, z) \in C$ if $(P' - P, z' - z)^\top (-\alpha\lambda, \alpha) \leq 0$ for all $(P', z') \in C$ (Rockafellar, 1970); that is, if $C \subseteq \{(P', z') \mid (P' - P, z' - z)^\top (-\alpha\lambda, \alpha) \leq 0\}$. This latter set is equal to H_λ if $\alpha > 0$ and $z = P^\top \lambda$. So if C is the hypograph of f and $f(P) = z = P^\top \lambda$, then λ is a supporting hyperplane to f at P if and only if $(-\lambda, 1)$ is normal to C.

The set of all vectors normal to C at (P, z) is called the *normal cone* at (P, z). The normal cone to H_λ at given $(P, P^\top \lambda)$ is the half-line $\{(-\alpha\lambda, \alpha) \mid \alpha \in [0, \infty)\}$.

For L randomized 0-1 loss, let the function $f_0 : \mathbf{R}^{\mathcal{X}} \to \mathbf{R}$ be given by $f_0(P) = \min_{Q \in \Delta_{\mathcal{X}}} \sum_{x' \in \mathcal{X}} P(x')L(x', Q)$; note that $f_0 \restriction \Delta_{\mathcal{X}} = H_L$, and that for all $y \in \mathcal{Y}$, any minimal supporting hyperplane $\lambda \in \Lambda_y$ to $H_L \restriction \Delta_y$ can be extended to a supporting hyperplane λ' to f_0 with $\lambda'_x = \lambda_x$ for all $x \in y$.

The hypograph of f_0 is $C = \bigcap_{x \in \mathcal{X}} H_{\lambda^{(x)}}$, with $\lambda^{(x)}_{x'} = L(x', e_x)$ (where e_x is the distribution that puts all mass on x). By Rockafellar (1970, Corollary 23.8.1), for C of this form and (P, z) a point on the boundary of C, the normal cone of C at (P, z) is the sum of the individual normal cones. The normal cone of any set at a point in the interior of that set is just $\{0\}$, so we can ignore those halfspaces when determining the normal cone. Then the corollary says that any vector $(-\lambda, 1)$ normal to f_0 at $(P, f_0(P))$ can be written as $\sum_{x \in \mathcal{X} : P^\top \lambda^{(x)} = f_0(P)} (-\alpha_x \lambda^{(x)}, \alpha_x)$: λ is a convex combination of those $\lambda^{(x)}$.

Conclusion: any minimal supporting hyperplane λ to $H_L \restriction \Delta_y$ at $P \in \Delta_y$ with randomized 0-1 loss is a convex combination of the hyperplanes realized by hard 0-1 loss that are supporting at P. Therefore, randomizing allows the contestant to realize λ. □

Proof of Theorem 6.7. From Theorem 6.5, we know that a strategy exists for the contestant that achieves loss $\sum_x p_x \lambda^*_x$ where λ^* is any KT-vector, and that

this is worst-case optimal. Hence Q^* is worst-case optimal if and only if it achieves the same worst-case expected loss. The worst-case expected loss of a strategy $Q \in \mathcal{Q}$ is

$$\max_{P \in \mathcal{P}} \sum_{x,y} P(x,y) L(x, Q_{|y}) = \sum_x p_x \max_{y \ni x} L(x, Q_{|y}).$$

Therefore if for all x, y with $x \in y$, we have $L(x, Q_{|y}) \leq \lambda^*_x$ for some KT-vector λ^*, Q is worst-case optimal.

For the converse, pick any $Q \in \mathcal{Q}$ and suppose the vector given by $\lambda_x \coloneqq \max_y L(x, Q_{|y})$ is not a KT-vector. Then by Theorem 6.3, there is no $P \in \mathcal{P}$ such that P and λ satisfy the conditions of that theorem. Equivalently, for all $P \in \mathcal{P}$, there either is a message y such that the hyperplane defined by λ passes below H_L somewhere in Δ_y, or there is a message y with $P(y) > 0$ but the hyperplane lies strictly above H_L at $P(\cdot \mid y)$. The former contradicts the definition of H_L, so for λ not a KT-vector, the latter must be the case. But then against any $P \in \mathcal{P}$ (in particular against worst-case optimal P), there is a different strategy $Q' \in \mathcal{Q}$ that is equal to Q except for its response to the message y: $Q'_{|y}$ realizes a supporting hyperplane to $H_L \restriction \Delta_y$ at $P(\cdot \mid y)$. This strategy Q' obtains strictly smaller expected loss than Q, so Q is not worst-case optimal. (In other words: in a Nash equilibrium (P^*, Q^*), the contestant can only do worse against P^* by changing strategy, but here he can do better.) □

Proof of Lemma 6.8. For proper loss functions, L and H_L are related as follows: at all $y \in \mathcal{Y}$ and $P \in \Delta_y$, if the vector $\lambda = L(\cdot, P)$ is finite at all $x \in y$, it describes the nonvertical hyperplane realized by P, which is a supporting hyperplane to $H_L \restriction \Delta_y$ at P.

Now suppose that at some $P \in \Delta_y$, there exists a minimal supporting hyperplane $\lambda' \in \Lambda_y$ other than $\lambda \coloneqq L(\cdot \mid P)$, the supporting hyperplane realized by P (here we allow vertical hyperplanes, for which λ may include infinities). Let $x \in y$ be an outcome where $\lambda'_x < \lambda_x$, which exists by minimality of λ'. Write e_x for the probability distribution that puts all mass on this outcome, and define $P_\alpha \coloneqq (1-\alpha)P + \alpha e_x$ for $\alpha \in (0, 1]$. For each of these points P_α, the hyperplane $L(\cdot, P_\alpha)$ realized by P_α is at most as high as λ' at P_α (because $L(\cdot, P_\alpha)$ is supporting there) and at least as high as λ' at P (where λ' is supporting), so $L(x, P_\alpha)$ is bounded away from λ_x by λ'_x: $L(x, P_\alpha) \leq \lambda'_x < \lambda_x$. Therefore $\lim_{\alpha \downarrow 0} L(x, P_\alpha) \neq L(x, P)$, and L is not continuous. For L proper and continuous, this proves the 'at most one' part of the lemma.

For the 'exactly one' part: by Rockafellar (1970, Theorem 23.3), a nonvertical supporting hyperplane may only fail to exist at P if there is a line segment through P falling inside Δ_y on one side of P and outside on the other; that is, for P on the relative boundary of Δ_y.

Finally, suppose that $\lambda = L(\cdot, P)$ (the hyperplane realized by P) is finite at all $x \in y$ but not minimal. Then by Lemma 6.4, a different minimal supporting hyperplane λ' exists at P, which by the above gives a contradiction. This shows that if λ is finite, it is the minimal supporting hyperplane. □

Proof of Theorem 6.9. Theorem 6.3 applies, showing existence of a KT-vector and a worst-case optimal strategy for the quizmaster.

A worst-case optimal Q^* exists and forms a Nash equilibrium with P^*: For each $y \in \mathcal{Y}$ and each $P \in \Delta_y$, by Lemma 6.8 there is at most one minimal supporting hyperplane at P which is then realized by $Q = P$. So all minimal supporting hyperplanes are realizable on y, and Theorem 6.5 applies.

Next we show that the characterization of P^* and λ^* in Theorem 6.3 is equivalent to the one in this theorem. We consider y with $P^*(y) > 0$ first. If P^* is a worst-case optimal strategy for the quizmaster λ^* defines a supporting hyperplane to $H_L \restriction \Delta_y$ at $P^*(\cdot \mid y)$, then by Lemma 6.4 there exists a minimal $\lambda' \in \Lambda_y$ which is also supporting at $P^*(\cdot \mid y)$ and which satisfies $\lambda'_x \leq \lambda_x$ for $x \in y$, with equality for $P^*(x \mid y) > 0$. By Lemma 6.8, $\lambda'_x = L(\cdot, P^*(\cdot \mid y))$ for all $x \in y$, showing that the conditions of this theorem hold. Conversely, if λ^* satisfies the equality in this theorem, then $\sum_{x \in y} P^*(x \mid y) L(x, P(\cdot \mid y)) = H_L(P^*(\cdot \mid y))$, so λ^* defines a supporting hyperplane at $P^*(\cdot \mid y)$.

For y with $P^*(y) = 0$, if the hyperplane defined by λ^* is nowhere below $H_L \restriction \Delta_y$ as in Theorem 6.3, then using Lemma 6.4 it can be lowered to become a minimal supporting hyperplane, for which a realizing $Q^*_{|y}$ exists; conversely, the existence of a supporting hyperplane to $H_L \restriction \Delta_y$ at $Q^*_{|y}$ that is nowhere above λ^* implies that λ^* is itself nowhere below $H_L \restriction \Delta_y$.

Uniqueness of the KT-vector: For any worst-case optimal strategy P^*, the characterization in this theorem puts an equality constraint on λ^*_x for each x, so only one vector can satisfy these conditions. We just saw that these conditions are equivalent to those in Theorem 6.3, so λ^* is the unique KT-vector.

Characterization of Q^*: By Theorem 6.7, Q^* is worst-case optimal for the contestant if and only if the (unique) KT-vector equals the left-hand side of (6.9). Similarly, if a strategy P^* is worst-case optimal for the quizmaster, then the KT-vector equals the right-hand side of (6.9). Therefore: if for given Q^* a worst-case optimal P^* exists for which (6.9) holds, then both sides equal the KT-vector and Q^* is worst-case optimal; if Q^* is worst-case optimal, then (6.9) holds for all worst-case optimal P^*; and if, for given Q^*, (6.9) holds for all worst-case optimal P^*, then it holds for at least one worst-case optimal P^* by the existence of worst-case optimal P^*. □

Proof of Theorem 6.10. For local L, by definition $L(x, Q) = f_x(Q(x))$ for some sequence of functions $f_x : [0,1] \to [0,\infty]$. Given a point $P \in \Delta_y$, the vector λ given by $\lambda_x = f_x(P(x))$ defines a supporting hyperplane to $H_L \restriction \Delta_y$ at P, because L is proper. Each f_x is nonincreasing. (To see this, consider moving the point P along any line that goes through the vertex of the simplex $\Delta_{\mathcal{X}}$ which puts all mass on some x. Because H_L is concave along this line, the farther away P is from that vertex, the higher a supporting hyperplane to H_L at P will be at that vertex.)

Given P^* and q satisfying the conditions in this theorem, let λ^*_x be $f_x(q_x)$ for each $x \in \mathcal{X}$. We show that P^* is worst-case optimal by verifying that P^* and λ^* satisfy Theorem 6.3. For each y with $P^*(y) > 0$, λ^* defines a supporting hyperplane to $H_L \restriction \Delta_y$ at $P^*(\cdot \mid y)$. For each other y, consider a support-

ing hyperplane to $H_L \restriction \Delta_y$ at $Q_{|y}(x) = q_x / \sum_{x' \in y} q_{x'}$: because $Q_{|y}(x) \geq q_x$, $f_x(Q_{|y}(x)) \leq f_x(q_x) = \lambda^*_x$, the hyperplane defined by λ^* is everywhere at least as high as this supporting hyperplane, as required.

For the converse: For strictly concave H_L, f_x is strictly decreasing. Define functions g_x as follows: $g_x(\lambda_x) = \inf\{q \in [0,1] \mid f_x(q) \leq \lambda_x\}$. If f_x is continuous, g_x is just the ordinary inverse of f_x, but if f_x has a jump discontinuity at q, there will be an interval where g_x is constantly equal to q. In either case, g_x satisfies $g_x(f_x(q)) = q$ for all $q \in [0,1]$.

Take P^* some worst-case optimal strategy, and λ^* a KT-vector. Define $q \in [0,1]^{\mathcal{X}}$ by $q_x = g_x(\lambda^*_x)$. We use Theorem 6.3 to show that q satisfies (6.10). For each $y \in \mathcal{Y}$, let $\lambda' \in \Lambda_y$ be a minimal supporting hyperplane to $H_L \restriction \Delta_y$ that obeys $\lambda' \leq \lambda^*$; such a λ' exists by Lemma 6.4. Let $Q_{|y}$ be the (unique) point at which λ' supports $H_L \restriction \Delta_y$. It satisfies $f_x(Q_{|y}(x)) = \lambda'_x$, from which it follows that $g_x(\lambda'_x) = g_x(f_x(Q_{|y}(x))) = Q_{|y}(x)$. Applying g_x to both sides of $\lambda'_x \leq \lambda^*_x$, we get $Q_{|y}(x) \geq q_x$ for all $x \in y$, so that $\sum_{x \in y} q_x \leq \sum_{x \in y} Q_{|y}(x) = 1$. If $P^*(y) > 0$, then the hyperplane defined by λ^* is itself a supporting hyperplane and $Q_{|y}$ is the point where it touches $H_L \restriction \Delta_y$, namely the point $P^*(\cdot \mid y)$. Because $Q_{|y}(x)$ also satisfies $Q_{|y}(x) = g_x(\lambda^*_x) = q_x$, the equality $q_x = Q_{|y}(x) = P^*(x \mid y)$ follows. □

Proof of Lemma 6.11. At least one vector q must exist because for the game with logarithmic loss and $\mathcal{X}, \mathcal{Y}, p$ as in the lemma, a worst-case optimal strategy P^* must exist by Theorem 6.3, and an associated RCAR vector q must exist for it by Theorem 6.10 using that H_L is strictly concave.

For logarithmic loss, the RCAR vector q and KT-vector λ^* are related by $\lambda^*_x = -\log q_x$. By Theorem 6.3, any strategy $P \in \mathcal{P}$ that does not agree with the KT-vector λ^* is not worst-case optimal, showing that q is unique. □

Proof of Lemma 6.13. We will show that any nonvertical supporting hyperplane $\lambda \in \Lambda_y$ is supporting at no more than one point $P \in \Delta_y$. By Lemma 6.4, if a supporting hyperplane exists at P, then a minimum supporting hyperplane also exists at that point, so it suffices to restrict our attention to minimal $\lambda \in \Lambda_y$. We know that such a λ is realizable on y; let Q be a distribution realizing it. Then Q minimizes the expected loss against any P at which λ supports $H_L \restriction \Delta_y$. For strictly proper L, there can be at most one such P, proving strict concavity. □

Proof of Lemma 6.14. The generalized entropy function of L' is given by

$$H_{L'}(P) = aH_L(P) + \sum_{x \in \mathcal{X}} b_x P(x),$$

where $a \in \mathbf{R}_{>0}$ and $b \in \mathbf{R}^{\mathcal{X}}$ are the constants in the affine transformation (6.12). $H_{L'}$ is again finite and continuous. If P^* is worst-case optimal for the quizmaster in game $\mathcal{G}$, then by Theorem 6.3 there exists a KT-vector λ^* satisfying that theorem's conditions. Define a transformed vector by $\lambda' = a\lambda^* + b$. This is a KT-vector for $\mathcal{G}'$, showing that P^* is also worst-case optimal in that game.

If the conditions of Theorem 6.5 hold for $\mathcal{G}$, then they also holds for $\mathcal{G}'$: If λ' is a minimal supporting hyperplane to $H_{L'} \upharpoonright \Delta_y$, then $\lambda = (1/a)(\lambda' - b)$ is a minimal supporting hyperplane to $H_L \upharpoonright \Delta_y$. By assumption, λ is realizable on y in game $\mathcal{G}$, say by $Q \in \Delta_{\mathcal{X}}$. Then the same Q also realizes λ' in game $\mathcal{G}'$.

If Q^* is worst-case optimal for the contestant in $\mathcal{G}$, then by Theorem 6.7, λ given by $\lambda_x := \max_{y \ni x} L(x, Q^*_{|y})$ is a KT-vector. The transformed vector $\lambda' = a\lambda + b$ is then a KT-vector in $\mathcal{G}'$, so that by Theorem 6.7, Q^* is worst-case optimal in that game.

Because the affine transformation from L into L' can be reversed by a second affine transformation (with $a' = 1/a$ and $b' = -(1/a)b$), the reverse implications follow. □

Chapter 7

Properties of Message Structures in Probability Updating Games

This chapter continues the analysis of probability updating games, which were introduced in Chapter 6. Familiarity with parts of that chapter will be assumed here. In particular, Section 6.2 introduced the problem statement and terminology, and Section 6.5.2 discussed RCAR strategies and RCAR vectors, which will play a central role in this chapter.

7.1 Introduction

Our results in Chapter 6 have focused on properties of the loss function L. However, the characterization theorems in the previous chapter tell us how to *recognize* worst-case optimal strategies, but not how to *find* them efficiently. To progress with this task, we also need to understand a game's message structure $\mathcal{Y}$. That is the motivation behind this chapter. Though the task remains hard to solve in general (for example, if the messages in $\mathcal{Y}$ have different sizes, we cannot do much unless we can reduce the game to a simpler one), we find several interesting results for specific cases.

As we saw in the previous chapter, the loss function and its properties play a large role in the study of worst-case optimal probability updating strategies. In particular, different strategies will in general be worst-case optimal under different loss functions. This is very different for situations where our uncertainty is expressed by a single distribution rather than a set of distributions (the possible quizmaster strategies / coarsening mechanisms). In those situations, the only rational approach to probability updating is naive conditioning, which requires just the original distribution (p) and the message $y \subset \mathcal{X}$ to compute $P(X = x \mid X \in y)$. If however $\mathcal{Y}$ is not a partition of $\mathcal{X}$, then

our uncertainty is expressed by a set of many distributions, and in general we also need to know $\mathcal{Y}$ and L to determine worst-case optimal strategies in our games. However, one of our main results in this chapter shows that for certain classes of message structures, the choice of loss function does not affect the quizmaster's worst-case optimal strategy. In these situations, the procedure of worst-case optimal probability updating becomes more similar to that of naive conditioning, because now it suffices to know just $\mathcal{Y}$ on top of what naive conditioning requires. For these message structures, the distributions of outcomes given messages we derive are in a more general sense 'optimal', expressing what a cautious experimenter *should* believe after receiving new data.

We first show a simple method of simplifying message structures in Section 7.2; there we will also see that if $\mathcal{Y}$ is a partition of $\mathcal{X}$, naive conditioning is worst-case optimal. In Section 7.3, we consider symmetry properties that worst-case optimal strategies must have, provided that the loss function also obeys a form of symmetry defined in Section 7.3.1. Then in Section 7.4, we show two classes of message structures for which the worst-case optimal strategy for the quizmaster can be characterized by the RCAR condition (6.10). This is the condition that also characterizes worst-case optimal strategies for local loss functions and for Kelly gambling with arbitrary payoffs (by Theorem 6.10 and Lemma 6.14); the results in this chapter show that the same characterization sometimes holds for a much more general class of loss functions (as displayed in Figure 7.1). This leads to an interesting property of those (and only those) message structures, discussed in Section 7.4.3: the same strategy P^* will be optimal for the quizmaster regardless of the loss function.

Motivated by the importance and simplicity of the RCAR condition, in Section 7.5 we explore the problem of efficiently computing an RCAR strategy for the quizmaster. Depending on the messages structure, this may still be a hard problem, and we fully solve it only for a small class of message structures $\mathcal{Y}$. We encounter several other classes of messages structures in Sections 7.5.1 and 7.5.3 (also illustrated in Figure 7.1), and find some interesting properties of these classes. The topic of efficient algorithms will be explored more thoroughly in Chapter 8.

We will look at the game from the perspective of the quizmaster, and consider worst-case optimal strategies P^* for him. In games for which a Nash equilibrium exists, the contestant's worst-case optimal strategies can be found easily once we know P^* and a KT-vector certifying its optimality as in Theorem 6.3: given a KT-vector, Q^* can be constructed message-by-message to satisfy the condition in Theorem 6.7. This is even easier in the case of proper loss functions, where for each y with $P^*(y) > 0$, an optimal response is simply $Q^*_{|y} = P^*(\cdot \mid y)$. Another advantage of looking at the game from the quizmaster's side is that our Theorem 6.3 characterizing worst-case optimal P^* requires weaker conditions than Theorem 6.7 characterizing worst-case optimal Q^*.

The proofs of all lemmas and theorems can be found in the appendix at the end of this chapter.

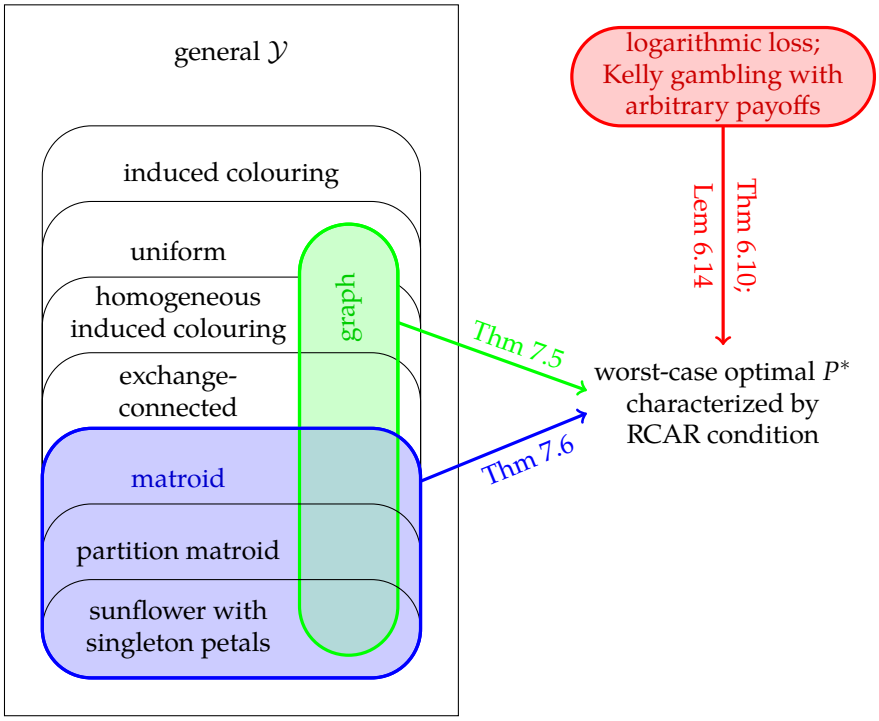


Figure 7.1: Overview of classes of message structures (compare to Table 6.1). One of the main results of this chapter is that for two classes of message structures, regardless of which loss function is used, we have the same RCAR characterization of worst-case optimal strategies for the quizmaster that we saw in Theorem 6.10 for a class of loss functions which includes logarithmic loss. Additionally, Theorem 7.7 shows that the same is not generally true for games with other message structures. The other classes shown in this figure are defined and explored in Section 7.5. (The border around 'homogeneous induced colouring' vanishes behind 'graph' because any graph with a homogeneous induced colouring is also (exchange-)connected.)

7.2 Decomposition of games

For some message structures, regardless of the marginal and loss function, the problem of finding a worst-case optimal strategy for the quizmaster can be solved by considering a smaller message structure instead. It will be useful to look at such simplifications first, so that in the rest of this chapter we will only need to deal with message structures that have already been simplified.

We have already seen one example of the type of result we are looking for earlier, in Lemma 6.2 on page 126, where we saw that if a message is *dominated* by another (meaning that it is a subset of the other), then the quizmaster always has a worst-case optimal strategy that assigns probability 0 to the dominated message.

7.2.1 Decomposition and connected games

Connectivity is a fundamental concept from graph theory. However, in general, our message structures are not graphs, but *hypergraphs*. Like an ordinary graph, a hypergraph is defined by a set of nodes and a set of edges, but the edges are allowed to be arbitrary subsets of the nodes; in a graph, all edges must contain exactly two nodes. Thus for a probability updating game, we can talk about the hypergraph $(\mathcal{X}, \mathcal{Y})$, having the outcomes as its nodes and the messages as its edges.

The terminology of connectivity can be generalized from graphs to hypergraphs (Schrijver, 2003a). We will say that a game is connected if its underlying hypergraph is connected. This leads to the following definitions.

If for some game $\mathcal{G} = (\mathcal{X}, \mathcal{Y}, p, L)$, there is a set $\emptyset \subsetneq S \subsetneq \mathcal{X}$ such that for each message y, either $y \subseteq S$ or $y \subseteq \mathcal{X} \setminus S$, then the game can be *decomposed* into two games $\mathcal{G}_1 = (\mathcal{X}_1, \mathcal{Y}_1, p^{(1)}, L)$ and $\mathcal{G}_2 = (\mathcal{X}_2, \mathcal{Y}_2, p^{(2)}, L)$ with $\mathcal{X}_1 = S$, $\mathcal{X}_2 = \mathcal{X} \setminus S$, $\mathcal{Y}_i = \{y \in \mathcal{Y} \mid y \subseteq \mathcal{X}_i\}$, and $p^{(i)}(x) = p(x) / \sum_{x' \in \mathcal{X}_i} p(x')$. If no such set S exists, we say the game $\mathcal{G}$ is *connected*.

Lemma 7.1 (Decomposition)**.** *If a game $\mathcal{G}$ can be decomposed into $\mathcal{G}_1$ and $\mathcal{G}_2$ as described above, and its loss function L is such that H_L is finite and continuous and $H_L(P) = \inf_{Q \in \Delta_y} \sum_{x \in y} P(x) L(x, Q)$ for each $y \in \mathcal{Y}$ and each $P \in \Delta_y$, then a strategy P^* is worst-case optimal for the quizmaster in $\mathcal{G}$ if and only if there exist worst-case optimal strategies for the quizmaster P_1^* and P_2^* in $\mathcal{G}_1$ and $\mathcal{G}_2$ respectively such that*

$$P^*(x, y) = \begin{cases} P_1^*(x, y) \cdot \sum_{x' \in \mathcal{X}_1} p(x') & \text{for } x \in \mathcal{X}_1; \\ P_2^*(x, y) \cdot \sum_{x' \in \mathcal{X}_2} p(x') & \text{for } x \in \mathcal{X}_2. \end{cases}$$

(The extra condition on L is necessary to exclude some 'very improper' loss functions: those that reward the contestant for predicting outcomes known to have probability 0.) In particular, if the messages of $\mathcal{G}$ form a partition of $\mathcal{X}$, then $\mathcal{G}$ can be decomposed into games that each contain only one message. In a game $\mathcal{G}$ of this form, the quizmaster has only one strategy to choose from. If the loss function is proper, naive conditioning is an optimal response to this strategy, and thus worst-case optimal.

Together with Lemma 6.2, this lemma allows us to reduce any game in which we want to find a worst-case optimal strategy for the quizmaster to a set of connected games containing no dominated messages. These reduced games will not contain any messages of size one, unless one of the games consists of only that message: a message of size one is either dominated, or it forms a trivial component containing no other messages.

7.2.2 Substitution decomposition and modules

In graph theory, the concept of connected components can be generalized to *modules*. A module of a graph is a subset of its nodes such that each node outside the module is either adjacent to all or to none of the nodes in the module (Spinrad, 2003). This concept can be generalized to hypergraphs, for example as in Möhring and Radermacher (1984), by defining a module as a set $\emptyset \subsetneq \mathcal{X}' \subseteq \mathcal{X}$ such that for all $y_1, y_2 \in \mathcal{Y}$, both with $y_i \cap \mathcal{X}' \neq \emptyset$, also $(y_1 \setminus \mathcal{X}') \cup (y_2 \cap \mathcal{X}') \in \mathcal{Y}$. The sets consisting of a single outcome and the set $\mathcal{X}$ itself are always modules, and are called *trivial modules*. Any connected component of a hypergraph is also a module.

The following lemma applies in particular if $\mathcal{X}'$ is a nontrivial module, but also somewhat more generally. However, its application is restricted to logarithmic loss. (It can be extended to other local proper loss functions L with H_L finite and continuous if L is additionally symmetric on $\mathcal{X}'$, as defined in Section 7.3.1.) Thus it will not play as big a role in the rest of this text as the ordinary decomposition lemma.

Lemma 7.2 (Substitution decomposition)**.** *Given a game $\mathcal{G} = (\mathcal{X}, \mathcal{Y}, p, L)$ with L logarithmic loss, and a set $\emptyset \subsetneq \mathcal{X}' \subsetneq \mathcal{X}$, define two new games: the 'inner' game $\mathcal{G}^{in} = (\mathcal{X}^{in}, \mathcal{Y}^{in}, p^{in}, L)$, with*

$$\begin{aligned}
\mathcal{X}^{in} &= \mathcal{X}';\\
\mathcal{Y}^{in} &= \{y \cap \mathcal{X}' \mid y \cap \mathcal{X}' \neq \emptyset\};\\
p^{in}_x &= p_x \Big/ \sum_{x \in \mathcal{X}'} p_x;
\end{aligned}$$

and the 'outer' game $\mathcal{G}^{out} = (\mathcal{X}^{out}, \mathcal{Y}^{out}, p^{out}, L)$, with

$$\begin{aligned}
\mathcal{X}^{out} &= \mathcal{X} \setminus \mathcal{X}' \cup \{x'\};\\
\mathcal{Y}^{out} &= \{y \in \mathcal{Y} \mid y \cap \mathcal{X}' = \emptyset\} \cup \{y \setminus \mathcal{X}' \cup \{x'\} \mid y \in \mathcal{Y}, y \cap \mathcal{X}' \neq \emptyset\};\\
p^{out}_x &= \begin{cases} p_x & \text{for } x \neq x';\\ \sum_{x \in \mathcal{X}'} p_x & \text{for } x = x', \end{cases}
\end{aligned}$$

where $x' \in \mathcal{X}'$ is an arbitrary outcome. Let P^{in}, P^{out} be worst-case optimal strategies for these games, with respective RCAR vectors q^{in}, q^{out}. If for all $y^{in} \in \mathcal{Y}^{in}$, $y^{out} \in \mathcal{Y}^{out}$ with $P^{in}(y^{in}) > 0$, $y^{out} \ni x'$ and $P^{out}(y^{out}) > 0$ in the new games, we have $y^{out} \setminus \{x'\} \cup y^{in} \in \mathcal{Y}$ in the original game, then a worst-case optimal strategy for the

original game is given by

$$P^*(y) = \begin{cases} P^{out}(y) & \text{for } y \cap \mathcal{X}' = \emptyset; \\ P^{out}(y \setminus \mathcal{X}' \cup \{x'\}) \cdot P^{in}(y \cap \mathcal{X}') & \text{otherwise;} \end{cases}$$

and RCAR vector

$$q_x = \begin{cases} q_x^{out} & \text{for } x \notin \mathcal{X}'; \\ q_{x'}^{out} \cdot q_x^{in} & \text{for } x \in \mathcal{X}'. \end{cases}$$

An example of a nontrivial module appeared in the game in Example 6.D on page 129, which has message structure $\mathcal{Y} = \{\{x_1, x_2\}, \{x_2, x_3, x_4\}\}$; there $\mathcal{X}' = \{x_3, x_4\}$ is a module. (Another nontrivial module is $\{x_1, x_3, x_4\}$.) The set $\{x_3, x_4\}$ is the simplest kind of module: there is only one message that intersects it, and so the condition is satisfied trivially. Somewhat more generally, if in a message structure, all messages y that intersect with $\mathcal{X}'$ contain all of $\mathcal{X}'$, then $\mathcal{X}'$ is a module. Applying the lemma to this case tells us that for logarithmic loss, the game can be simplified by *merging the outcomes* in $\mathcal{X}'$ into a single outcome; the original game's worst-case optimal strategy will then distribute the mass on this single outcome among the outcomes in $\mathcal{X}$ proportionally to their marginals. We also saw in Example 6.D that for loss functions other than logarithmic loss, the strategy found this way may not be worst-case optimal.

7.3 Outcome symmetry

Sometimes, the problem of finding a worst-case optimal strategy is simplified because certain 'symmetry' properties of the message structure and loss function allow us to conclude that worst-case optimal strategies satisfying an additional condition must have the same symmetries.

7.3.1 Symmetry of loss functions

We now briefly return to the topic of loss functions to define a property we will need next.

For a probability distribution $Q \in \Delta_{\mathcal{X}}$ and x_1, x_2 distinct elements of $\mathcal{X}$, define $Q^{x_1 \leftrightarrow x_2}$ as

$$Q^{x_1 \leftrightarrow x_2}(x) = \begin{cases} Q(x_2) & \text{for } x = x_1; \\ Q(x_1) & \text{for } x = x_2; \\ Q(x) & \text{otherwise,} \end{cases}$$

and similarly for a contestant's strategy $Q \in \mathcal{Q}$ by applying this transformation to the conditional for each y. We say L is *symmetric between x_1 and x_2* if for all $Q \in \Delta_{\mathcal{X}}$, we have $L(x_1, Q) = L(x_2, Q^{x_1 \leftrightarrow x_2})$ and $L(x, Q) = L(x, Q^{x_1 \leftrightarrow x_2})$ for all $x \in \mathcal{X} \setminus \{x_1, x_2\}$. If L is symmetric between x_1 and x_2 and between x_2 and x_3, then it is also symmetric between x_1 and x_3, because $((Q^{x_1 \leftrightarrow x_2})^{x_2 \leftrightarrow x_3})^{x_1 \leftrightarrow x_2} = Q^{x_1 \leftrightarrow x_3}$. In words: we can apply the first symmetry, then the second, then the

first again to find that we have exchanged x_1 and x_3. We also consider any loss function to be symmetric between x and x for any x. So this symmetry of L is an equivalence relation on $\mathcal{X}$, and we are justified in talking about L being symmetric on sets $S \subseteq \mathcal{X}$, meaning that all pairs of elements of that set can be exchanged. If L is symmetric on $\mathcal{X}$, we say it is *fully symmetric*.

The loss functions we have seen so far were fully symmetric with the exception of the loss function in Example 6.K. The affine transformations of loss functions discussed at the end of Section 6.5.2 may change the symmetries of a loss function, while they do not change which strategies are worst-case optimal for the two players. This means that sometimes, an asymmetric loss functions can be transformed into an essentially equivalent loss function with better symmetry properties. The loss function from Example 6.K cannot be transformed this way. Other loss functions that may exhibit this kind of *inherent* asymmetry are given in the following two examples.

Example 7.A (Matrix loss)**.** Given a $[0,\infty)$-valued $\mathcal{X} \times \mathcal{X}$ matrix of losses A, define *hard matrix loss* by

$$L(x,Q) = \begin{cases} A_{x,x'} & \text{if } Q(x') = 1 \text{ for some } x'; \\ \infty & \text{otherwise.} \end{cases}$$

This generalizes hard 0-1 loss, which is obtained for the matrix A with zeroes on the diagonal and ones elsewhere (except that the definition above may give infinite loss for some Q, but a rational contestant would never use such Q). It is symmetric between x_1 and x_2 if and only if swapping row x_1 with x_2 and column x_1 with x_2 results in matrix A again; that is, if and only if $A_{x_1,x_1} = A_{x_2,x_2}$, $A_{x_1,x_2} = A_{x_2,x_1}$, $A_{x',x_1} = A_{x',x_2}$, and $A_{x_1,x'} = A_{x_2,x'}$, for all $x' \in \mathcal{X} \setminus \{x_1, x_2\}$.

We can also define randomized matrix loss as an analogous generalization of randomized 0-1 loss, by taking an expectation over Q in hard matrix loss:

$$L(x,Q) = \sum_{x' \in \mathcal{X}} Q(x') A_{x,x'}.$$

It has the same symmetry properties as hard matrix loss. The proof of Proposition 6.6 also applies to randomized matrix loss without modification, showing that a Nash equilibrium exists in games using this loss function.

Example 7.B (Skewed logarithmic loss)**.** Fix a vector $c \in \mathbf{R}^{\mathcal{X}}_{\geq 0}$, and define the function $F : \Delta_{\mathcal{X}} \to \mathbf{R}_{\geq 0}$ by

$$F(P) := -\sum_{x \in \mathcal{X}} c_x P(x) \log P(x).$$

This is a sum of differentiable concave functions, and therefore differentiable and concave; if $c \in \mathbf{R}^{\mathcal{X}}_{>0}$, it is strictly concave (in fact, it is also strictly concave if c contains a single 0). We use the construction of *Bregman scores* in Grünwald

and Dawid (2004, Section 3.5.4) to construct a proper loss function L having F as its generalized entropy, and find

$$L(x, Q) = F(Q) + (e_x - Q) \cdot \nabla F(Q) = -c_x(1 + \log Q(x)) + \sum_{x' \in \mathcal{X}} c_{x'} Q(x'),$$

where e_x is the distribution that puts all mass on x. This loss function is strictly proper if H_L is strictly concave. Unlike logarithmic loss and its affine transformations, it is not local for $|\mathcal{X}| > 2$. Also, it is not generally fully symmetric, but is symmetric between pairs of outcomes $x_1, x_2 \in \mathcal{X}$ with $c_{x_1} = c_{x_2}$.

7.3.2 Symmetry of KT-vectors

Using the definition of symmetry of loss functions introduced in the previous section, we can now state the following lemma.

Lemma 7.3 (Loss exchange)**.** *Consider a game with $y_1, y_2 \in \mathcal{Y}$, $y_1 \setminus y_2 = \{x_1\}$, $y_2 \setminus y_1 = \{x_2\}$, H_L finite and continuous and L symmetric between x_1 and x_2. If a worst-case optimal strategy P^* for the quizmaster exists with $P^*(x_1, y_1) > 0$, then all KT-vectors λ^* satisfy $\lambda^*_{x_1} \leq \lambda^*_{x_2}$.*

When two messages $y_1, y_2 \in \mathcal{Y}$ satisfy $y_1 \setminus y_2 = \{x_1\}$ and $y_2 \setminus y_1 = \{x_2\}$, we say that they differ by the *exchange* of one outcome.

In order to find worst-case optimal strategies, we would like to be able to relate $\lambda^*_{x_1}$ to $\lambda^*_{x_2}$ whenever $P^*(y_1) > 0$, but the previous lemma requires something stronger: that $P^*(x_1, y_1) > 0$. We call a strategy P *degenerate* if there exist $y_1, y_2 \in \mathcal{Y}$, $y_1 \setminus y_2 = \{x_1\}$, $y_2 \setminus y_1 = \{x_2\}$ as in the above lemma with $P(y_1) > 0$ and $P(y_2) > 0$ but $P(x_1, y_1) = 0$. Otherwise, P is called *nondegenerate*; then $P(y_1) > 0$, $P(y_2) > 0$ implies $P(x_1, y_1) > 0$, $P(x_2, y_2) > 0$.

We similarly want a term for the symmetry conditions on L that allow us to apply Lemma 7.3 to any pair of messages in some set $\mathcal{Y}' \subseteq \mathcal{Y}$ satisfying the statement of the lemma. We say L is *symmetric with respect to exchanges in* $\mathcal{Y}'$ if L is symmetric between any pair of outcomes x_1, x_2 such that messages $y_1, y_2 \in \mathcal{Y}'$ exist with $y_1 \setminus y_2 = \{x_1\}$ and $y_2 \setminus y_1 = \{x_2\}$.

Lemma 7.4 (Transfer of λ^*)**.** *Consider a game and a worst-case optimal strategy P^* for the quizmaster such that H_L finite and continuous and L symmetric with respect to exchanges in $\{y \in \mathcal{Y} \mid P^*(y) > 0\}$. Then we have one of the following:*

- *If P^* is nondegenerate, then $\lambda^*_{x_1} = \lambda^*_{x_2}$ for all x_1, x_2 such that messages $y_1, y_2 \in \mathcal{Y}$ exist with $y_1 \setminus y_2 = \{x_1\}$, $y_2 \setminus y_1 = \{x_2\}$, $P^*(y_1) > 0$, and $P^*(y_2) > 0$;*
- *If P^* is degenerate, then a nondegenerate worst-case optimal strategy P' exists with $\{y \in \mathcal{Y} \mid P'(y) > 0\} \subsetneq \{y \in \mathcal{Y} \mid P^*(y) > 0\}$.*

Example 7.C (Degenerate P^*)**.** Consider the game

P^*	x_1	x_2	x_3	x_4
y_1	1/8	2/8	–	0
y_2	–	2/8	1/8	0
y_3	0	–	0	2/8
p_x	1/8	4/8	1/8	2/8

with loss function

$$L(x, Q) = \begin{cases} 1 - Q(x) & \text{for } x \neq x_4; \\ 1 & \text{for } x = x_4. \end{cases}$$

This instance of randomized matrix loss (introduced in Example 7.A; here, the loss matrix A has $A_{x,x} = 0$ for $x \in \{x_1, x_2, x_3\}$ and equals 1 elsewhere) gives the contestant no incentive to predict $Q(x_4) > 0$, always assigning him the same loss if that outcome does occur. It is not fully symmetric, but it is symmetric between x_1, x_2 and x_3, so symmetric with respect to exchanges in $\mathcal{Y}$. The strategy P^* given in the table is worst-case optimal, as witnessed by $\lambda^* = (1, 0, 1, 1)$. Though y_1 and y_3 differ by the exchange of one outcome (x_2 for x_3) and have positive probability, we have $\lambda^*_2 \neq \lambda^*_3$; similar for y_2 and y_3 and $\lambda^*_1 \neq \lambda^*_2$.

In this example, the quizmaster has worst-case optimal strategies $P' \neq P^*$ with $P'(y_3) = 0$, dividing the mass $P^*(x_4, y_3)$ among $P'(x_4, y_1)$ and $P'(x_4, y_2)$.

7.4 The RCAR characterization for general loss functions

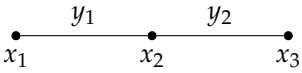
We saw in Theorem 6.10 that for logarithmic loss, worst-case optimal strategies for the quizmaster can be characterized in terms of a simple condition, the *RCAR condition* (6.10). We also saw that sometimes (in Examples 6.B and 6.C on pages 127 and 129, but not in Example 6.D), those same strategies were also worst-case optimal for other loss functions. This suggests that even for some types of games where Theorem 6.10 does not apply, it is possible to recognize worst-case optimal strategies using the easily verifiable RCAR condition. We show that there are two classes of message structures in which this is possible regardless of the marginal, and explore the consequences in Section 7.4.3.

7.4.1 Graph games

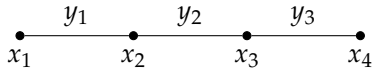
The first of these classes consist of all message structures $\mathcal{Y}$ for which each message contains at most two outcomes. After removing singleton messages (which are either dominated or are decomposable from the rest of the game), we have $|y| = 2$ for all $y \in \mathcal{Y}$. This corresponds to a simple undirected graph (that is, a graph containing no loops or multiple edges) with a node for each outcome in $\mathcal{X}$ and an edge for each message in $\mathcal{Y}$. For this reason, a game where each message in $\mathcal{Y}$ contains at most two outcomes is called a *graph game*.

Many games we saw in the examples in Chapter 6 were graph games. Their underlying graphs are shown in Figure 7.2.

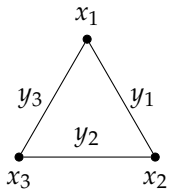
Theorem 7.5 (RCAR for graph games)**.** *If each message in $\mathcal{Y}$ contains at most two outcomes and $P^* \in \mathcal{P}$ is an RCAR strategy, then P^* is worst-case optimal for all L symmetric with respect to exchanges in $\{y \in \mathcal{Y} \mid |y| = 2\}$ with H_L finite and continuous. If additionally H_L is strictly concave, only such P^* are worst-case optimal for L.*



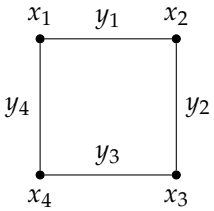
(a) Example 6.B (Monty Hall)



(b) Example 6.C



(c) Examples 6.F and 6.H



(d) Example 6.G

Figure 7.2: Underlying graphs of the graph games seen in Chapter 6

The statement of the theorem is very similar to that of Theorem 6.10 in Section 6.5.2, and the restrictions on L in the present theorem (except for symmetry) were also seen in the previous theorem. Sufficient conditions for these restrictions to hold were given by Lemma 6.1 in Section 6.3 (H_L finite and continuous) and Lemma 6.13 in Section 6.5.2 (strict concavity of H_L).

The intuition behind the proof is that for binary predictions Q, the probability assigned by Q to one outcome determines the probability Q assigns to the other outcome. Thus all loss functions are essentially local when used to assess such predictions, and their behaviour is similar to logarithmic loss.

7.4.2 Matroid games

The other class is that of *matroid games*. A *matroid* over a finite *ground set* $\mathcal{X}$ can be defined by a nonempty family $\mathcal{Y}$ of subsets of $\mathcal{X}$ (the *bases* of the matroid) satisfying the *basis exchange* property (Oxley, 2011, Corollary 1.2.5): for all $y_1, y_2 \in \mathcal{Y}$ and $x_1 \in y_1 \setminus y_2$,

$$(y_1 \setminus \{x_1\}) \cup \{x_2\} \in \mathcal{Y} \text{ for some } x_2 \in y_2 \setminus y_1. \tag{7.1}$$

In words, for any pair of messages, if an outcome that is not in the second message is removed from the first message, it must be possible to replace it by an outcome from the second message that is not in the first message, in such a way that the resulting set of outcomes is again a message.

A matroid game is a game in which $\mathcal{Y}$ is the set of bases of a matroid. The Monty Hall game (Example 6.B) is a matroid game: taking one of the two messages and replacing the outcome unique to it by the only other outcome will result in the other message. By our definition of a game, it is required in addition to (7.1) that each element of the ground set $\mathcal{X}$ of the matroid occurs in some basis.

Many alternative characterizations of matroids exist. For example, a matroid with ground set $\mathcal{X}$ and bases $\mathcal{Y}$ can also be represented by its family of *independent sets* $\mathcal{I} = \{I \subseteq \mathcal{X} \mid I \subseteq y \text{ for some } y \in \mathcal{Y}\}$, and a different set of axioms analogous to (7.1) characterizes whether a given set $\mathcal{I}$ is the family of independent sets of some matroid.

The concept of a matroid was introduced by Whitney (1935) to study the abstract properties of the notion of dependence, as seen for example in linear algebra and graph theory (explained below). Different characterizations of the concept, applied to different examples, were given independently by other authors, but then turned out to be equivalent to matroids. One field where matroids play an important role is combinatorial optimization. We refer to Schrijver (2003b, Section 39.10b) for extensive historical notes.

We give two example classes of matroids, taken from Schrijver (2003b, Section 39.4):

- Given an $m \times n$ matrix A over some vector space, let $\mathcal{X} = \{1, 2, \ldots, n\}$ and $\mathcal{I}$ the family of all subsets I of $\mathcal{X}$ such that the set of column vectors with index in I is linearly independent. Then $\mathcal{I}$ is the family of independent sets of a matroid. A subset that spans the column space of A is a basis of this matroid.

- Given a simple undirected graph G, let $\mathcal{X}$ be its set of edges and $\mathcal{I}$ consist of all acyclic subsets of $\mathcal{X}$. Then $\mathcal{I}$ is the family of independent sets of a matroid. This matroid is called the *cycle matroid* of G. The bases are the maximal independent sets; if G is connected, these are its spanning trees.

One interesting class of games for which $\mathcal{Y}$ are the bases of a matroid is the class of *negation games*. In such a game, each element of $\mathcal{Y}$ is of the form $\mathcal{X} \setminus \{x\}$ for some x. (Not all sets of this form need to be in $\mathcal{Y}$.) Thus the quizmaster will tell the contestant, "The true outcome is *not* x," as in the original Monty Hall problem where one of the three doors is opened to reveal a goat. A family $\mathcal{Y}$ of this form satisfies (7.1) trivially: for y_1, y_2 distinct elements of $\mathcal{Y}$, there is only one choice for each of x_1 and x_2, and with these choices we get $(y_1 \setminus \{x_1\}) \cup \{x_2\} = y_2 \in \mathcal{Y}$.

Another class of matroids is formed by the *uniform matroids*, in which *every* set of some fixed size k is a basis. These also have a natural interpretation when they occur as the message structure of a game: the quizmaster is allowed to leave any set of k doors shut.

As the following theorem shows, matroid games share with graph games the property that RCAR strategies are worst-case optimal for a wide variety of loss functions. Section 7.5 will provide an intuition of why these message structures have this property, and the proof of the theorem uses some terminology introduced in that section.

Theorem 7.6 (RCAR for matroid games)**.** *If $\mathcal{Y}$ are the bases of a matroid and $P^* \in \mathcal{P}$ is an RCAR strategy, then P^* is worst-case optimal for all L symmetric with respect to exchanges in $\mathcal{Y}$ with H_L finite and continuous. If additionally H_L is strictly concave, only such P^* are worst-case optimal for L.*

7.4.3 Loss invariance

We saw in the preceding sections that in graph and matroid games, worst-case optimal strategies for the quizmaster are characterized by the RCAR property. This property does not depend on what loss function is used in the game (though the theorems do put some conditions on the loss function, such as some symmetry requirements). Consequently, in such games, strategies exist that are worst-case optimal regardless of what loss function is used (at least, for a large class of loss functions). We call this phenomenon *loss invariance*.

For such message structures, we can really think of the worst-case optimal strategies as 'conditioning' (as a purely probability-based operation) rather than as worst-case optimal strategies for some game. This conditioning operation can be seen as the generalization of naive conditioning to message structures other than partitions (where naive conditioning gives the right answer). Unlike naive conditioning, which requires just the distribution p and the message y to compute $P(x \mid y)$, we also need the message structure $\mathcal{Y}$ to compute that conditional probability. But like naive conditioning, we do not need to fix a loss function in order to talk about the worst-case optimal prediction of x given a message y.

A subtlety appears when improper loss functions are considered. Our theorems show that the worst-case optimal strategies for the quizmaster are characterized independently of the loss function; however, the worst-case optimal strategies for the contestant will not necessarily coincide with these if the loss function is not proper. In this case, loss invariance tells us that the loss function does not affect *what the contestant should believe* about the true outcome, but it may affect how the contestant translates this belief into a prediction.

In the cases of graph and matroid games, our analysis of worst-case optimal strategies becomes more widely applicable in situations where the probability updating game is really played by two players (as opposed to being a theoretical tool for defining safe updating strategies):

- the same strategies continue to be worst-case optimal if the two players use different loss functions (so that the game is no longer zero-sum);
- both players will be able to play optimally without knowing the loss function(s) in use.

This is true for the Monty Hall game (Example 6.B), which lies in the intersection of graph and matroid games. This provides some justification for the prevailing intuition that the Monty Hall problem should be analysed using probability theory, without mention of loss functions.

Theorems 7.5 and 7.6 apply only to loss functions that are sufficiently symmetric and for which H_L is continuous and finite. We make no claim about the question whether RCAR strategies are also worst-case optimal for loss functions that do not satisfy these properties. However, note that by Lemma 6.14, sometimes affine transformations can be used to convert an asymmetric loss function into a symmetric one without affecting the players' strategies.

Lemma 6.14 also shows that a limited form of loss invariance holds regardless of the message structure. If the players are using different affine transformations of the same loss function (for example, of logarithmic loss; this corresponds to Kelly gambling where the pay-offs for the contestant are different from those for the quizmaster), both players can play optimally without knowing the transformations in use.

An obvious question that remains is: are there any other classes of message structure for which we have loss invariance? This is answered in the negative by the following theorem.

Theorem 7.7. *If a connected game containing no dominated messages is neither a matroid game nor a graph game, then there exists a marginal such that no strategy P for the quizmaster is worst-case optimal for both logarithmic loss and Brier loss.*

7.5 Finding RCAR strategies

We have now seen three situations in which worst-case optimal strategies for the quizmaster can be characterized using the RCAR condition: if L is local and proper (such as logarithmic loss; see Theorem 6.10), if $\mathcal{Y}$ is a graph (Theorem 7.5), and if $\mathcal{Y}$ is a matroid (Theorem 7.6). Thus in order to find a worst-case optimal strategy, it would be helpful to be able to find RCAR strategies.

We also saw in Theorem 7.7 that if $\mathcal{Y}$ is neither a graph nor a matroid, there exist pairs of marginals and loss functions for which RCAR strategies are not worst-case optimal. But this does not hold for all marginals and loss functions, so even if L is not logarithmic loss and $\mathcal{Y}$ is not a graph or a matroid (i.e. if no RCAR theorem holds), it may be worthwhile to look for an RCAR strategy, and check if it is optimal using the results of Chapter 6.

In this section, we establish a computational procedure that tries to find an RCAR strategy given a message structure and a marginal. We do not call this procedure an 'algorithm' because, unless the input satisfies special conditions, it may not be applicable, or give an inconclusive answer. In Chapter 8, we will see algorithms that efficiently find RCAR strategies. However, these algorithms are restricted to the cases of graph and matroid games, and understanding those algorithms does not give an understanding of many of the problems we may run into when looking for RCAR strategies. To gain such understanding, the present section is more useful: in the course of developing our computational procedure, we acquire more insight into why sometimes the RCAR property characterizes worst-case optimal strategies P^*, and what makes graph and matroid games special.

7.5.1 Induced colourings

Fix a set $\mathcal{Y}' \subseteq \mathcal{Y}$ with $\bigcup_{y \in \mathcal{Y}'} y = \mathcal{X}$, and assume that an RCAR strategy P exists with support $\mathcal{Y}_P := \{y \in \mathcal{Y} \mid P(y) > 0\}$ equal to $\mathcal{Y}'$. (For example, we may in many cases take $\mathcal{Y}' = \mathcal{Y}$.) It follows from the RCAR property that $P(x \mid$

$y) > 0$ for all $x \in y \in \mathcal{Y}_P$, so in particular that P is nondegenerate. We will now consider different properties of $\mathcal{Y}_P$ that may help us find P. The classes of message structures defined by these properties, and the inclusion relations between them that we establish here, were shown graphically in Figure 7.1 on page 151, and examples are given in Figure 7.3.

Consider the system of linear equations

$$\sum_{x \in y} q_x = 1 \text{ for all } y \in \mathcal{Y}'. \tag{7.2}$$

A positive solution q of this system shows the existence of an RCAR strategy with support $\mathcal{Y}'$ and RCAR vector q for some marginal p; the nonexistence of such a solution implies that no such strategies exist for any positive marginal. (A similar system is studied in the CAR literature, where it plays a role in characterizing message structures that admit a CAR coarsening mechanism; see Grünwald and Halpern (2003); Jaeger (2005b); Gill and Grünwald (2008). Since we study RCAR rather than CAR, the roles of outcomes and messages are reversed here. We say more about this correspondence in Section 7.6.1, where we relate some of the classes defined below to classes defined in the CAR literature.)

Define a *colouring* as a partition of $\mathcal{X}$. We say a colouring is *induced* by a set of messages $\mathcal{Y}'$ if the system of linear equations (7.2) has at least one solution q with $q_x > 0$ for all x, and x, x' are in the same class of the colouring ('have the same colour') if and only if $q_x = q_{x'}$ for all such solutions to that system. If the system has at least one positive solution, then the colouring induced by $\mathcal{Y}'$ is unique; otherwise, there is no induced colouring.

We say a colouring is *homogeneous on* $\mathcal{Y}'$ if the number of outcomes of each colour is the same for every message in $\mathcal{Y}'$ (for example, if each message consists of one 'red' and two 'blue' outcomes). This is only possible if $\mathcal{Y}'$ is *uniform*: all messages in $\mathcal{Y}'$ have the same size. We are interested in $\mathcal{Y}'$ whose induced colouring is homogeneous. One class of such $\mathcal{Y}'$ that is easy to recognize consists of those $\mathcal{Y}'$ that are *exchange-connected*: for each pair of messages in $\mathcal{Y}'$, there is a path of messages in $\mathcal{Y}'$ (an *exchange-path*) whose adjacent messages differ by the exchange of one outcome as in the conditions of Lemma 7.3 and 7.4.

Figure 7.3 illustrates these definitions with a few examples. The tables are of the same form as those used to display message structures in previous examples, except that the cells now show a colouring instead of a strategy for the quizmaster.

The message structure shown in Figure 7.3a has no induced colouring: any solution of (7.2) must have $q_{x_2} = 0$, so there is no positive solution, and it follows that no RCAR strategy P exists with $P(y) > 0$ for all $y \in \mathcal{Y}'$. On the other hand, any uniform game has an induced colouring, because there is at least one solution to (7.2):

$$q_x = 1/k \quad \text{for all } x \in \mathcal{X}, \tag{7.3}$$

where k is the size of the game's messages.

	x_1	x_2	x_3	x_4	x_5
y_1	$*$	$*$	$*$	$-$	$-$
y_2	$-$	$-$	$*$	$*$	$-$
y_3	$-$	$-$	$-$	$*$	$*$
y_4	$*$	$-$	$-$	$-$	$*$

(a) No induced colouring

	x_1	x_2	x_3	x_4
y_1	$*$	$*$	$-$	$-$
y_2	$-$	$*$	$*$	$*$

(b) Induced colouring but not uniform

	x_1	x_2	x_3	x_4	x_5	x_6
y_1	$*$	$*$	$*$	$-$	$-$	$-$
y_2	$-$	$-$	$*$	$*$	$*$	$-$
y_3	$*$	$-$	$-$	$-$	$*$	$*$

(c) Uniform but induced colouring not homogeneous

	x_1	x_2	x_3	x_4	x_5	x_6
y_1	$*$	$*$	$*$	$-$	$-$	$-$
y_2	$-$	$-$	$*$	$*$	$*$	$-$
y_3	$*$	$-$	$-$	$-$	$*$	$*$
y_4	$-$	$*$	$-$	$*$	$-$	$*$

(d) Homogeneous induced colouring but not exchange-connected

	x_1	x_2	x_3	x_4	x_5
y_1	$*$	$*$	$*$	$-$	$-$
y_2	$-$	$*$	$*$	$*$	$-$
y_3	$-$	$-$	$*$	$*$	$*$

(e) Exchange-connected but not a matroid

	x_1	x_2	x_3	x_4	x_5
y_1	$*$	$*$	$*$	$-$	$-$
y_2	$*$	$*$	$-$	$*$	$-$
y_3	$*$	$-$	$*$	$*$	$-$
y_4	$*$	$-$	$*$	$-$	$*$
y_5	$*$	$-$	$-$	$*$	$*$

(f) Matroid

Figure 7.3: Examples of messages structures and their induced colourings

Figures 7.3b and 7.3c are examples of message structures that do have an induced colouring, but one that is not homogeneous. In both these examples, all outcomes have different colours in the induced colouring, because no pair of outcomes necessarily has the same value of q in a solution of (7.2). The message structure shown in Figure 7.3c will be revisited in Example 7.D in the next section.

The three remaining message structures do have homogeneous induced colourings. Figure 7.3d shows that it is possible for a message structure to have a homogeneous induced colouring without being exchange-connected. In this message structure, which adds the message y_4 to the structure in Figure 7.3c, it is still the case that each pair of messages differs by *two* exchanges. Yet the added message changes the induced colouring: for example, $q_{x_1} = q_{x_4}$ follows because by the equalities from (7.2) on y_1 and y_3, $1 - q_{x_1} = q_{x_2} + q_{x_3} = q_{x_5} + q_{x_6}$, and by y_2 and y_4, $1 - q_{x_4} = q_{x_3} + q_{x_5} = q_{x_2} + q_{x_6}$; thus $2 - 2q_{x_1} = 2 - 2q_{x_4} = q_{x_2} + q_{x_3} + q_{x_5} + q_{x_6}$.

The message structure shown in Figure 7.3e is exchange-connected. For such structures, it easy to determine the induced (homogeneous) colouring: if messages y_1, y_2 differ by the exchange of one outcome (x_1 for x_2), then any

solution of (7.2) must satisfy $q_{x_1} = q_{x_2}$, so such x_1, x_2 must be the same colour. Any vector q that satisfies all these equalities and satisfies $\sum_{x \in y} q_x = 1$ for any one message $y \in \mathcal{Y}'$ satisfies (7.2) for all messages in $\mathcal{Y}'$, so this determines the induced colouring. This colouring is clearly homogeneous on any pair of message that differ by the exchange of one outcome; because exchange-paths exist between all pairs of messages, it follows that the induced colouring of an exchange-connected game is homogeneous.

The structure in Figure 7.3e is not a matroid: there is no outcome in $y_3 \setminus y_1$ that can be added to $y_1 \setminus \{x_2\} = \{x_1, x_3\}$ to make a message. If a message $y_4 = \{x_1, x_3, x_5\}$ is added, the resulting message structure is a matroid.

Finally, the class of matroid games is a subclass of exchange-connected games: (7.1) requires the existence of not just one, but possibly many different exchange-paths between any pair of messages. Negation matroids and uniform matroids were already described in the previous section as examples of matroids; Figure 7.3f shows another example.

The following lemma gives two alternate characterizations of the induced colouring of a matroid. The first of these is in terms of a concept from matroid theory: the colour classes of the induced colouring coincide with the *2-connected components* of the matroid. (We refer to Oxley (2011) for the definition. In matroid theory, these components are usually simply called 'connected components', but we keep the 2 to avoid confusion with the notion of connectedness used in Lemma 7.1.) We observed above (when discussing Figure 7.3e) that if messages exist that differ in the exchange of one outcome, then the outcomes being exchanged must be the same colour. The second characterization shows that for matroids, the converse also holds. Finally, the lemma shows that every colour class of a matroid is a module as defined in Section 7.2.2 (though not every module is a colour class).

Lemma 7.8 (Matroid colouring)**.** *Given a matroid $(\mathcal{X}, \mathcal{Y})$ and two elements $x_1, x_2 \in \mathcal{X}$, the following statements are equivalent:*

1. *x_1 and x_2 are in the same colour class of the induced colouring of $\mathcal{Y}$;*
2. *x_1 and x_2 are in the same 2-connected component of $(\mathcal{X}, \mathcal{Y})$;*
3. *There exist $y_1, y_2 \in \mathcal{Y}$ such that $y_1 \setminus y_2 = \{x_1\}$ and $y_2 \setminus y_1 = \{x_2\}$.*

Further, if $C \subseteq \mathcal{X}$ is a colour class of the induced colouring of $\mathcal{Y}$, then C is a module.

7.5.2 A computational procedure

Consider the case that $\mathcal{Y}'$ induces a homogeneous colouring, and assume as before that an RCAR strategy P exists with $\mathcal{Y}_P = \mathcal{Y}'$. Then the RCAR vector q must be a solution of the linear system (7.2). Additionally, P (and thus q) must agree with the game's marginal p. These constraints allow us to compute the vector q directly.

Let S be the set of all outcomes with a particular colour. Then there is some value q_S such that $P(x \mid y) = q_x = q_S$ for all $x \in S$, $x \in y \in \mathcal{Y}$. Let $k_S = |S \cap y|$ (this is independent of y by homogeneity). We must have

$$k_S q_S = k_S \sum_y P(y) q_S = \sum_{x \in S} \sum_{y \ni x} P(y) P(x \mid y) = \sum_{x \in S} p_x,$$

so that q_S can be computed by

$$q_S = \frac{1}{k_S} \sum_{x \in S} p_x. \tag{7.4}$$

A simple case is when the induced colouring assigns the same colour to all outcomes: then we see that as in (7.3), we get $q_x = 1/k$ for all $x \in \mathcal{X}$, where k is the size of the messages. When a colour consists of just one outcome x (which must then be an element of every message for the colouring to be homogeneous), we find $q_x = p_x$.

If an RCAR strategy P exists with $\mathcal{Y}_P = \mathcal{Y}'$ where $\mathcal{Y}'$ induces a homogeneous colouring, then P must have the vector q given by (7.4) as its RCAR vector. However, it may be the case that no such strategy exists. To find P if it exists, we still need to determine the $P(y)$'s. We can find a nonnegative solution or determine that no nonnegative solution exists by solving the following linear programming problem (which we can do in polynomial time):

$$\begin{aligned} &\text{maximize} && \sum_{y \in \mathcal{Y}} r_y \\ &\text{subject to} && \sum_{y \ni x} r_y \leq \frac{p_x}{q_x} \quad \text{for all } x \in \mathcal{X}, \end{aligned} \tag{7.5}$$

with $r \in \mathbf{R}^{\mathcal{Y}}_{\geq 0}$. If a vector achieving $\sum_{y \in \mathcal{Y}} r_y = 1$ is found, we have a strategy P with r as the marginal on messages ($P(x, y) = q_x r_y$ for all $x \in y$). If no vector r achieves the value 1, there is no RCAR strategy P satisfying the assumption $\mathcal{Y}_P = \mathcal{Y}'$.

Now we may want to apply this procedure in practice to find an RCAR strategy for a given game. (Note that by Lemma 6.11, such a strategy always exists.)

When doing so we encounter two problems: we need to provide the procedure with an $\mathcal{Y}'$ such that $\bigcup \mathcal{Y}' = \mathcal{X}$, and even if we have an idea about what $\mathcal{Y}'$ to take, it may not have a homogeneous induced colouring. Still, let us investigate what happens if we just guess an $\mathcal{Y}'$. We will then encounter one of the cases 1, 2a-2c which we now describe. Briefly, in case 1, the procedure cannot be used because q cannot be determined, and in case 2a and 2b it gives an inconclusive result; in case 2c we have success. We now consider each case in detail.

1. *$\mathcal{Y}'$ has no homogeneous induced colouring.*

In this case, the procedure is not applicable. Indeed, finding an RCAR vector may be a more difficult type of problem, as illustrated by the following example

which uses the message structure from Figure 7.3c. (This example is a uniform game; the class of uniform games is the smallest class among those identified in the previous section that contains the class of games with a homogeneous induced colouring.)

Example 7.D (Irrational RCAR vector)**.** Consider…

P	x_1	x_2	x_3	x_4	x_5	x_6
y_1	$1/10$	$1/10$	$\frac{3}{10} - \frac{1}{10}\sqrt{5}$	–	–	–
y_2	–	–	$\frac{1}{10}\sqrt{5} - \frac{1}{10}$	$1/5$	$\frac{1}{10}\sqrt{5} - \frac{1}{10}$	–
y_3	$1/10$	–	–	–	$\frac{3}{10} - \frac{1}{10}\sqrt{5}$	$1/10$
q_x	$\frac{1}{4} + \frac{1}{20}\sqrt{5}$	$\frac{1}{4} + \frac{1}{20}\sqrt{5}$	$\frac{1}{2} - \frac{1}{10}\sqrt{5}$	$\frac{1}{5}\sqrt{5}$	$\frac{1}{2} - \frac{1}{10}\sqrt{5}$	$\frac{1}{4} + \frac{1}{20}\sqrt{5}$
p_x	$1/5$	$1/10$	$1/5$	$1/5$	$1/5$	$1/10$

…with marginal on the messages $P(y_1) = P(y_3) = \frac{1}{2} - \frac{1}{10}\sqrt{5}$, $P(y_2) = \frac{1}{5}\sqrt{5}$. The vector q is also shown, so the RCAR property can be easily verified. We see that the RCAR strategy P and RCAR vector q (both of which are unique) contain irrational numbers, while the marginal p was rational. The solution techniques used in this section (the formula (7.4) for q and linear optimization for (7.5)) do not yield irrational results when given rational inputs, so this example shows that these techniques will not suffice in general for games that do not have a homogeneous induced colouring.

Conclusion: in this case, an RCAR strategy P with $\mathcal{Y}_P = \mathcal{Y}'$ may exist, but it may be not be easy to find. So in general, for such $\mathcal{Y}'$, we do not know how to efficiently determine if such a P exists.

2. *$\mathcal{Y}'$ does have a homogeneous induced colouring.*

In this case, we can use (7.4) to compute a candidate q for the RCAR vector. We distinguish three subcases:

2a. *If $\mathcal{Y}' \neq \mathcal{Y}$, there may be a message $y \in \mathcal{Y} \setminus \mathcal{Y}'$ for which $\sum_{x \in y} q_x > 1$.*

This may happen because the described procedure ignores the existence of messages not in $\mathcal{Y}'$. However, the RCAR condition (6.10) puts an inequality constraint on $\sum_{x \in y} q_x$ even for messages y with $P(y) = 0$. If the vector q computed by (7.4) does not satisfy this constraint, then q is not an RCAR vector: we chose the wrong $\mathcal{Y}'$.

2b. *No solution r of (7.5) achieves $\sum_{y \in \mathcal{Y}} r_y = 1$.*

This also means that our choice of $\mathcal{Y}'$ was incorrect.

2c. *Otherwise, q is an RCAR vector, and together with r determines an RCAR strategy P.*

In this case, we can report success.

In cases 2a and 2b, $\mathcal{Y}'$ has a homogeneous induced colouring but we find that no RCAR strategy P exists with $\mathcal{Y}_P = \mathcal{Y}'$. Then we may face two problems. First, it is not clear how we might choose a different $\mathcal{Y}'$ on which to try the procedure next. For small message structures, it may be feasible to try all candidates. For larger structures, the number of possible choices grows exponentially, and a more efficient way of searching would be needed.

The second problem is that in general, $\mathcal{Y}'$ might not induce a homogeneous colouring even though $\mathcal{Y}$ does. For example, if $\mathcal{Y}$ is the message structure shown in Figure 7.3e, but there is no RCAR strategy P with $\mathcal{Y}_P = \mathcal{Y}$ for our marginal, we have to conclude that the RCAR strategy must have $\mathcal{Y}_P = \{y_1, y_3\}$ (because this is the only other choice of $\mathcal{Y}'$ that satisfies $\bigcup \mathcal{Y}' = \mathcal{X}$). However, this message structure is no longer exchange-connected, and in fact does not have a homogeneous induced colouring, so that we end up in case 1.

In Section 7.5.3, we will see a subclass of matroid games for which the procedure is guaranteed to succeed for the choice $\mathcal{Y}' = \mathcal{Y}$. So for that class of inputs, the procedure discussed here is an efficient algorithm for finding an RCAR strategy (which is worst-case optimal for any loss function by Theorem 7.6).

In Chapter 8, we will see efficient algorithms for graph games and matroid games. The two algorithms in Section 8.3.8 (graphs) and Section 8.5 (matroids) can also be viewed as instances of the computational procedure in this section: both algorithms essentially compute q and r as we did here; then, if $\sum_{y \in \mathcal{Y}} r < 1$, they pick a new set $\mathcal{Y}'$, guided by properties of the linear optimization problem (7.5). The choice of $\mathcal{Y}'$ is such that each new $\mathcal{Y}'$ is a subset of the previous $\mathcal{Y}'$ (i.e. no backtracking is needed), and such that case 2a will never occur.

Case 1 will never occur either for these algorithms: the chosen $\mathcal{Y}'$ will always have a homogeneous induced colouring. This happens for different reasons for the two cases of graph and matroid games. These reasons shed light on what makes graphs and matroids special as message structures of probability updating games, so we conclude this section by giving brief explanations.

For graphs Any connected component of a graph is additionally exchange-connected, and thus induces a homogeneous colouring. While some choices of $\mathcal{Y}'$ may produce a disconnected graph $(\mathcal{X}, \mathcal{Y}')$, each component of this graph will have a homogeneous induced colouring, and the algorithm can be applied to each of these components recursively. (We saw such a decomposition in Example 6.C on page 129, where $\mathcal{Y}$ was exchange-connected, but the strategy that was worst-case optimal for the three standard loss functions used only two disjoint messages.)

For matroids On a matroid game, for any RCAR strategy P, $\mathcal{Y}_P$ determines a homogeneous colouring. (This colouring is not induced in the usual sense, but is uniquely determined by the equalities on $\mathcal{Y}_P$ combined with inequalities for $\mathcal{Y} \setminus \mathcal{Y}_P$; see the proof of Theorem 7.6 for details.) The conditional probabilities $P(x \mid y)$ respect this colouring. This property is stronger than that of graph games, where each component of $\mathcal{Y}_P$ induces a homogeneous colouring, but $\mathcal{Y}_P$ as a whole might not.

7.5.3 Subclasses of matroid games

We now describe a class of games for which a worst-case optimal strategy can be completely computed using the procedure from the previous section, because regardless of the marginal, no messages will need to be discarded.

A message structure $\mathcal{Y}$ is called a *partition matroid* if $\mathcal{X}$ can be partitioned into nonempty sets $S_1, \ldots, S_k$ such that $\mathcal{Y}$ contains precisely those subsets of $\mathcal{X}$ that take one element from each of the sets S_i (Oxley, 2011). This class forms a subclass of matroids, so if $\mathcal{Y}$ is a partition matroid, it induces a homogeneous colouring. Using Lemma 7.8, it is easy to see that this colouring is given by the sets S_i. An example of a partition matroid is given in Figure 7.4a; the matroid we saw in Figure 7.3f is not a partition matroid.

	x_1	x_2	x_3	x_4	x_5
y_1	*	–	*	–	–
y_2	*	–	–	*	–
y_3	*	–	–	–	*
y_4	–	*	*	–	–
y_5	–	*	–	*	–
y_6	–	*	–	–	*

(a) Partition matroid but not a sunflower

	x_1	x_2	x_3	x_4	x_5
y_1	*	*	*	–	–
y_2	*	*	–	*	–
y_3	*	*	–	–	*

(b) Sunflower with singleton petals

Figure 7.4: More examples of messages structures and their induced colourings

Because a partition matroid induces a homogeneous colouring, we can perform the procedure described in the previous section to find for each x that $q_x = \sum_{x' \in S_i} p_{x'}$, where S_i is the set containing x. Now a solution for the $P(y)$'s that satisfies $\sum_{y \ni x} P(y) q_x = p_x$ always exists:

$$P(y) = \prod_{x \in y} \frac{p_x}{q_x}.$$

In words, this means that given the true outcome x, it is worst-case optimal for the quizmaster to choose a message by randomly sampling an outcome from each set $S_i \not\ni x$ according to the marginal probabilities conditioned on S_i, and give the message consisting of x and these outcomes. The existence of this strategy shows that, for partition matroid games, the procedure always succeeds in finding a worst-case optimal strategy for the choice $\mathcal{Y}' = \mathcal{Y}$.

What does a message Y generated by this strategy tell the contestant about the true (random) outcome X? Clearly, it means that *if* $X \in S_i$ for some i, then X must be the unique outcome in $Y \cap S_i$. Of course, the contestant does not know which of these sets contains X. Write I for the (random) index of the set containing X. Does Y tell the contestant anything about I? The answer is no: For each index i, regardless of whether $I = i$, the outcome in $Y \cap S_i$ will be randomly distributed according to the marginal p conditioned on S_i, independently of $Y \cap S_j$ for $j \neq i$. This implies that Y is independent of I.

Then for each outcome $x \in Y$, the probability that $X = x$ given message Y equals the probability that $I = i$, where i is the index of the set containing x. These are exactly the probabilities that appear in the RCAR vector q. We know from Theorem 7.6 that the same is true also if the quizmaster is using a worst-case optimal strategy different from the one described above.

For more general message structures, the quizmaster may have to discard a message, so that his worst-case optimal strategy cannot be computed so easily:

Theorem 7.9. *If a game induces a homogeneous colouring but is not a partition matroid, then there exist a marginal and a message $y \in \mathcal{Y}$ such that $P(y) = 0$ for all RCAR strategies P.*

We distinguish one subclass of the class of partition matroid games. A message structure in which the intersection of any two messages is constant is called a *sunflower* (Jukna, 2001). The common intersection is called the *core*, and each set difference between a member and the core is called a *petal*. An example of a *sunflowers with singleton petals* is shown in Figure 7.4b. The Monty Hall game itself (Example 6.B) is another example, and the version of the Monty Hall game with 100 doors from Section 1.2 is a sunflower with 99 petals.

If a message structure is a sunflower with singleton petals, it is a partition matroid: each outcome in the core forms a (singleton) class of the partition, and another class contains all the petals. Among partition matroids, sunflowers can be recognized by the property that all of its colour classes except one are singleton outcomes. For this class of games, the strategy P described above is the *unique* RCAR strategy: a strategy P' with $P'(y) \neq P(y)$ for some $y \in \mathcal{Y}$ would disagree with the unique RCAR vector.

The message structure shown in Figure 7.4a is a partition matroid, but not a sunflower. Because at least two of its colour classes are not singletons, such a message structure contains a cycle of four messages in which neighbouring messages differ by the exchange of one outcome, but the pairs of messages on opposite sides of the cycle differ by two outcomes. (Example 6.G on page 134 is the simplest member of this class of message structures, consisting of just this cycle. In Figure 7.4a, there are three such cycles; one is (y_1, y_2, y_5, y_4).) For this type of game, the strategy P found above can be modified by increasing $P(y)$ for two messages at opposite sides of the cycle, and decreasing it by the same amount for the other two, leaving the conditionals unchanged. Thus P is not the unique RCAR strategy. In fact, RCAR strategies exist with $P(y) = 0$ for some $y \in \mathcal{Y}$. For such a strategy P, we have $\mathcal{Y}_P \subsetneq \mathcal{Y}$, but we do still have $\sum_{x \in y} q_x = 1$ even for messages y with $P(y) = 0$.

7.6 Discussion and conclusion

7.6.1 Connections to CAR

Recall from Section 6.1 the CAR condition, which characterizes the set of coarsening mechanisms (i.e. quizmaster strategies) for which naive conditioning is

optimal for the contestant. Part of the literature on CAR also addresses the question of whether, for a given message structure, a quizmaster strategy exists that satisfies the CAR condition. A result of Gill et al. (1997) suggests that this is always possible, but Grünwald and Halpern (2003) clarify that this is true only in a very strict sense: for some message structures, a CAR mechanism only exists if some outcomes get probability 0 (so that arguably, we are really dealing with a different message structure). We saw in Example 6.B on page 114 that this is the case in the Monty Hall problem. Further results were found by Jaeger (2005b) and Gill and Grünwald (2008).

Gill and Grünwald introduce the concept of a *uniform multicover* to characterize *rational CAR mechanisms* (and they also show that all CAR mechanisms are finite mixtures of rational CAR mechanisms). This combinatorial structure is closely related to our uniform games, and to games having an induced colouring.

A uniform multicover of $\mathcal{X}$ is a multiset of nonempty subsets $y \subseteq \mathcal{X}$ such that each $x \in \mathcal{X}$ is contained in exactly k such sets (counting multiplicities). Here k is a constant, the same for all outcomes. The translation to our RCAR case involves switching the roles of outcomes and messages. Define the *dual* of a message structure $(\mathcal{X}, \mathcal{Y})$ by $\mathcal{X}' := \mathcal{Y}$ and $\mathcal{Y}' := \{y'_x \mid x \in \mathcal{X}\}$ where $y'_x := \{y \in \mathcal{X}' \mid y \ni x\}$, again allowing multiplicities in $\mathcal{Y}'$. (This operation corresponds to taking the transpose of the incidence matrix of $\mathcal{Y}$.)

It is easy to see that the dual of a uniform game is a uniform multicover. Conversely, if we take the dual of a uniform multicover, respecting its multiplicities (i.e. for each distinct message in the uniform multicover, the dual contains a number of outcomes equal to that message's multiplicity), we obtain a uniform game, though one that may have duplicate messages. If we discard these messages to conform to our usual definition of a game, where multiplicities were not allowed in the set of messages, the result is still a uniform game. But if we then take the dual a second time, we will retrieve a different uniform multicover from the one we started out with. Thus if we want the operation of taking the dual to be its own inverse and want to restrict ourselves to games without duplicate messages, we must also restrict ourselves to uniform multicovers without duplicate outcomes.

Similarly, we may want to forbid duplicate messages on the side of uniform multicovers. If we take the dual of a uniform multicover but ignore its multiplicities, the resulting game may not be uniform, but it will have an induced colouring: if the multiplicity of each message $y \in \mathcal{Y}$ in the uniform multicover is denoted by n_y, then for its dual $(\mathcal{X}', \mathcal{Y}')$, the RCAR vector $q_{x'} = q_y = n_y/k$ is a positive solution to (7.2). For example, in the dual of the uniform multicover shown in Figure 7.5, q would assign probability $2/3$ to the single outcome corresponding to both messages y_1 and y_2, and $1/3$ to each of the other three outcomes.

A different question is under what conditions on the message structure CAR is *guaranteed* to hold. Grünwald and Halpern (2003, Proposition 4.1) show that this class of message structures is much more limited: this is the case only if $\mathcal{Y}$ is a partition of $\mathcal{X}$. For other message structures, the quizmaster

	x_1	x_2	x_3
y_1	$*$	$*$	$-$
y_2	$*$	$*$	$-$
y_3	$*$	$-$	$*$
y_4	$-$	$*$	$*$
y_5	$-$	$-$	$*$

Figure 7.5: A uniform multicover with a multiple message $y_1 = y_2$. If we take the dual and respect multiplicities, we obtain a uniform game; if we ignore multiplicities, the game obtained is not uniform but does have an induced colouring.

can choose from a nontrivial set of strategies. Among these strategies, the CAR ones and the worst-case optimal ones will in general not coincide, so that naive conditioning may not be worst-case optimal. Whether naive conditioning is a worst-case optimal strategy for given marginal and loss function can easily be checked using our theorems from the previous chapter.

7.6.2 Conclusion

In this chapter, we have seen many classes of message structures, and we found interesting qualitative differences between probability updating games with message structures from different classes. An overview of these results is given in Table 7.1. Possibly the most important of these is the property of loss invariance, shared by graph and matroid games and discussed in Section 7.4.3.

Also, some progress was made in this chapter on the topic of efficiently finding worst-case optimal strategies. First, the results of Section 7.2 can be used to simplify some message structures. If the resulting message structures have homogeneous induced colourings, we may apply the computational procedure of Section 7.5.2; however, this procedure is not guaranteed to produce an answer except in the special case of partition matroids. In Chapter 8, we will see algorithms that efficiently find worst-case optimal strategies for the classes of games for which loss invariance holds: graph and matroid games.

Table 7.1: Results for different classes of message structures

Class	Results	Details
(no induced colouring)	all RCAR strategies must discard a message	page 162
induced colouring	occur as 'duals' of uniform multicovers when ignoring multiplicities	page 170
uniform	duals of uniform multicovers without duplicate outcomes	page 170
homogeneous induced colouring	computational procedure to find RCAR strategy is applicable	Section 7.5.2
exchange-connected	induced colouring easy to find	page 163
matroid	RCAR strategies worst-case optimal; loss invariance; efficient algorithm exists; induced colouring very easy to find; $\mathcal{Y}_P$ (with P RCAR) determines a unique homogeneous colouring	Theorem 7.6 Section 7.4.3 Chapter 8 Lemma 7.8 page 167
partition matroid	an RCAR strategy exists that does not discard any messages; computational procedure to find RCAR strategy always succeeds	page 168 page 168
sunflower with singleton petals	there is a unique RCAR strategy (which does not discard messages)	page 169
graph	RCAR strategies worst-case optimal; loss invariance; efficient algorithm exists	Theorem 7.5 Section 7.4.3 Chapter 8

Appendix 7.A Proofs

Proof of Lemma 7.1. For each $y \in \mathcal{Y}$, assume without loss of generality that $y \in \mathcal{Y}_1$. Then observe that the generalized entropies for $\mathcal{G}$ and $\mathcal{G}_1$ are identical on Δ_y; $P^*(y) > 0$ if and only if $P_1^*(y) > 0$; and $P^*(x \mid y) = P_1^*(x \mid y)$ for all $x \in y$. Now the claim follows from Theorem 6.3. □

Proof of Lemma 7.2. We need to show for the P^* and q constructed in the proof that $\sum_{x \in y} q_x \leq 1$ for each $y \in \mathcal{Y}$, with equality if $P^*(y) > 0$, and that P^* satisfies the marginal constraints.

For each $y \in \mathcal{Y}$,

$$\sum_{x \in y} q_x \;=\; \sum_{x \in y \setminus \mathcal{X}'} q_x^{\text{out}} + q_{x'}^{\text{out}} \cdot \sum_{x \in y \cap \mathcal{X}'} q_x^{\text{in}} \;\leq\; \sum_{x \in y} q_x^{\text{out}} \;\leq\; 1. \tag{7.6}$$

If $P^*(y) > 0$, then one of the following holds: if $y \cap \mathcal{X}' = \emptyset$, then $P^{\text{out}}(y) > 0$; if $y \cap \mathcal{X}' \neq \emptyset$, then $P^{\text{in}}(y \cap \mathcal{X}') > 0$ and $P^{\text{out}}(y \setminus \mathcal{X}' \cup \{x'\}) > 0$. In either case, both inequalities in (7.6) are equalities because P^{in} and P^{out} are RCAR strategies.

Now we must show $q_x \cdot \sum_{y \in \mathcal{Y}, y \ni x} P^*(y) = p_x$ for all x. We have for any $x \in \mathcal{X}'$,

$$\begin{aligned} q_x \cdot \sum_{y \in \mathcal{Y}, y \ni x} P^*(y) \;&=\; q_{x'}^{\text{out}} \cdot q_x^{\text{in}} \cdot \sum_{y \in \mathcal{Y}, y \ni x} P^{\text{out}}(y \setminus \mathcal{X}' \cup \{x'\}) \cdot P^{\text{in}}(y \cap \mathcal{X}') \\ &=\; \Big(q_{x'}^{\text{out}} \cdot \sum_{\substack{y^{\text{out}} \in \mathcal{Y}^{\text{out}}, \\ y^{\text{out}} \ni x'}} P^{\text{out}}(y^{\text{out}})\Big) \cdot \Big(q_x^{\text{in}} \cdot \sum_{\substack{y^{\text{in}} \in \mathcal{Y}^{\text{in}}, \\ y^{\text{in}} \ni x}} P^{\text{in}}(y^{\text{in}})\Big) \;=\; p_{x'}^{\text{out}} \cdot p_x^{\text{in}} \;=\; p_x. \end{aligned}$$

For $x \notin \mathcal{X}'$ (using that if $P^{\text{out}}(y^{\text{out}}) > 0$ and $x' \in y^{\text{out}}$ for some $y^{\text{out}} \in \mathcal{Y}^{\text{out}}$, then for each $y^{\text{in}} \in \mathcal{Y}^{\text{in}}$ with $P^{\text{in}}(y^{\text{in}}) > 0$, $(y^{\text{out}} \setminus \{x'\}) \cup y^{\text{in}} \in \mathcal{Y}$),

$$\begin{aligned} q_x \cdot \sum_{y \in \mathcal{Y}, y \ni x} P^*(y) \;&=\; q_x^{\text{out}}\Big(\sum_{\substack{y \in \mathcal{Y}, y \ni x, \\ y \cap \mathcal{X}' = \emptyset}} P^{\text{out}}(y) + \sum_{\substack{y \in \mathcal{Y}, y \ni x, \\ y \cap \mathcal{X}' \neq \emptyset}} P^{\text{out}}(y \setminus \mathcal{X}' \cup \{x'\}) \cdot P^{\text{in}}(y \cap \mathcal{X}')\Big) \\ &=\; q_x^{\text{out}} \cdot \sum_{\substack{y^{\text{out}} \in \mathcal{Y}^{\text{out}}, \\ y^{\text{out}} \ni x}} P^{\text{out}}(y^{\text{out}}) \;=\; p_x^{\text{out}} \;=\; p_x. \end{aligned}$$

This shows that $P^* \in \mathcal{P}$ and satisfies the RCAR condition. □

Proof of Lemma 7.3. By Theorem 6.3, λ^* is supporting to $H_L \restriction \Delta_{y_1}$ at $P^*(\cdot \mid y_1)$. Define $\lambda^1 \in \Lambda_{y_1}$ equal to λ^* on y_1. Then by Lemma 6.4, any $\lambda' \in \Lambda_{y_1}$ with $\lambda' \leq \lambda^1$ obeys $\lambda'_{x_1} = \lambda^1_{x_1}$.

Again by Theorem 6.3, λ^* is dominating to $H_L \restriction \Delta_{y_2}$. Define λ^2 by $\lambda^2_x = \lambda^*_x$ for $x \in y_1 \cap y_2$, $\lambda^2_{x_1} = \lambda^*_{x_2}$, and 0 elsewhere. Because L is symmetric between

x_1 and x_2, $\lambda^2 \in \Lambda_{y_1}$. For $x \in y_1 \cap y_2$, $\lambda^1_x = \lambda^2_x$. If $\lambda^2_{x_1} \le \lambda^1_{x_1}$, then $\lambda^2 \le \lambda^1$; then we must have $\lambda^2_{x_1} = \lambda^1_{x_1}$. So $\lambda^2_{x_1} < \lambda^1_{x_1}$ is impossible, and we find $\lambda^*_{x_1} = \lambda^1_{x_1} \le \lambda^2_{x_1} = \lambda^*_{x_2}$. □

Proof of Lemma 7.4. If P^* is degenerate, then there exist messages y_1, y_2 such that $y_1 \setminus y_2 = \{x_1\}$ and $y_2 \setminus y_1 = \{x_2\}$, with ${}^*P(y_1) > 0$ and $P^*(y_2) > 0$ but $P^*(x_2, y_2) = 0$. Then Lemma 6.2 shows how to construct a worst-case optimal P' with $P'(y_2) = 0$, but otherwise using the same messages that P^* uses. After a finite number of applications of this procedure, we must terminate with a nondegenerate strategy.

Otherwise P^* is nondegenerate. Then for all x_1, x_2 and y_1, y_2 as above with $P^*(y_1) > 0$ and $P^*(y_2) > 0$, we also have $P^*(x_1, y_1) > 0$ and $P^*(x_2, y_2) > 0$. Then we find $\lambda^*_{x_1} = \lambda^*_{x_2}$ by two applications of Lemma 7.3. □

Proof of Theorem 7.5. For graph games, all loss functions are essentially local. We will make this precise by constructing functions f_x, analogous to those in the proof of Theorem 6.10: they have the property that for all $y \in \mathcal{Y}$, a supporting hyperplane to $H_L \restriction \Delta_y$ at $P \in \Delta_y$ is given by λ with $\lambda_x = f_x(P(x))$ for all $x \in y$. (Note that we may not get $f_x(Q(x)) = L(x, Q)$ as in the case of local proper loss functions in the proof of Theorem 6.10, because the hyperplane realized by Q may not be supporting at Q if L is improper.)

For each $x \in \mathcal{X}$, if the only message in which x occurs is $\{x\}$, then $f_x(q)$ is only defined for $q = 1$, where it is $f_x(1) = H_L(e_x)$ (where e_x is the unique element of Δ_y). For other x, pick any message $y \in \mathcal{Y}$ with $y \ni x$ and $|y| = 2$. For all these y, the generalized entropies $H_L \restriction \Delta_y$ are identical copies of the same function, by symmetry of L. For each $q \in [0,1]$, pick a supporting hyperplane λ to $H_L \restriction \Delta_y$ at the unique $P \in \Delta_y$ with $P(x) = q$, and let $f_x(q) = \lambda_x$. If H_L is not differentiable at P (including when $q \in \{0,1\}$), we can choose a supporting hyperplane arbitrarily as long as the same one is used to define $f_x(q)$ and $f_{x'}(1-q)$ wherever $\{x, x'\} \in \mathcal{Y}$. (In particular this means that if a connected component of $\mathcal{Y}$ viewed as a graph contains an odd cycle, $f_x(1/2)$ must take the same value for all x in that component.)

As for local L, each f_x is nonincreasing because H_L is concave, and f_x is strictly decreasing if H_L is strictly concave. The rest of the proof is the same as for Theorem 6.10. □

Proof of Theorem 7.6. We know from Theorem 6.10 that a quizmaster strategy P^* is worst-case optimal for logarithmic loss if and only if it is RCAR, and from Theorem 6.3 that such a P^* exists. Take any such P^*. Let λ be the KT-vector with respect to logarithmic loss, and $\mathcal{Y}_{P^*} = \{y \in \mathcal{Y} \mid P^*(y) > 0\}$. For any pair $y \in \mathcal{Y}_{P^*}, y' \in \mathcal{Y}$, we will show that there exists a bijection π from $y \setminus y'$ to $y' \setminus y$ such that $\lambda_x \le \lambda_{\pi(x)}$ for all $x \in y \setminus y'$. This follows from Schrijver (2003b, Corollary 39.12a), but here we give a direct proof by induction on $|y' \setminus y|$:

- $|y' \setminus y| = 1$: Apply Lemma 7.3 to $y_1 = y$ and $y_2 = y'$, using that P^* is nondegenerate, to find the required inequality.

- $|y' \setminus y| > 1$: Let $y'_1 = y'$ and pick any $x_1 \in y \setminus y'$. Starting with $i = 1$, apply the basis exchange property on $y \setminus \{x_i\}$ and y'_i to find x'_i (it will be in $y'_i \setminus y \subseteq y'$); then apply it again on $y'_i \setminus \{x'_i\}$ and y to find $x_{i+1} \in y \setminus y'_i$, defining the message $y'_{i+1} = y'_i \setminus \{x'_i\} \cup \{x_{i+1}\}$ (which may not be in $\mathcal{Y}_{P^*}$). Continue until $x_{i+1} = x_1$. Now π defined by $\pi(x_1) = x'_1, \ldots, \pi(x_i) = x'_i$ is a bijection from $\{x_1, \ldots, x_i\} = (y \cap y'_{i+1}) \setminus y' \subseteq y \setminus y'$ to $\{x'_1, \ldots, x'_i\} = y' \setminus y'_{i+1} \subseteq y' \setminus y$ (to see this, note that an element x'_j found in the basis exchange from y is then removed from y'_{j+1} so that it will not be found again; an element x_{j+1} found in the other basis exchange is added to y'_{j+1} with the same result), and for each $1 \leq j \leq i$, applying Lemma 7.3 to y and $y \cup \{x'_j\} \setminus \{x_j\}$ tells us that $\lambda_{x_j} \leq \lambda_{x'_j}$ as required. If $y'_{i+1} = y$, then this is the bijection we are looking for; otherwise, it can be completed by combining it with a bijection from $y \setminus y'_{i+1}$ to $y'_{i+1} \setminus y$, which exists by the induction hypothesis.

If also $y' \in \mathcal{Y}_{P^*}$, a bijection π' from $y' \setminus y$ to $y \setminus y'$ such that $\lambda_{x'} \leq \lambda_{\pi'(x')}$ is found by the same argument. Together, π and π' divide the outcomes in the two sets into disjoint cycles that must all have the same value for λ, defining a colouring of $\mathcal{X}$ that is homogeneous on $\mathcal{Y}_{P^*}$. (Homogeneous colourings are defined in Section 7.5.1.) For logarithmic loss, the RCAR vector q obeys $q_x = e^{-\lambda_x}$, so it must follow the same colouring. Because H_L is strictly concave, the conditionals of P^* must agree with q by Theorem 6.10.

Now take an arbitrary loss function L satisfying the conditions in the theorem, and the same strategy P^*. At an arbitrary message y with $P^*(y) > 0$, choose a supporting hyperplane $\lambda' \in \Lambda_y$ to $H_L \restriction \Delta_y$ at $P^*(\cdot \mid y)$ with $\lambda'_x = \lambda'_{x'}$ wherever x and x' have the same colour: there $P^*(x \mid y) = P^*(x' \mid y)$ and L is symmetric between x and x', so such a supporting hyperplane exists. For all $x, x' \in y$ with $q_x > q_{x'}$ (equivalently, $\lambda_x < \lambda_{x'}$) between which L is symmetric, this λ' satisfies $\lambda'_x \leq \lambda'_{x'}$. (A supporting hyperplane to $H_L \restriction \Delta_y$ at $P^*(\cdot \mid y)$ with $\lambda'_x > \lambda'_{x'}$ would be lower at $(P^*)^{x_1 \leftrightarrow x_2}(\cdot \mid y)$ than at $P^*(\cdot \mid y)$, while by symmetry H_L is the same at those points: a contradiction.)

Because the colouring is homogeneous on $\mathcal{Y}_{P^*}$, the values of λ'_x for $x \in y$ can be copied to all outcomes with the same colour, defining λ' on all of $\mathcal{X}$; for each $y' \in \mathcal{Y}_{P^*}$, λ' defines a supporting hyperplane to $H_L \restriction \Delta_{y'}$ at $P^*(\cdot \mid y')$.

Also, for each $y' \in \mathcal{Y} \setminus \mathcal{Y}_{P^*}$ and $y \in \mathcal{Y}_{P^*}$, we have that a bijection π exists from $y \setminus y'$ to $y' \setminus y$ such that for all $x \in y \setminus y'$, L is symmetric between x and $\pi(x)$, and $\lambda_x \leq \lambda_{\pi(x)}$; then also $\lambda'_x \leq \lambda'_{\pi(x)}$, so λ' defines a dominating hyperplane to $H_L \restriction \Delta_{y'}$. Thus λ' is a KT-vector certifying that P^* is also worst-case optimal for L.

For the converse: If H_L is strictly concave, the supporting hyperplanes defined by a KT-vector λ' each touch H_L at only one point, so that any worst-case optimal strategy P' for the quizmaster must have $P'(x \mid y) = q_x$ for all $x \in y$ with $P'(y) > 0$. Therefore any worst-case optimal P' must be RCAR. □

Proof of Theorem 7.7. We will first show how to construct a vector $q \in \mathbf{R}^{\mathcal{X}}_{>0}$

that satisfies $\sum_{x \in y} q_x \leq 1$ for all $y \in \mathcal{Y}$, and for all $x \in \mathcal{X}$, there is a message $x \in y \in \mathcal{Y}$ with $\sum_{x \in y} q_x = 1$. Then we will determine a marginal so that this vector q is the RCAR vector of the game with that marginal. We will additionally find two intersecting messages, both having sum 1, such that q represents the uniform distribution on one, but not on the other.

Two different constructions are given: one for nonuniform and one for uniform games.

If the game is not uniform, let k_2 be the size of the largest message in $\mathcal{Y}$. By connectedness, there exists a message of size less than k_2 that has nonempty intersection with a message of size k_2. From among such messages, let y_1 be one of maximum size, and let $k_1 < k_2$ be that size. Finally, let y_2 be a messages of size k_2 that maximizes $|y_2 \cap y_1|$. Set initial values for q as follows:

$$q_x = \begin{cases} \frac{1}{k_1} & \text{for } x \in y_1; \\ \frac{|y_1 \setminus y_2|}{|y_2 \setminus y_1|} \cdot \frac{1}{k_1} & \text{for } x \in y_2 \setminus y_1; \\ \frac{1}{|y_2 \setminus y_1|} \cdot \frac{1}{k_1} & \text{otherwise.} \end{cases}$$

Note that the three cases of q_x are listed in nonincreasing order. Now $\sum_{x \in y_1} q_x = \sum_{x \in y_2} q_x = 1$, while $\sum_{x \in y} q_x \leq 1$ for general $y \in \mathcal{Y}$: $\max_x q_x = 1/k_1$, so a message $y \in \mathcal{Y}$ with $|y| \leq k_1$ will have sum at most 1; a message with $|y| = k_2$ will share no more outcomes with y_1 than y_2 does and thus cannot have a larger sum; and because a message with $k_1 < |y| < k_2$ has empty intersection with y_2, the $k_1 - 1$ largest elements of $(q_x)_{x \in y}$ sum to at most $(k_1 - 1)/k_1$, while the fewer than $|y_2 \setminus y_1|$ remaining elements all equal $1/(|y_2 \setminus y_1| \cdot k_1)$ and hence sum to less than $1/k_1$.

A greedy algorithm that repeatedly increments some q_x until none can be increased further, while maintaining the inequality $\sum_{x \in y} q_x \leq 1$ on each y, will terminate with a q satisfying the conditions stated at the beginning of the proof. This q will be unchanged and thus still be uniform on y_1, while on the intersecting message y_2, q also still sums to 1 but is not uniform.

For the case of uniform games, the construction is similar. Let k be the size of the game's messages. By Oxley (2011, Corollary 2.1.5), a nonempty family of sets $\mathcal{Y}$ is the collection of bases of a matroid if and only if for all $y_1, y_2 \in \mathcal{Y}$ and $x_2 \in y_2 \setminus y_1$,

$$y_1 \cup \{x_2\} \setminus \{x_1\} \in \mathcal{Y} \text{ for some } x_1 \in y_1 \setminus y_2. \tag{7.7}$$

Because our $\mathcal{Y}$ is not a matroid, it follows that there exist $y_1, y_2 \in \mathcal{Y}$ and $x_2 \in y_2 \setminus y_1$ for which no corresponding x_1 exists. For $k \geq 3$ (which holds because the game we consider is not a graph game), we claim something stronger: that there exist y_1, y_2, x_2 as above with the additional property that y_1 and y_2 intersect. The proof of this claim is below.

Using such y_1 and x_2 and some $0 < \epsilon < 1/k$, initialize q as follows:

$$q_x = \begin{cases} \frac{1}{k} & \text{for } x \in y_1; \\ \frac{1}{k} + \epsilon & \text{for } x = x_2; \\ \frac{1}{k} - \epsilon & \text{otherwise.} \end{cases}$$

Because any message containing x_2 also contains at least one other outcome not in y_1, we again have $\sum_{x \in y} q_x \leq 1$ for all $y \in \mathcal{Y}$.

For $k \geq 3$, the initial q has the property that the set of outcomes x for which q_x cannot be increased further (we call these outcomes *maximized*) is connected by messages y with $\sum_{x \in y} x = 1$ (that is, the maximized outcomes cannot be partitioned into two nonempty sets such that each sum-1 message is contained in one of these sets); this is because y_1 has sum 1, and any other message with sum 1 must intersect y_1. (For $k = 2$, this would not be the case: the only messages having sum 1 would be y_1 and all messages that contain x_2, but y_1 would not intersect any of these.) We can have the greedy algorithm maintain this as an invariant: Because the game is connected, there is always a message partially in the set of maximized outcomes and partially outside. We call such a message a *crossing* message. Each round, we pick an outcome x that is not maximized yet and is contained in a crossing message; if x_2 is not maximized, we always pick $x = x_2$ (using that it is contained in the crossing message y_2). The tightest constraint on increasing q_x will come from a crossing message, because for any non-crossing message $y \ni x$, we have $\sum_{x' \in y \setminus \{x\}} q_{x'} = (k-1)(1/k - \epsilon)$, which is the smallest possible value of this sum. So increasing q_x as much as possible will cause a crossing message to get sum equal to 1. This message connects x to the set of previously maximized outcomes, and any other outcomes that were maximized by this increment must be contained in some message that also contains x.

When the greedy algorithm terminates, q will still be uniform on y_1, while there will be another message on which q sums to one but is not uniform. (This may not be y_2, which may not have sum 1.) Because all outcomes are connected by sum-1 messages, we can also find a pair of intersecting messages, one of which is uniform and one of which is not. Use these two messages as y_1 and y_2 in the sequel.

Having found, for both nonuniform and uniform games, a vector q and messages y_1 and y_2 as described above, we let strategy P be RCAR with vector q and $P(y)$ uniform on $\{y \mid \sum_{x \in y} q_x = 1\}$. This P is a worst-case optimal strategy for the game with logarithmic loss and marginal $p_x = \sum_{y \ni x} P(x, y)$, and q is its unique RCAR vector.

We will show that P is not worst-case optimal for the game with the same marginal and Brier loss. Brier loss is proper and continuous, so by Theorem 6.9, $L(x, P(\cdot \mid y_1)) = L(x, P(\cdot \mid y_2))$ for worst-case optimal P. These are squared Euclidean distances from a vertex of the simplex to the predicted distribution. However, the equality will not hold for P:

Among all predictions in Δ_{y_i} with $Q(x) = q_x$ for each $x \in y_1 \cap y_2$ (this set of predictions is the intersection of Δ_{y_i} with an affine subspace), the squared Euclidean distance $L(x, Q)$ between such Q and given vertex $x \in y_1 \cap y_2$ is uniquely minimized by Q uniform on the outcomes not in $y_1 \cap y_2$ (this is the orthogonal projection of the vertex onto that subspace). For a uniform game, $P(\cdot \mid y_1)$ is uniform and thus $L(x, P(\cdot \mid y_1))$ equals this minimum; $P(\cdot \mid y_1)$ differs from the uniform distribution at some outcomes not in $y_1 \cap y_2$ and thus $L(x, P(\cdot \mid y_2))$ is larger than the minimum.

For a nonuniform game, $P(\cdot \mid y_1)$ is uniform on $y_1 \setminus y_2$ and $P(\cdot \mid y_2)$ is uniform on $y_2 \setminus y_1$, so both minimize the distance to the vertex in their respective subspaces. However, the subspace for y_2 is isomorphic to a subspace contained in the subspace for y_1 and not containing $P(\cdot \mid y_1)$. Therefore $L(x, P(\cdot \mid y_1)) < L(x, P(\cdot \mid y_2))$. □

Proof of claim. Suppose for a contradiction that any pair of intersecting messages y, y' obeys the above exchange property (7.7) for all $x' \in y' \setminus y$. Let y_1, y_2 be two messages that fail (7.7) for some outcome $x_2 \in y_2 \setminus y_1$; it follows from our assumption that they are disjoint. Because $\mathcal{Y}$ is connected, there exists a sequence of messages starting with y_1 and ending with y_2 in which adjacent messages intersect. Using (7.7), we can extend this sequence to one where adjacent messages differ by the exchange of one outcome: given intersecting $y, y'' \in \mathcal{Y}$ with $d := |y'' \setminus y| > 1$, we find $y' \in \mathcal{Y}$ with $|y' \setminus y| = 1$ and $|y'' \setminus y'| = d - 1$. Write the entire sequence as $y^0 = y_1, y^1, \ldots, y^n = y_2$.

We have $n \geq k$, because $n < k$ would imply that $y_1 \cap y_2 \neq \emptyset$. If $n > k$, we can find a shorter sequence as follows: pick $0 \leq i < j \leq n - k$ for which $y^i \cap y^{j+1} \neq \emptyset$; this holds if $j + 1 - i < k$, so such i, j can always be found if $k \geq 3$. Let x' be the unique outcome in $y^{j+1} \setminus y^j$.

- If $x' \notin y^{j+k}$ (intuitively, adding x' leads us on a detour that can be avoided when going to y^{j+k}): In each of the k exchange steps from y^j to y^{j+k}, one outcome was removed. One of those outcomes was x', which is not in y^j, so at most $k - 1$ outcomes from y^j were removed. Thus y^j and y^{j+k} intersect, and a shorter path between them can be found using (7.7).

- If $x' \in y^{j+k}$ and $x' \in y^i$ (removing x' is the start of a detour): We can use (7.7) to find a shorter path between y^i and y^{j+k}.

- If $x' \in y^{j+k}$ but $x' \notin y^i$ (adding x' is apparently useful, but can be done sooner): Apply (7.7) to messages y^i and y^{j+1} (which intersect) and outcome x' (which is in y^{j+1} but not in y^i) to find a message y' that is one step away from y^i and contains x'. From y', we can find a path to y^{j+k} by (7.7) taking fewer than k steps. Thus we can get from y^i to y^{j+k} in at most k steps.

Thus we can always find a sequence with $n = k$.

Given such a sequence $y^0, y^1, \ldots, y^n$, we will now show a contradiction with the assumption that $y_1 = y^0$ and $y_2 = y^n$ fail (7.7) by showing that for any $x_2 \in y_2$, a message exists that differs from y_1 by adding x_2 and removing one other outcome. If $x_2 \in y^1$, then y^1 is such a message and we are done. Otherwise, we can apply (7.7) to y^1 and $x_2 \in y_2$ to find a message y' containing x_2; because $k \geq 3$, this message still intersects y_1, so applying (7.7) to y_1 and $x_2 \in y'$ gives the message we are looking for. This shows by contradiction that if a connected uniform game with $k \geq 3$ is not a matroid game, there exists a pair of intersecting messages y_1, y_2 and an outcome $x_2 \in y_2 \setminus y_1$ that do not satisfy (7.7). □

Proof of Lemma 7.8. ($2 \Leftarrow 3$) Two elements $x_1 \neq x_2$ of $\mathcal{X}$ are in the same 2-connected component if and only if there is a *circuit* (minimal dependent set) containing both. Since a basis is a maximal dependent set, $y_1 \cup y_2$ is independent. Find a circuit $C \subseteq y_1 \cup y_2$; this circuit contains both x_1 and x_2, as otherwise it would be contained in a basis and thus independent.

($2 \Rightarrow 3$) Let C be a circuit with $\{x_1, x_2\} \subseteq C$; our goal is to find the bases y_1, y_2, which we will do iteratively. Let y_1 be a basis containing the independent set $C \setminus \{x_2\}$, and y_2 a basis containing $C \setminus \{x_1\}$. While $y_1 \setminus \{x_1\} \neq y_2 \setminus \{x_2\}$, pick any $x_1' \in y_1 \setminus (y_2 \cup \{x_1\})$ and use basis exchange to find a basis $y' = (y_1 \setminus \{x_1'\}) \cup \{x_2'\}$ for some $x_2' \in y_2 \setminus y_1$. Note that $x_2' \neq x_2$, as that would result in $C \subseteq y'$. Replace y_1 by y' and repeat until $y_1 \setminus \{x_1\} = y_2 \setminus \{x_2\}$. This process terminates, as the set difference becomes smaller with each step.

($1 \Leftrightarrow 3$) For exchange-connected message structures, the colour classes are the equivalence classes of the transitive reflexive closure of the relation on $\mathcal{X}$ stated in point 3. For matroids, the equivalence of points 2 and 3 shows that this relation is already transitive. Thus for all $x_1 \neq x_2$, points 1 and 3 are equivalent.

Finally, a matroid is equal to the direct sum of its 2-connected components (Oxley, 2011, Corollary 4.2.9), which shows that the 2-connected components (or equivalently, the classes of the induced colouring) are modules. □

Proof of Theorem 7.9. The proof technique is similar to the one used to prove Theorem 7.7: we construct a marginal with the required property by first finding a vector q that is the RCAR vector for some game with the given message structure.

We distinguish two cases. If there exists $y' \subset \mathcal{X}$ that is consistent with the homogeneous induced colouring but $y' \notin \mathcal{Y}$, then pick $0 < \epsilon < 1/(k(k-1))$ and set initial values for q as follows:

$$q_x = \begin{cases} \frac{1}{k} + \epsilon & \text{for } x \in y'; \\ \frac{1}{k} - (k-1)\epsilon & \text{otherwise.} \end{cases}$$

Because each message contains at least one outcome with $q_x = 1/k - (k-1)\epsilon$, we have $\sum_{x \in y} q_x \leq 1$ for all $y \in \mathcal{Y}$.

Otherwise there must exist a colour class $C \subseteq \mathcal{X}$ for which the number of outcomes of this colour occurring in a message is at least two. (If all colours occur exactly once in each message and all $y' \subseteq \mathcal{X}$ consistent with this colouring are $y' \in \mathcal{Y}$, then $\mathcal{Y}$ is a partition matroid.) Then pick any $x^+ \in C$ and $0 < \epsilon < 1/k$, and initialize q according to

$$q_x = \begin{cases} \frac{1}{k} + \epsilon & \text{for } x = x^+; \\ \frac{1}{k} - \epsilon & \text{for } x \in C \text{ but } x \neq x^+; \\ \frac{1}{k} & \text{otherwise.} \end{cases}$$

Again we see $\sum_{x \in y} q_x \leq 1$ for all $y \in \mathcal{Y}$.

We apply the same greedy algorithm we used in the proof of Theorem 7.7: repeatedly increase q_x for some x until none can be increased further, maintaining $\sum_{x \in y} q_x \leq 1$ for all $y \in \mathcal{Y}$. For a vector q obtained by this algorithm, let P be the joint distribution on x, y with $x \in y$ for which the marginal $P(y)$ is uniform on $\{y \in \mathcal{Y} \mid \sum_{x \in y} q_x = 1\}$ and for which $P(x \mid y) = q_x$ for all $x \in y$. This P is an RCAR strategy for the game with marginal $p_x = \sum_{y \ni x} P(x, y)$, and q is the RCAR vector.

In the first case, there must exist some $x^- \in \mathcal{X}$ with $q_{x^-} \leq 1/k$. Let C be the colour class containing x^-, and let x^+ be the unique outcome in $C \cap y'$. In the second case, there must exist some $x^- \in C$ with $q_{x^-} \leq 1/k$. Thus in either case, we have two outcomes x^- and x^+ of the same colour C but with $q_{x^-} \leq 1/k < 1/k + \epsilon \leq q_{x^+}$. Because this contradicts the definition of an induced colouring, there must be a message for which q violates the equality (7.2). This message must be discarded by any RCAR strategy for this game. □

Chapter 8

Algorithms for Probability Updating Games

In the previous chapters, we saw how worst-case optimal strategies for the probability updating game can be recognized. We considered the question of *finding* such strategies in Section 7.5.2, with limited success: the procedure described there still required some degree of trial and error (except for the class of partition matroid games, where it could be guaranteed to succeed). In this chapter, we give algorithms that are guaranteed to find worst-case optimal strategies for two other classes of message structures. These algorithms enable us to solve many more probability updating games (though compared to *all* probability updating games, this is still only a small portion).

The general problem of finding a worst-case optimal strategy is a convex optimization problem. Reasonably efficient algorithms exist for such problems (Boyd and Vandenberghe, 2004), but they approximate the solution rather than computing it exactly: the larger the desired accuracy, the more running time is needed. We observed in Section 7.5.2 that for some message structures, if it is known which of the messages available to the quizmaster receive positive probability in the quizmaster's worst-case optimal strategy, this strategy can be computed exactly using only simple arithmetic operations. If algorithms can be found that efficiently determine which messages should be used, then it should be possible to find a worst-case optimal strategy with running time unaffected by the desired accuracy. However, to compete with general purpose convex optimization algorithms, we need a clever algorithm for finding the set of used messages, as trying all candidate sets quickly becomes infeasible.

The algorithms we investigate in this chapter are *strongly polynomial* (Schrijver, 2003a). A strongly polynomial algorithm finds the exact solution in a number of steps polynomial in the number of elements in the input, regardless of the precision of any numeric elements. We will find such algorithms for the two classes of message structures for which the trial-and-error procedure referenced earlier is guaranteed to work for some subset of the messages. These

are the classes of graph games (where each message contains exactly 2 outcomes; defined in Section 7.4.1) and matroid games (where the messages satisfy the basis exchange property; see Section 7.4.2). A central result of Chapter 7 was that these are precisely the classes for which *loss invariance* holds; see Section 7.4.3 for a discussion of the importance of this concept.

8.1 Introduction

A probability updating game is given by a finite set $\mathcal{X}$, a family $\mathcal{Y}$ of subsets of $\mathcal{X}$ with $\bigcup \mathcal{Y} = \mathcal{X}$, a distribution p on $\mathcal{X}$, and a loss function L. We will not need the loss function in this chapter, but assume that it satisfies the conditions in the theorems of the previous chapter (L is symmetric with respect to exchanges in $\mathcal{Y}$, and H_L is finite and continuous).

A distribution P on the set of pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ with $x \in y$ is a strategy for the quizmaster if $\sum_{y \ni x} P(x, y) = p_x$ for each x. We saw in the previous chapters that for certain probability updating games, a strategy is worst-case optimal for the quizmaster if it satisfies the 'RCAR' condition: for logarithmic loss by Theorem 6.10; for graph games by Theorem 7.5; and for matroid games by Theorem 7.6. We repeat the RCAR condition here:

$$\begin{aligned} &\text{There exists a vector } q \in [0,1]^{\mathcal{X}} \text{ such that} \\ &q_x = P(x \mid y) \text{ for all } x \in y \in \mathcal{Y} \text{ with } P(y > 0), \text{ and} \\ &\sum_{x \in y} q_x \leq 1 \text{ for all } y \in \mathcal{Y}. \end{aligned}$$

The vector q is called the RCAR vector; it exists and is unique by Lemma 6.11. As in the previous chapter, we concentrate on finding a worst-case optimal strategy P for the quizmaster, because once such a strategy is known, worst-case optimal strategies Q for the contestant are easily determined.

The question is: how do we find such P? That is the problem solved by the algorithms in this chapter, for different classes of games. We first consider in Section 8.2 the very limited case of path graph games, where the solution is given by a surprising and intuitive algorithm. Before we can give an algorithm for the more general class of games on arbitrary graphs in Section 8.4, we need to study the properties of a type of network flow, which we do in Section 8.3. An algorithm for matroid games is presented in Section 8.5. Finally, Section 8.6 concludes. All proofs are in the appendix to this chapter.

8.2 Path graphs and the taut string algorithm

Even though the results in this section will be superseded later when we give an algorithm for general graph games, we devote this section to the subclass of graph games whose graph consists of just a single path. The reason is that for such games, the worst-case optimal strategies for the quizmaster turn out to be described by taut strings. This provides both an intuitive image for what is going on, as well as a very efficient (linear time) way of finding these strategies.

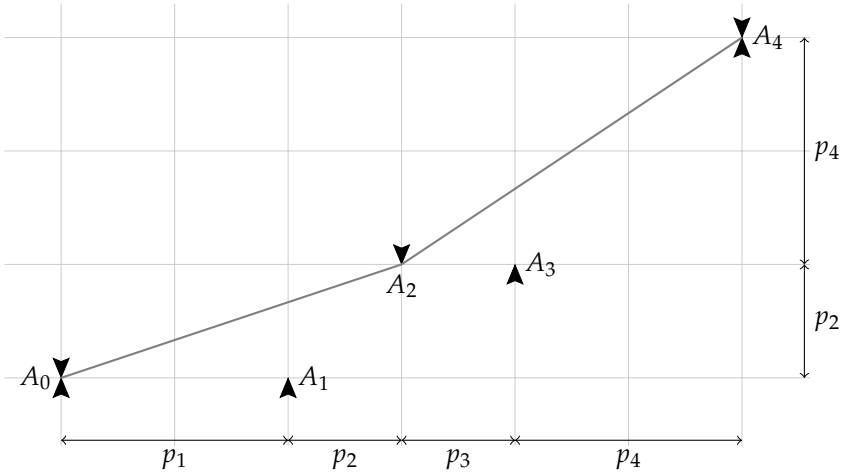


Figure 8.1: The taut string problem corresponding to the path game with $\mathcal{X} = \{1,2,3,4\}$, $\mathcal{Y} = \{y_1, y_2, y_3\}$ with $y_i = \{i, i+1\}$, and marginal on the outcomes $p = (1/3, 1/6, 1/6, 1/3)$. The arrowheads at the points $A_0, \ldots, A_4$ show on what side the string must pass. We see that the string, when pulled taut, touches the point A_2; its slope is $1/3$ to the left of A_2 and $2/3$ to the right.

8.2.1 Correspondence

Consider a graph game where the messages form a path: for the $n \geq 2$ outcomes $\mathcal{X} := \{1, 2, \ldots, n\}$, the messages are $y_1 = \{1,2\}, \ldots, y_{n-1} = \{n-1, n\}$. (A graph of the form $(\mathcal{X}, \mathcal{Y})$ is called a *path graph*.) Then the solution corresponds to that of a *taut string problem*. Imagine a string is constrained to pass above/below certain points (say, pins on a board). Then the string is pulled taut. The taut string will follow the shortest allowed path between its endpoints, going in straight line segments between the points it is pushed against.

Taut strings have been considered in the statistics literature before; see for example Barlow et al. (1972); Mammen and van de Geer (1997); Davies and Kovac (2001), where taut strings appear as a way of defining simple functions approximating noisy regression data.

The taut string problem we are interested in uses the constraining points $A_0, A_1, \ldots, A_n$, with $A_0 = (0,0)$ and

$$A_k = \Big(\sum_{i \leq k} p_i \,,\, \sum_{\substack{i \leq k, \\ i \text{ even}}} p_i \Big) \tag{8.1}$$

for $k \in \{1, \ldots, n\}$. The string must pass through the points A_0 and A_n; above points A_k with k odd; and below A_k for k even. See Figure 8.1 for an example.

Theorem 8.1. *Given a path game, find the solution of the taut string problem described in (8.1). Then a worst-case optimal strategy P for the quizmaster is given by:*

- *For $0 < k < n$ such that the string touches the point A_k, we have $P(y_k) = 0$;*
- *For $0 < k < n$ such that the string does not touch A_k, $P(y_k) = |\delta_k| / (\alpha_k(1 - \alpha_k))$, where δ_k is the vertical distance between A_k and the string, and α_k is the slope of the string at that point;*
- *Also for k such that the string does not touch the point A_k (so $P(y_k) > 0$), the conditional distribution $P(\cdot \mid y_k)$ puts mass on the even outcome in y_k equal to the slope of the string as it passes above or below A_k.*

For the path game and corresponding taut string problem shown in Figure 8.1, we conclude that:

- $P(y_2) = 0$;
- For y_1, $|\delta_1| = 1/9$ (it is two-thirds of $p_2 = 1/6$) and $\alpha_1 = 1/3$, so $P(y_1) = (1/9)/(9/2) = 1/2$. In the same way, we find $P(y_3) = 1/2$.
- Using that the slope of the string above A_1 equals $1/3$, we find $P(2 \mid y_1) = 1/3$ (so $P(1 \mid y_1 = 2/3)$. Above A_3, the slope equals $2/3$, so $P(4 \mid y_3) = 2/3$ and $P(3 \mid y_3) = 1/3$.

A path graph may occur in practice as the message structure of a probability updating game when, for example, a real-valued quantity of interest is reported to us as an integer, but we do not know if the value was rounded up or down. Then outcomes correspond to intervals $(a_i, a_i + 1)$ (we assume that the true value is a.s. not an integer), and messages to unions of two adjacent intervals.

8.2.2 Algorithm

We can now find worst-case optimal strategies for path games efficiently using the taut string algorithm, in $O(n)$ time (Davies and Kovac, 2001). This is clearly much more efficient than using a general purpose convex optimization algorithm. We list the taut string algorithm in Algorithm 8.1; for a more detailed explanation, we refer to Davies and Kovac (2001) and Barlow et al. (1972).

The algorithm keeps track of three sequences of points. These represent piecewise linear functions: K is the solution, G is the *greatest convex minorant* (the pointwise maximum convex function respecting the upper bounds) of the interval of the input that has been read but not added to the solution yet, and S the *smallest concave majorant* of that interval. Each of these sequences is a subsequence of the input points $A_0, \ldots, A_n$. The algorithm operates only on the beginning and end of each of these sequences, so these operations can be implemented efficiently without the aid of complex data structures.

We denote the number of elements in a sequence by $|\mathsf{K}|$, use zero-based indices (so $\mathsf{K}[0]$ is the first element of K), and use negative indices to refer to the end of a sequence: $\mathsf{K}[-1]$ is the last element, $\mathsf{K}[-2]$ the second-to-last, etc. We write $\alpha(A_i, A_j)$ for the slope of the line segment from A_i to A_j, with $i < j$.

Algorithm 8.1: The taut string algorithm

Input: Points $A_0, \ldots, A_n$ constraining the string, sorted from left to right
Output: Sequence of points K where the taut string touches the constraints

Let the sequences G, S, and K each consist of the single point A_0;
for i from 1 to n **do**
  if i is even or $\mathsf{i} = n$ **then**
    // A_i is an upper bound
    Append A_i to G;
    while $|\mathsf{G}| \geq 3$ and $\alpha(\mathsf{G}[-3], \mathsf{G}[-2]) \geq \alpha(\mathsf{G}[-2], \mathsf{G}[-1])$ **do**
      Delete second-to-last point from G;
    end
    // G is now convex
  end
  if i is odd or $\mathsf{i} = n$ **then**
    // A_i is a lower bound
    Append A_i to S;
    while $|\mathsf{S}| \geq 3$ and $\alpha(\mathsf{S}[-3], \mathsf{S}[-2]) \leq \alpha(\mathsf{S}[-2], \mathsf{S}[-1])$ **do**
      Delete second-to-last point from S;
    end
    // S is now concave
  end
  while $|\mathsf{G}| \geq 2$, $|\mathsf{S}| \geq 2$, and $\alpha(\mathsf{G}[0], \mathsf{G}[1]) < \alpha(\mathsf{S}[0], \mathsf{S}[1])$ **do**
    // No straight path remains between G and S
    Remove first point from G and from S;
    if (new) first point in G is to the left of first point in S **then**
      Append first point in G to K, and prepend it to S;
    else
      Append first point in S to K, and prepend it to G;
    end
  end
end
Append A_n to K.

For the points (8.1) used in the taut string problem corresponding to a path game, these slopes are given by $\alpha(A_i, A_j) = \sum_{i<k\leq j, k \text{ even}} p_k / \sum_{i<k\leq j} p_k$.

The same concept (though not the same algorithm) applies to cycle graphs, but now the string must be drawn taut through an infinite sequence of obstacles. These obstacles are given by $A_0 = (0, 0)$ and, for all integers i,

$$A_i - A_{i-1} = \begin{cases} (p_i, 0) & \text{for } i \text{ even;} \\ (p_i, p_i) & \text{for } i \text{ odd,} \end{cases}$$

where all indices of p are specified modulo n. The string must pass below

points A_i with i even, and above points A_i with i odd; there are no points that the string must pass exactly through as in the finite case.

For an even cycle, it is equivalent to consider a string drawn taut around a cylinder (on which the axes are tilted so that A_0 and A_n coincide). For an odd cycle, either we must let the points A_0 and A_{2n} coincide, or the string must be drawn taut in a Möbius strip. We think the second option is far more intriguing.

8.3 Intermezzo: Proportional flows

The algorithm discussed in the previous section is limited in application to games for which $(\mathcal{X}, \mathcal{Y})$ is a path graph. We wish to find an algorithm that can deal with the much larger class of games where $(\mathcal{X}, \mathcal{Y})$ is an arbitrary graph. That is, the messages (edges) may be arranged in any way, but each message must contain exactly two outcomes.

Before we can formulate an algorithm for general graph games and prove it correct, we require a fair amount of graph theory. The present section is devoted entirely to introducing this theory. First, in Section 8.3.1, we consider another physical analogy. This serves as motivation for the definitions studied in subsequent sections, where the general theory is developed. This section leads up to an efficient algorithm in Section 8.3.8.

In Section 8.4, we will find that the problem of finding a worst-case optimal strategy for the probability updating game on a graph can be seen as a special case of the theory developed in this section. Therefore the algorithm from Section 8.3.8 can be used to find these strategies.

8.3.1 Motivating example: Electrical circuits

This section is self-contained, and considers the problem of determining the current flow and voltages in a class of electrical circuits. These circuits consist of a number of resistors in parallel, followed by parallel diodes, followed again by parallel resistors. In Section 8.3.1.1, we state the necessary physical theory and introduce the notation we will use, using a single circuit as a running example. We consider the question of how to go about solving the circuit in Section 8.3.1.2. In Section 8.3.1.3, we generalize from the example circuit to other circuits of the same form.

8.3.1.1 Circuit elements and physical quantities

Consider the electrical circuit given in Figure 8.2. Due to the voltage source at the bottom of the figure, the voltages at s_0 and t_0 are $U_{s_0} = 1\text{V}$ and $U_{t_0} = 0\text{V}$, using the ground in the bottom right as the reference point. The circuit also contains diodes and resistors, which we discuss next.

A *diode* is an electrical circuit element that admits current in only one direction (Tooley, 2006). In our circuit, there are three diodes, between the nodes in $S := \{s_1, s_2\}$ and those in $T := \{t_1, t_2\}$. We will assume these diodes are

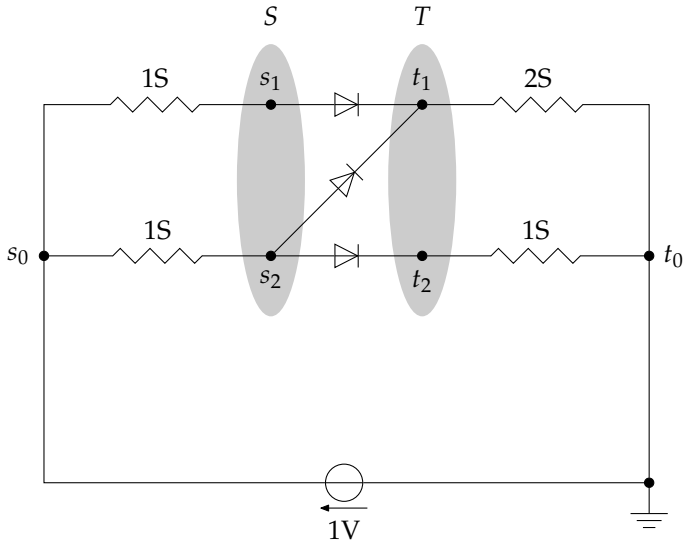


Figure 8.2: An electrical circuit containing resistors and diodes. For each resistor, its conductance is given in the unit siemens (S); this is the inverse of resistance in ohm (Ω).

ideal. In an ideal diode, the resistance is 0 for current flowing in one direction (from S to T), but no current may pass in the opposite direction. We write $A = \{(s_1, t_1), (s_2, t_1), (s_2, t_2)\}$ to describe which pairs of nodes in S and T are connected by a diode in the circuit.

If we think of electricity flowing through a circuit as water flowing through a system of pipes, we can view a diode as a valve that allows water to pass from S to T if the pressure is higher in S than in T, but does not allow water to flow in the backward direction if the pressures are reversed. In the electrical circuit, this role of pressure will be played by voltages.

Our circuit contains four resistors: one attached to each node in $S \cup T$, connecting each of these nodes to s_0 or t_0. The *voltage drop* across a circuit element is the difference in voltages at its endpoints. The voltage drop ΔU (in volt) and current I (in ampere) across a resistor with resistance R (in ohm) are related by Ohm's law: $R = \Delta U / I$. It will be convenient to work with conductance G rather than with resistance R. Conductance is the inverse of resistance, and is measured in the unit siemens. Thus $G = I / \Delta U$.

We associate each resistor with the node in S or T it is attached to, omitting s_0 and t_0 from the notation. So the physical quantities that describe the resistor connected to $v \in S \cup T$ are written as ΔU_v, I_v and G_v.

The final quantity we will need is the current through a diode, which we will denote by $I_{(s,t)}$ for $(s, t) \in A$. Note that due to the diodes, all current flows from s_0 to S, from S to T, and from T to t_0. We measure all currents in this direction, so that the variables we defined all take nonnegative values.

8.3.1.2 Solving the circuit

We wish to compute the voltages at the nodes $S = \{s_1, s_2\}$ and $T = \{t_1, t_2\}$.

Because the resistance of a conducting (ideal) diode is 0, the voltage drop across such a diode is also 0, and so we have $U_s = U_t$ if a positive current $I_{(s,t)}$ is flowing through the diode on the arc $(s,t) \in A$. A voltage difference $U_s \neq U_t$ can only arise if the diode on (s,t) is not conducting. Then we must have $U_s \leq U_t$: through a resistor, this would mean that current was flowing from t to s, but the diode blocks current from flowing in this direction.

Now consider the resistors in our circuit. The voltage drops across these resistors are given by $\Delta U_s = 1 - U_s$ for $s \in S$ and $\Delta U_t = U_t$ for $t \in T$.

Applying Ohm's law to the circuit, we find that the current flowing through the resistor at $s \in S$ is given by $I_s = (1 - U_s)G_s$; for sink nodes, the current through the resistor at $t \in T$ is given by $I_t = U_t G_t$.

By Kirchhoff's current law, the amount of current entering a node must equal the amount leaving that node. Thus $I_v = \sum_{a \ni v} I_a$ for $v \in S \cup T$, where we write $a \ni v$ to mean that v is one of the endpoints of the arc $a \in A$.

We say $C \subseteq S \cup T$ is an *I-component* if it is a connected component of the graph $(S \cup T, \{a \in A \mid I_a > 0\})$. In other words, each pair of nodes in C is connected by a path that traverses only those arcs along which a nonzero current is flowing (but unlike the current, this path may traverse an arc in either direction), and C is a maximal set having this property. An I-component may be a strict subset of an ordinary connected component of $(S \cup T, A)$ if there are sufficiently many diodes along which no current flows.

Because the voltages at the endpoints of a conducting diode must be equal, they must also be constant within every I-component. We write U_C for the common voltage in I-component C. Because no current flows along diodes that enter or leave C, by Kirchhoff's current law, the total current flowing into C (from s_0) and the total current flowing out of C (to t_0) are both equal to the total current flowing through diodes from $C \cap S$ to $C \cap T$. Using Ohm's law, we can express the currents through resistors in terms of the common voltage U_C. Then

$$\sum_{(s,t) \in A : s,t \in C} I_{(s,t)} \;=\; (1 - U_C) \sum_{s \in C \cap S} G_s \;=\; U_C \sum_{t \in C \cap T} G_t,$$

so

$$U_C \;=\; \sum_{s \in C \cap S} G_s \Big/ \sum_{v \in C} G_v. \tag{8.2}$$

The value U_C does not depend on the current I, but holds for all currents for which C is an I-component. With U_C known, the currents passing through the resistors can be computed using Ohm's law. If the components are chosen correctly, a way should exist within each component for the current to flow through the diodes obeying Kirchhoff's current law, and the inequality $U_s \leq U_t$ should hold between different I-components.

In our example circuit, we find that if we take all of $S \cup T$ to be a single component C, we get $U_C = 0.4\text{V}$. Then the currents through the resistors are

$$I_{s_1} = I_{s_2} = G_{s_1}(1 - U_C) = 0.6\text{A}; \qquad I_{t_1} = G_{t_1} U_C = 0.8\text{A}; \qquad I_{t_2} = 0.4\text{A}.$$

There is a unique solution for the currents through the diodes that satisfies Kirchhoff's current law:

$$I_{(s_1,t_1)} = 0.6\text{A}; \qquad I_{(s_2,t_1)} = 0.2\text{A}; \qquad I_{(s_2,t_2)} = 0.4\text{A}.$$

Because there are no other I-components, this demonstrates that this is a correct solution for the given circuit.

Another partition of $S \cup T$ we might consider is into the two components $\{s_1, t_1\}$ and $\{s_2, t_2\}$. Then we find $U_{\{s_1,t_1\}} = 0.33\text{V} < U_{\{s_2,t_2\}} = 0.5\text{V}$. But with these voltages, the diode on the arc (s_2, t_1) should be conducting, so no solution corresponds to this partition of $S \cup T$.

8.3.1.3 A class of similar circuits

A circuit of the same form as in Figure 8.2 can be constructed for any bipartite directed graph with nodes $S \cup T$ and all arcs going from nodes in S to nodes in T, and with a positive real-valued function c on the nodes, which we will call a *capacity function*. To be precise: given such a graph $D = (S \cup T, A)$ and capacity function c, the corresponding electrical circuit has s_0, t_0, voltage source and ground the same as the circuit in Figure 8.2; a node s for each $s \in S$ connected to s_0 by a resistor with conductance $G_s = c_s$; a node t for each $t \in T$ connected to t_0 by a resistor with conductance $G_t = c_t$; and a diode from s to t for each $(s, t) \in A$.

In such a circuit, a solution of U_C and $I_a \geq 0$ is characterized by:

$$U_s \leq U_t \text{ for each } (s,t) \in A, \text{ with equality if } I_{(s,t)} > 0; \tag{8.3}$$

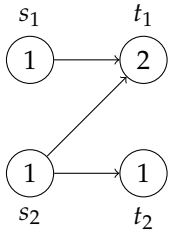
$$I_v = \Delta U_v G_v \text{ for all } v \in S \cup T; \tag{8.4}$$

$$I_v = \sum_{a \ni v} I_a. \tag{8.5}$$

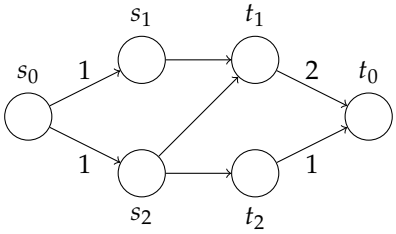
It is not evident from the mathematics that such a solution exists; this will follow from Corollary 8.8 using the more general concept of proportional flows. For some circuits, there may be multiple solutions: if a diode is added from s_1 to t_2 in Figure 8.2, there are several choices for I_a satisfying the above conditions. This happens because the diodes have 0 resistance when conducting; if the circuit were physically realized and non-ideal diodes were used, the current would likely follow some unique solution.

8.3.2 Definitions: Networks and flows

We saw that in order to find the solution of the current in certain electrical circuits, we need to find the right partition of the nodes in the circuit into components in which the voltage is constant. For large circuits, it is clearly computationally infeasible to try out all possible partitions. As a first step towards finding a more efficient algorithm, we translate the problem from the language of electrical circuits to that of flow networks. (The algorithm we will find also



(a) Network (D, c): capacities on nodes



(b) Augmented network (D', c'): capacities on arcs

Figure 8.3: The network (D, c) from which the electrical circuit in Figure 8.2 was constructed, and the corresponding augmented flow network (D', c').

solves the problem of finding worst-case optimal strategies in the probability updating game on a graph, but we will not be able to show this until Section 8.4.)

Consider again a directed graph $D = (S \cup T, A)$ with $A \subseteq S \times T$ and capacities $c : S \cup T \to \mathbf{R}_{>0}$, as were used in Section 8.3.1.3 to construct the class of electrical circuits similar to the one in Figure 8.2. We will call such a pair (D, c) a *network*. Given a network (D, c), define the *augmented network* (D', c') with $D' = (V', A')$, where $V' = S \cup T \cup \{s_0, t_0\}$ and $A' = A \cup \{(s_0, s) \mid s \in S\} \cup \{(t, t_0) \mid t \in T\}$, and with capacities c' on the arcs (rather than on the nodes as in the definition of c) given by

$$c'(v_1, v_2) = \begin{cases} c_{v_2} & \text{for } v_1 = s_0 \text{ and } v_2 \in S; \\ c_{v_1} & \text{for } v_1 \in T \text{ and } v_2 = t_0; \\ \infty & \text{otherwise} \end{cases}$$

(see Figure 8.3 for an illustration). The nodes in S are called *sources* and those in T *sinks*; node s_0 is the *supersource* and t_0 the *supersink*. The capacities on the sources $c_s = c'(s_0, s)$ are *supplies*, and the capacities on the sinks $c_t = c'(t, t_0)$ *demands*. The augmented network can be used to model how resources may be transferred from the set of sources to the set of sinks. The nodes s_0 and t_0 are added merely because it is technically convenient.

Following Schrijver (2003a), we say a function $f : A' \to \mathbf{R}_{\geq 0}$ is a *flow from* s_0 *to* t_0 if it obeys the *flow conservation law*:

$$\sum_{v' \in V' : (v', v) \in A'} f_{(v', v)} = \sum_{v' \in V' : (v, v') \in A'} f_{(v, v')} \qquad \text{for all } v \in V' \setminus \{s_0, t_0\} = V. \quad (8.6)$$

That is, except at the supersource and supersink, the amount of resources entering a node must be equal to the amount leaving it. Note that this definition allows flows that exceed the capacities c; we will indeed also consider some flows of this kind.

For an electrical circuit and an augmented flow network based on the same (D, c), the current I in the electrical circuit can be seen as a function from A' to $\mathbf{R}_{\geq 0}$, like the flow f from s_0 to t_0 on D'. Furthermore, Kirchhoff's current law (8.5) is equivalent to (8.6). Thus, a current I in the electrical circuit is also a flow from s_0 to t_0.

Given a flow f, we will write $f_s := f_{(s_0,s)}$ for $s \in S$ and $f_t := f_{(t,t_0)}$ for $t \in T$, as we did for currents I. If f is a flow from s_0 to t_0 on the augmented network (D', c'), we also say f is a flow on the original network (D, c); this way, we do not need to refer to the artificial extra nodes s_0 and t_0 any more.

We see there is a clear correspondence between the flow and the electrical current. The notion of voltage, on the other hand, is absent in the standard definition of flow from the literature. We will extend the analogy using the following new definitions:

$$\textit{utilization } u_v := f_v / c_v; \tag{8.7}$$

$$\textit{supply proportion } q_C := \frac{\sum_{s \in S \cap C} c_s}{\sum_{v \in C} c_v}, \tag{8.8}$$

where C is an *f-component*: a connected component of the graph $(S \cup T, \{a \in A \mid f_a > 0\})$. (This definition is identical to the one for currents in Section 8.3.1.2.) We also define $q_v := q_C$ for $v \in C$. We see that for any network (D, c), if we take the flow f equal to the current I in the corresponding electrical circuit, these new quantities are equal to quantities from the electrical circuit: $u_v = \Delta U_v$ by (8.4), and $q_C = U_C$ by (8.2). The correspondence is summarized in Table 8.1.

Table 8.1: Relation between electrical circuits and flow networks

Electrical circuit		Flow network	
Conductance	G_v	Capacity	c_v
Current	I_a	Flow	f_a
Voltage	U_C	Supply proportion	q_C
Voltage drop	ΔU_v	Utilization	u_v

8.3.3 Definitions: Proportional and maximum flows

It may seem that all we have done in the previous section is assigning new names to concepts from electrical circuits. However, a crucial difference is in the relation between flow and capacity. While the conductances in an electrical circuit put very specific requirements on the current, the capacity does not occur in the general definition of flow. In this section, we define two special kinds of flows, which are constrained by the capacities. The first of these, the proportional flow, is a novel definition that generalizes flows corresponding to electrical currents. The second is the standard definition of maximum flow, in

which the capacities are hard limits on the flow through an arc. Both definitions will be crucial in the development of the algorithm, which explains the need to introduce general flows in the previous section. The relations between currents, proportional flows, and maximum flows will be investigated further in Section 8.3.4.

Definition 8.1. A flow f on network (D, c) is *proportional* if

- for each f-component C, there are constants α_C, β_C such that $u_s = \alpha_C$ for all $s \in C \cap S$ and $u_t = \beta_C$ for all $t \in C \cap T$;
- for each arc $(s, t) \in A$, the supply proportions (8.8) obey $q_s \leq q_t$.

In this definition, the capacities c play a role of relative weights, not of absolute constraints on the flow. We see that if a flow f is equal to a current in the electrical circuit corresponding to (D, c), then f is proportional: α and β are constant within each f-component because U is, and (8.3) implies $q_s \leq q_t$ for $(s, t) \in A$.

Interpreting the flow as an allocation of resources, intuitively, the second requirement in the definition expresses that if the supply proportion is larger in f-component C_1 than in f-component C_2, then it would be 'unfair' if an opportunity to pass resources from C_1 to C_2 exists, but goes unused — even if supply is short (or long) in both components. We will briefly relate proportional flows to some other notions of fairness from the economic literature in Section 8.3.7.

If an f-component C of a proportional flow f consists of a single node v, we get $q_C = 1$ if $v \in S$, and $q_C = 0$ if $v \in T$; if C is a nonsingleton f-component, $q_C \in (0, 1)$. Due to the second condition in the definition, a proportional flow f on network (D, c) is not allowed to have nodes with zero flow leaving/entering them, unless they are isolated nodes in D.

For a proportional flow, the supply proportion q_C defined in (8.8) can be expressed in terms of α_C and β_C:

$$q_C = \frac{\sum_{s \in S \cap C} c_s}{\sum_{t \in T \cap C} c_t + \sum_{s \in S \cap C} c_s} = \frac{\frac{\sum_{t \in T \cap C} f_t}{\sum_{t \in T \cap C} c_t}}{\frac{\sum_{s \in S \cap C} f_s}{\sum_{s \in S \cap C} c_s} + \frac{\sum_{t \in T \cap C} f_t}{\sum_{t \in T \cap C} c_t}} = \frac{\beta_C}{\alpha_C + \beta_C},$$

using that $\sum_{s \in S \cap C} f_s = \sum_{t \in T \cap C} f_t$.

So far, we have not used the capacity function c' in its usual role as a hard upper limit on the amount of flow through each arc. We say f *is under* c' if $f_a \leq c'(a)$ for each arc $a \in A'$. Using our notation with f_s and f_t, this constraint is equivalent to $f_v \leq c_v$ for all $v \in S \cup T$, so we will write it as f *is under* c. (Note that no upper bound is imposed on arcs between S and T, where c' is infinite.) This condition is equivalent to $u_v \in [0, 1]$ for all $v \in S \cup T$.

A flow is *maximum* if it is under c, and no other flow under c has larger $\sum_{s \in S} f_s = \sum_{t \in T} f_t = \sum_{a \in A} f_a$. In general, the concepts of proportional flow and maximum flow are independent: a flow may be neither, both, or either one. If a proportional flow is also a maximum flow, then each f-component

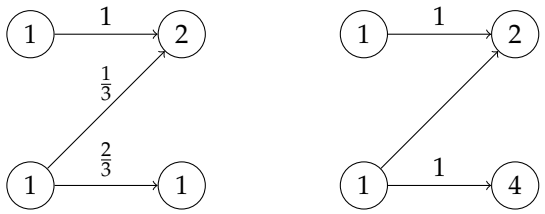


Figure 8.4: Two flow networks with the same underlying directed graph but different capacities, with their proportional maximum flows

will have $\max(\alpha_C, \beta_C) = 1$. With this restriction, the mapping from (α_C, β_C) to q_C is invertible; its inverse is given by

$$(\alpha_C, \beta_C) = \begin{cases} (1, \frac{q_C}{1-q_C}) & \text{if } q_C \leq \frac{1}{2}; \\ (\frac{1-q_C}{q_C}, 1) & \text{if } q_C > \frac{1}{2}. \end{cases} \tag{8.9}$$

We say a node v is *saturated* by a flow if $u_v = 1$.

Two examples of proportional maximum flows are shown in Figure 8.4. In Figure 8.4a, the unique proportional maximum flow has one f-component $C = S \cup T$ with $\alpha_C = 1$ and $\beta_C = 2/3$, so $q_C = 2/5$. In Figure 8.4b, the capacity of the bottom sink has been increased, and a different (again unique) proportional maximum flow is found. This flow has two f-components. The maximum flow which does not use the sink with value 4 is not proportional: that node is a singleton f-component and thus has supply proportion 0, while the other f-component has supply proportion $1/2$; this violates the inequality on the bottom arc.

8.3.4 Componentwise rescaling of flows

Consider the following operation on a flow f: choose an f-component C, and multiply the flow along all arcs within C by some positive constant. We can also perform this operation on several f-components simultaneously, possibly using different constants for each f-component. The resulting flow (and also the operation itself) is called a *componentwise rescaling* of f.

The definition of proportional flows imposes constraints on the utilizations of the nodes within each f-component of the flow, and on the supply proportions among different f-components. These constraints are preserved by componentwise rescaling: the componentwise rescaling of a proportional flow will again be a proportional flow (but with different α and β). Conversely, if a flow is not proportional, then neither are its componentwise rescalings.

We saw in the previous section that every current is a proportional flow. The concept of componentwise rescaling shows that currents take a special place among a network's proportional flows: Given a proportional flow f in a flow network, a solution of the corresponding electrical circuit can be found by

componentwise rescaling, as follows. Take the current I to be the componentwise rescaling of f in which each nonsingleton f-component has $\alpha_C + \beta_C = 1$. The flow conservation law (8.6) is equivalent to Kirchhoff's current law (8.5), and we know from the remarks below equations (8.7) and (8.8) that for this current, we must have $\Delta U_v = u_v$ and $U_C = q_C$ (these values satisfy (8.4) and the equality in (8.3)). For the chosen componentwise rescaling, this is consistent with the definition of voltage drops in terms of voltages: for any source $s \in S$, it gives $\Delta U_s = 1 - U_s = \alpha_s$, and for any sink $t \in T$, it gives $\Delta U_t = U_t = \beta_t$. Finally, the inequality in (8.3) is now equivalent to the inequality in the definition of proportional flows. This shows that this componentwise rescaling of f gives a solution of the electrical circuit. For an example, observe that the solution for the electrical current found in Section 8.3.1.2 is a componentwise rescaling of the proportional flow for the same network shown in Figure 8.4a.

Comparing currents to maximum flows, we find that a current is under c, but is not a maximum flow: in each nonsingleton f-component C, both $\alpha_C < 1$ and $\beta_C < 1$. The following theorem establishes a relation between maximum and proportional flows similar to the relation between currents and proportional flows, but more powerful.

Theorem 8.2 (Weighting characterization of proportional flows)**.** *Given a flow f on a network (D, c), that flow is proportional if and only if for all $w > 0$, the network $(D, c^{(w)})$ with modified capacities given by*

$$c_v^{(w)} = \begin{cases} c_v & \text{for } v \in S, \\ w \cdot c_v & \text{for } v \in T, \end{cases} \tag{8.10}$$

has a maximum flow that is a componentwise rescaling of f.

In particular (for $w = 1$), any proportional flow can be rescaled componentwise to be a maximum flow. Obviously, the (unique) componentwise rescaling that turns f into a maximum flow is the one that rescales each nonsingleton f-component C so that it satisfies $\max(\alpha_C, \beta_C) = 1$. We call this the *maximum componentwise rescaling* of f.

8.3.5 The capacitated Edmonds-Gallai decomposition

Now that we are done talking about voltages, it will be useful to use the letter V for the full set of nodes in the (nonaugmented) network: $V := S \cup T$.

Definition 8.2. Given a network (D, c) with $D = (V, A)$, and a maximum flow f on that network, let $(V_<, V_=, V_>)$ be a partition of V such that:

- $V_=$, $V_< \cap S$ and $V_> \cap T$ contain no unsaturated nodes;
- Nodes in the same f-component are in the same class of the partition;
- There are no arcs in D from $(V_> \cup V_=) \cap S$ to $V_< \cap T$, and no arcs from $V_> \cap S$ to $(V_< \cup V_=) \cap T$.

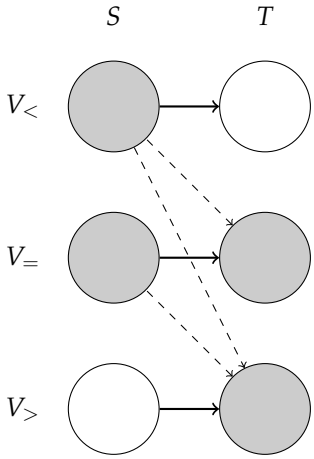


Figure 8.5: Schematic of the capacitated Edmonds-Gallai decomposition. Each source and sink of a network belongs to one of the six circles in this diagram: Grey circles contain only saturated nodes, white circles may contain both saturated and unsaturated nodes; solid arrows represent where positive flow may be; and dashed arrows represent that there might be arcs, but with zero flow.

(See Figure 8.5 for an illustration.) Such a partition with $V_=$ maximal (i.e. there is no other such partition $(V'_<, V'_=, V'_>)$ with $V'_= \supsetneq V_=$) is called a *capacitated Edmonds-Gallai decomposition*.

(The subscripts compare the total supply with the total demand within each part of the decomposition. For instance, within $V_<$, the supply is smaller than the demand.)

Lemma 8.3. *A capacitated Edmonds-Gallai decomposition exists and is unique, and it is the same for all maximum flows on* (D, c).

This allows us to talk about *the* capacitated Edmonds-Gallai decomposition of a network (D, c), without reference to a flow f.

This decomposition can be seen as a version of the Edmonds-Gallai decomposition for general (i.e. not necessarily bipartite, as we defined our networks to be) undirected graphs without capacities (Schrijver, 2003a, Section24.4b). A decomposition similar to the one we defined was used by Bochet et al. (2013).

The decomposition is also related to *minimum cuts* of the augmented network. Any minimum s_0–t_0 cut of the augmented network (D', c') must cut off all arcs from s_0 to $V_< \cap S$ and from $V_> \cap T$ to t_0; nodes in $V_=$ may be cut off at either the source or the sink side. (We will not use this terminology in the rest of this chapter.)

The proof of Lemma 8.3 shows how the capacitated Edmonds-Gallai decomposition of a network can be found efficiently given an arbitrary maximum flow on that network.

The capacitated Edmonds-Gallai decomposition of a network tells us something about what a proportional flow on that network must look like, as the following lemma shows.

Lemma 8.4. *If f is a proportional flow on (D, c) with supply proportions q, the capacitated Edmonds-Gallai decomposition of (D, c) is given by*

$$\begin{aligned} V_< &= \{v : q_v < 1/2\}; \\ V_= &= \{v : q_v = 1/2\}; \\ V_> &= \{v : q_v > 1/2\}. \end{aligned}$$

By applying this lemma to networks with the same graph D but modified capacities $c^{(w)}$ as in (8.10), we can show the following uniqueness result.

Theorem 8.5 (Uniqueness of q_v for proportional flows)**.** *All proportional flows on (D, c) have the same supply proportions q.*

8.3.6 Characterization in terms of lexicographic maximality

We now consider a notion that is at first glance different from the concepts we have seen: lexicographically maximum flows. Given a flow f on a network (D, c), let $u_S^{(i)}$ be the i-th smallest utilization among the nodes in S (including duplicate utilizations), and define $u_T^{(i)}$ analogously. Then a flow is called *lexicographically maximum* if it is below c and lexicographically maximizes both $(u_S^{(i)})_i$ and $(u_T^{(i)})_i$.

In other words, a lexicographically maximum flow maximizes the smallest utilization among all sources/sinks. If multiple flows below c exist achieving the same smallest utilization value, the second-smallest utilization is used as a tie-breaker, and so on. Of course, the ordering of the utilizations will be different for different flows.

Megiddo (1974) also studied lexicographically optimal maximum flows, but he looked at absolute amounts of flow at each source and sink, while we define utilizations as fractions of available capacity. He showed that the absolute source and sink utilizations may always be lexicographically maximized by the same flow, a result that also holds for our fractional utilizations.

Lemma 8.6. *In any network (D, c), a lexicographically maximum flow exists, and any such flow is also a maximum flow.*

Perhaps surprisingly, lexicographically maximum flows and proportional flows are intimately related.

Theorem 8.7 (Lex-max characterization of proportional maximum flows)**.** *A flow is lexicographically maximum if and only if it is a proportional maximum flow.*

Corollary 8.8. *A proportional flow always exists among a network's maximum flows.*

8.3.7 Proportional flows and economic fairness

Proportionality of a flow can be seen as a notion of fairness when limited resources must be divided among multiple parties. In such a setting, the bipartite network represents a constraint on the pairs of supply and demand nodes between which the resource can be transferred. Such a network may arise in practice, for example, when all suppliers offer the same resource but geographical constraints prohibit the transfer of the resource between some pairs of suppliers and demanders; or when different suppliers offer different 'types' of the resource, and each demander can use only some of these types, while each demand can be fulfilled by any combination of compatible resource types.

This is the problem studied by Moulin and Sethuraman (2013). However, the fair resource allocations derived there differ from the allocations assigned by proportional maximum flows, as Moulin and Sethuraman take a *connection-responsible* viewpoint: agents at the nodes are 'held responsible' for their connections, so that if a demand node is not connected to some supply node, it will not be compensated for the resources that might otherwise have been transferred along the missing connection. Our proportional flows correspond to a *connection-neutral* view: we try to allocate the resource as equally as possible within the limits of the flow constraints, and do not 'punish' nodes for lacking arcs if we can help it.

Two other versions of this problem are addressed by Bochet et al. (2012, 2013). While these papers also adopt our connection-neutral view, the notions of fairness considered there are *egalitarian* rather than proportional; in particular, this means that they are not based on fractions of utilized capacity, but on absolute utilizations.

Our proportional maximum flows appear closely related to (possibly being a generalization of) the *proportional rule* of İlkılıç and Kayı (2014) (published after the present section was initially written). However, the exact relationship between these two concepts remains to be investigated.

Another related problem occurs when the resource is not divisible. Economic aspects of this problem are studied for example by Roth et al. (2005), and an efficient algorithm for finding an optimal solution is given by Li et al. (2014). Because the solution is allowed to randomize over possible matchings, it appears similar to our proportional flow problem, but this similarity is only superficial.

8.3.8 Algorithms

Definition 8.3. *Proportional Maximum Flow problem*: Given a network (D, c), find a flow that is both maximum and proportional.

We first describe a straightforward algorithm for the proportional maximum flow problem; it works by finding a sequence of maximum flows.

First delete any isolated nodes; their supply proportions are given by $q_s = 1$ for sources and $q_t = 0$ for sinks. Then apply the following recursive procedure: Let $w = \sum_{s \in S} c_s / \sum_{t \in T} c_t$, and compute the modified capacities $c^{(w)}$ as in

(8.10); with this choice of w, the total demand equals the total supply. Find any maximum flow, and use it to determine the capacitated Edmonds-Gallai decomposition. (The proof of Lemma 8.3 shows how to do this efficiently.) If the decomposition has a nonempty $V_=$ part, then for nodes $v \in V_=$, we know the supply proportions q_v in the original network (D, c) using Lemma 8.4: they can be computed using $q_v/(1-q_v) = w$. For arcs adjacent to $V_=$, the maximum proportional flow f we are looking for is the componentwise rescaling of the flow in $(D, c^{(w)})$ that is maximum in (D, c). If the decomposition has only a $V_=$ part, we are done; otherwise, recursively solve the $V_<$ and the $V_>$ part. Both parts must then be nonempty for our choice of w, so this algorithm will eventually terminate.

In the worst case, this algorithm will require $O(n)$ levels of recursion, where n is the number of nodes; this will happen for instance if the graph is a path, the supplies are all equal and the demands are $1!, 2!, 3!, \ldots$. Of course, the total running time will depend on the algorithm used to solve the maximum flow problem. Many strongly polynomial maximum flow algorithms exist. Using such an algorithm as a subroutine, our overall algorithm also runs in polynomial time: in $O(n^4)$ if for instance the FIFO push-relabel algorithm of Goldberg and Tarjan (1988) is used, or in $O(n^2 m \log \frac{n^2}{m})$ with the version of the algorithm using the dynamic tree structure defined in Sleator and Tarjan (1983) (this is asymptotically more efficient if m, the number of arcs, is small).

More efficient algorithms are possible. The one we describe below is based on Gallo et al. (1989), who consider the *parametric maximum flow problem*. In this generalization of the maximum flow problem, arc capacities depend on a real-valued parameter. Their algorithms extend the maximum flow algorithms of Goldberg and Tarjan (1988) referred to above, and have the same running times: $O(n^3)$ if implemented using a FIFO rule and linked lists, or $O(nm \log \frac{n^2}{m})$ with dynamic trees. For bipartite graphs, further improvements are possible: Let n_S and n_T denote the number of sources and sinks, respectively. Then assuming without loss of generality that $n_S \leq n_T$, the dynamic tree algorithm can be modified to run in $O(n_S \cdot m \log(n_S^2/m + 2))$ time (Ahuja et al., 1994).

Gallo et al. list many applications of parametric maximum flows, which includes (in section 4.1) finding the flow that lexicographically maximizes both the source and the sink utilizations, each sorted in ascending order. By Theorem 8.7, this is equivalent to our proportional maximum flow problem. They also observe that the lexicographically maximum flow maximizes the minimum utilization and minimizes the maximum utilization, for both sources and sinks. This result carries over to proportional maximum flows.

Their algorithm follows the same idea as the straightforward algorithm described above (finding the maximum flow in a reweighted graph, then solving $V_<$ and $V_>$ recursively), with the following improvements:

- It uses the push-relabel maximum flow algorithm of Goldberg and Tarjan. With this algorithm, it is possible to reuse the results from a previous run to solve a network in which the capacities have been adjusted in a certain way. A sequence of such runs has the same asymptotic time

complexity as a single run of the algorithm.

- Two instances of the push-relabel algorithm are run concurrently: one that pushes flow from the supersource to the supersink, the other working in the reverse direction (and started from a different earlier result). When one instance finishes, the other is terminated, unless its results are needed by another run in the same sequence.

8.4 General graph games

It is time to return to the problem of finding worst-case optimal strategies for the quizmaster in a probability updating game. In Section 8.4.1, we will give a relation between proportional flows and worst-case optimal strategies for the case that $(\mathcal{X}, \mathcal{Y})$ is a bipartite graph. Section 8.4.2 generalizes to the case that $(\mathcal{X}, \mathcal{Y})$ is an arbitrary graph.

8.4.1 Bipartite graph games

Consider a probability updating game on a bipartite graph $(\mathcal{X}, \mathcal{Y})$, and marginal distribution p on the outcomes $\mathcal{X}$. We arbitrarily call one of the two colour classes of this graph S and the other T, and direct all edges from S to T: $A \coloneqq \{(s,t) \mid \{s,t\} \in \mathcal{Y}, s \in S, t \in T\}$. Define capacities c of the nodes as equal to the marginal probabilities p. Now (D, c) with $D = (\mathcal{X}, A)$ is a network as defined in Section 8.3.2. On this network, we can find a proportional maximum flow, using one the algorithms discussed in Section 8.3.8. We claim that this gives a worst-case optimal strategy to our game as follows:

Theorem 8.9. *A worst-case optimal strategy P for a probability updating game on a bipartite graph $(\mathcal{X}, \mathcal{Y})$ with marginal p can be computed from a proportional maximum flow f on the network (D, c) defined above as follows:*

$$P(x \mid y) \;=\; q'_x \;=\; \begin{cases} q_x & \text{for } x \in y \text{ and } x \in S; \\ 1 - q_x & \text{for } x \in y \text{ and } x \in T; \\ 0 & \text{for } x \notin y; \end{cases}$$

$$P(\{x_1, x_2\}) \;=\; (\alpha_C^{-1} + \beta_C^{-1}) f_{(x_1,x_2)} \quad \text{for } \{x_1, x_2\} \in \mathcal{Y},\, x_1 \in S,\, x_2 \in T,$$

where C is the f-component containing both x_1 and x_2. The RCAR vector of P is given by q'.

We see that $P(y)$ is given by another componentwise rescaling of f. The scale factor $\alpha_C^{-1} + \beta_C^{-1}$ in each f-component C can also be characterized as the one that results in $(1 - \alpha'_C)(1 - \beta'_C) = 1$ for the rescaled flow; compare this to the rescalings in Section 8.3.4, where we saw that maximum flows could be found by componentwise rescaling proportional flows to have $\max(\alpha'_C, \beta'_C) = 1$, and electrical currents could be found by the rescaling such that $\alpha'_C + \beta'_C = 1$. Note that the RCAR vector q' is the opposite of the voltage drop in the electrical current: $q'_x = 1 - \Delta U_x$.

8.4.2 Extension to general graph games

In a nonbipartite graph, we cannot assign nodes to be sources and sinks in such a way that each edge can be directed to go from a source to a sink. So we can no longer speak of a flow along the edges in the same way as before. Instead, an RCAR strategy on a general graph would correspond to a special kind of *fractional matching* (Schrijver, 2003a, Section 30.2: compare (30.4) there with (7.5) in our Section 7.5.2), similar to how RCAR strategies on bipartite graphs correspond to proportional flows. However, it turns out that worst-case optimal strategies on general graph games can be found using the proportional maximum flow algorithm on a modified flow network.

For a nonbipartite game, construct a network (D, c) with $D = (S \cup T, A)$ as follows:

$$\begin{aligned} S &= \{s_x \mid x \in \mathcal{X}\}; \\ T &= \{t_x \mid x \in \mathcal{X}\}; \\ A &= \{(s_{x_1}, t_{x_2}), (s_{x_2}, t_{x_1}) \mid \{x_1, x_2\} \in \mathcal{Y}\}, \end{aligned}$$

and capacities $c_{s_x} = c_{t_x} = p_x$. This network contains two nodes for each outcome and two arcs for each message.

Lemma 8.10. *A proportional maximum flow on the network (D, c) defined above must have $u_{s_x} = u_{t_x}$ for all $x \in \mathcal{X}$.*

So we have $q_{s_x} = 1 - q_{t_x}$ for all x. (In particular, we have that if an f-component C contains both s_x and t_x, then $q_{s_x} = q_{t_x} = 1/2$.)

Theorem 8.11. *A worst-case optimal strategy P for a probability updating game on a general graph $(\mathcal{X}, \mathcal{Y})$ with marginal p can be computed from a proportional maximum flow f on the network (D, c) defined above as follows:*

$$\begin{aligned} P(x \mid y) = q'_x &= \begin{cases} q_{s_x} = 1 - q_{t_x} & \text{for } x \in y; \\ 0 & \text{for } x \notin y; \end{cases} \\ P(\{x_1, x_2\}) &= \frac{1}{2}(\alpha_C^{-1} + \beta_C^{-1})\left(f_{(s_{x_1}, t_{x_2})} + f_{(s_{x_2}, t_{x_1})}\right) \\ &\qquad \text{for } \{x_1, x_2\} \in \mathcal{Y},\ x_1 \in S,\ x_2 \in T, \end{aligned}$$

where C is the f-component containing s_{x_1} (taking the f-component containing one of the other three nodes involved yields the same result). The RCAR vector of P is given by q'.

The following practical improvement on the algorithm is easy to make: If we first determine the capacitated Edmonds-Gallai decomposition of the network, $V_=$ will contain the sources and sinks of the outcomes that get conditional $q_v = 1/2$, while $V_<$ and $V_>$ will be bipartite and each other's mirror image, so it suffices to solve just one of those. We can reuse the maximum flow that was used to find the decomposition as a starting point for the parametric flow algorithm.

8.5 Matroid games

The other class of message structures besides graphs for which RCAR strategies are worst-case optimal strategies of the probability updating game is formed by matroids. More precisely, we assume in this section that $\mathcal{Y}$ is the set of bases of a matroid. Since matroids admit many efficient optimization algorithms (Schrijver, 2003b), one would hope that an efficient algorithm exists also for our problem of finding worst-case optimal strategies. To this end, we define the *proportional matroid basis packing problem* (named in analogy with proportional flows) as follows: Given a matroid $(\mathcal{X}, \mathcal{Y})$ with $\bigcup \mathcal{Y}$ equal to the ground set $\mathcal{X}$ and capacities $p \in \mathbf{R}^{\mathcal{X}}_{>0}$ with $\sum_{x\in\mathcal{X}} p_x = 1$, find vectors $m \in \mathbf{R}^{\mathcal{Y}}_{\geq 0}$ and $q \in \mathbf{R}^{\mathcal{X}}_{\geq 0}$ such that

$$\begin{aligned} q_x \cdot \sum_{y\ni x} m_y &= p_x \quad \text{for all } x \in \mathcal{X}; \\ \sum_{x\in y} q_x &\leq 1 \quad \text{for all } y \in \mathcal{Y}; \\ \sum_{x\in y} q_x &= 1 \quad \text{for all } y \in \mathcal{Y} \text{ with } m_y > 0. \end{aligned} \tag{8.11}$$

(From this, it follows that $\sum_y m_y = 1$.) Then let P be given by

$$P(x,y) = \begin{cases} q_x m_y & \text{for } x \in y; \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to see that P is a worst-case optimal strategy (with RCAR vector q) if and only if m and q satisfy (8.11).

Algorithm 8.2 solves this problem. Before we give the algorithm, we need to introduce some of the concepts it builds on.

The algorithm works by solving a sequence of *maximum capacitated fractional matroid basis packing* problems (Schrijver, 2003b, Corollary 42.7a): given a matroid and a vector of capacities $c \in \mathbf{R}^{\mathcal{X}}_{\geq 0}$, find weights on bases $z \in \mathbf{R}^{\mathcal{Y}}_{\geq 0}$ with

$$\sum_{y\ni x} z_y \leq c_x \text{ for all } x \in \mathcal{X} \tag{8.12}$$

such that $\sum_{y\in\mathcal{Y}} z_y$ is maximized. We say an element $x \in \mathcal{X}$ is *possibly unsaturated* if there is a maximum packing z with $\sum_{y\ni x} z_y < c_x$.

It will follow from our algorithm and its proof that if m, q is a proportional basis packing, $(\min_{x\in\mathcal{X}} q_x) \cdot m$ is a maximum fractional basis packing. Similarly, $(\max_{x\in\mathcal{X}} \cdot q_x)m$ is a *minimum capacitated fractional basis covering* (compare Schrijver, 2003b, Corollary 42.6a): a weight vector of minimal sum satisfying $\sum_{y\ni x} z_y \geq c_x$ for all $x \in \mathcal{X}$.

We need a few more definitions from matroid theory (Oxley, 2011). The *rank function* r of a matroid is given by $r(S) := \max_{y\in\mathcal{Y}} |y \cap S|$. The algorithm repeatedly splits up the matroid into smaller parts: first into the colour classes

of its induced colouring (defined in Section 7.5.1; recall from Lemma 7.8 that in matroid theory terms, these classes are exactly the 2-connected components of the matroid); and second using restrictions and contractions. The *restriction* of a matroid to a set $S \subset \mathcal{X}$ is the matroid with elements S and bases $\mathcal{Y}|S := \{y \cap S \mid y \in \mathcal{Y}, |y \cap S| = r(S)\}$. The rank function of $(S, \mathcal{Y}|S)$ is equal to that of $(\mathcal{X}, \mathcal{Y})$ for all subsets of S. The *contraction* of a matroid from a set $S \subset \mathcal{X}$ is the matroid with elements $\mathcal{X} \setminus S$ and bases $\mathcal{Y}/S := \{y \subseteq \mathcal{X} \setminus S \mid y \cup y' \in \mathcal{Y} \text{ for some } y' \in \mathcal{Y}|S\}$.

When the algorithm takes the restriction of a matroid, it takes the *conditioning* of its capacities (called that because the capacities correspond to marginal probabilities on outcomes): $p_{\cdot|S} \in \mathbf{R}^S_{\geq 0}$ is given by $p_{x|S} := p_x / \sum_{x' \in S} p_{x'}$. To combine a maximum fractional basis packing $z \in \mathbf{R}^{\mathcal{Y}}_{\geq 0}$ on a matroid $(\mathcal{X}, \mathcal{Y})$ with the solution m of a restriction $(U, \mathcal{Y}|U)$ of that matroid, we define the *contraction of z from U* as the vector $z_{\cdot/U} \in \mathbf{R}^{\mathcal{Y}/U}_{\geq 0}$ with $z_{y'/U} := \sum_{y \in \mathcal{Y}: y \setminus U = y'} z_y$. Finally, in the algorithm and its proof, we will write p_S for $\sum_{x \in S} p_x$.

Algorithm 8.2: Proportional matroid basis packing algorithm

Input: Matroid $(\mathcal{X}, \mathcal{Y})$ with $\bigcup \mathcal{Y} = \mathcal{X}$; capacities $p \in \mathbf{R}^{\mathcal{X}}_{>0}$ with $p_{\mathcal{X}} = 1$
Output: Nonnegative vectors m and q satisfying (8.11)
Determine the induced colouring of $\mathcal{Y}$;
for each colour class C **do**
  Find a maximum fractional basis packing $z \in \mathbf{R}^{\mathcal{Y}|C}_{\geq 0}$ for matroid $(C, \mathcal{Y}|C)$ with capacities $c_x := r(C) \cdot p_{x|C}$, and write $Z := \sum_{y \in \mathcal{Y}|C} z_y$;
  if $Z = 1$ **then**
    // Solution found!
    Set $q_x = p_C / r(C)$ for all $x \in C$;
    Set $m^C_y = z_y$ for all $y \in \mathcal{Y}|C$;
  else
    Determine the set of possibly unsaturated elements $U \subset C$;
    Recursively find m^U, q^U for matroid $(U, \mathcal{Y}|U)$ with capacities $p_{\cdot|U}$;
    Set $q_x = \begin{cases} q^U_x \cdot p_U & \text{for } x \in U, \\ p_{C \setminus U} / (r(C) - r(U)) & \text{for } x \in C \setminus U; \end{cases}$
    Set $m^C_y = \begin{cases} m^U_{y \cap U} \cdot z_{(y \setminus U)/U} / Z & \text{for } y \in \mathcal{Y}|C \text{ with } y \cap U \in \mathcal{Y}|U, \\ 0 & \text{for other } y \in \mathcal{Y}|C; \end{cases}$
  end
end
Set $m_y = \prod_C m^C_{y \cap C}$ for all $y \in \mathcal{Y}$.

Because a matroid on $n = |\mathcal{X}|$ elements can have an exponential number of bases, the complexity analysis of matroid algorithms usually assumes that $\mathcal{Y}$ is presented to the algorithm in the form of an *oracle*: an efficient function that

the algorithm can query to answer a particular question about the matroid. We use here an *independence testing oracle*, which answers the question whether or not a set $S \subseteq \mathcal{X}$ is an independent set of the matroid (that is, whether or not a basis $y \in \mathcal{Y}$ exists with $S \subseteq y$).

Theorem 8.12. *Algorithm 8.2 solves the proportional matroid basis packing problem in strongly polynomial time given an independence testing oracle.*

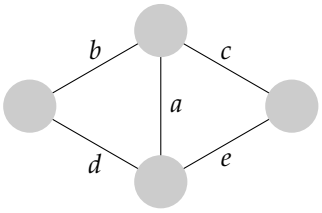
To improve the running time of the algorithm, we observe that that maximum fractional basis packing algorithm works by finding a sequence of subsets of $\mathcal{X}$ of increasing rank, until it establishes that is has found the set of possibly unsaturated elements. Within a recursive call of Algorithm 8.2, these intermediate results can be reused to find a maximum fractional basis packing more quickly.

Example 8.A. Consider the graph shown in Figure 8.6a. For this graph, we can define the cycle matroid as on page 159. This matroid has an element in its ground set for each *edge* (not node!) in the graph, and it has a basis for each *spanning tree* of the graph. So there are eight bases: all sets of three elements except for $\{a, b, d\}$ and $\{a, c, e\}$. We will run Algorithm 8.2 on this matroid with capacities p shown in Figure 8.6b. (The edges of the graph are now drawn as shaded areas, giving us room to display more information in later figures.)

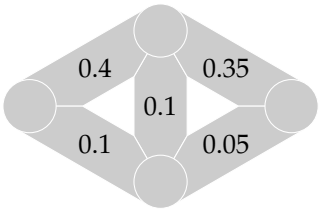
First, we determine the induced colouring of $\mathcal{Y}$, and find that all elements in the ground set $\mathcal{X}$ have the same colour. (Lemma 7.8 may be helpful for this task.) So $C = \mathcal{X}$ and the restriction to C is simply the original matroid, which has rank $r(C) = 3$. Next, we must solve a maximum fractional basis packing problem with capacities given by $c_x = 3p_x$. The unique solution is shown in Figure 8.6c; it assigns positive weight to three different bases (spanning trees), filling elements a, d and e to capacity while leaving b and c unsaturated. (In the figure, the saturated elements are shown in grey and the unsaturated elements in white. To see that no packing can do better, observe that each spanning tree contains at least one of the three saturated edges.) The value $Z = \sum_y z_y$ of this packing is only 0.75, so we are not done yet.

The algorithm is called recursively on the matroid with ground set $U = \{b, c\}$, bases $\mathcal{Y}|U = \{U\}$, and capacities $p_{b|U} = 8/15$, $p_{c|U} = 7/15$. (Note that this matroid game is trivial, containing only one message.) In the recursive call, we first determine that the the elements in U have different colours in the induced colouring of $\mathcal{Y}|U$. For each of the two colour classes $\{b\}$ and $\{c\}$, we find a maximum fractional basis packing on a matroid with one element x, one basis y, and capacity $c_x = 1$: this packing is given by $z_y = 1$. The solution returned by the recursive call is $q_b = 8/15$, $q_c = 7/15$, and $m_{\{b,c\}} = 1 \cdot 1 = 1$.

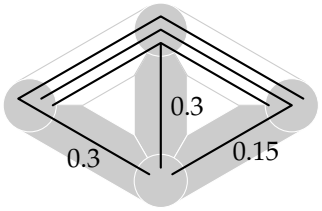
Using q^U and m^U computed by the recursive call, we can now compute the solution for the original matroid: $q_b = 8/15 \cdot 0.75 = 0.4$, $q_c = 7/15 \cdot 0.75 = 0.35$, $q_a = q_d = q_e = 0.25$; for the three bases with $y \cap U \in \mathcal{Y}|U$, $m_y = m_y^C = 1 \cdot z_{(y \setminus U)/U}/0.75 = z_y/0.75$. The proportional packing m is shown in Figure 8.6d. Note that the colours used to show regions of constant q form the induced colouring of the set of bases having positive weight.



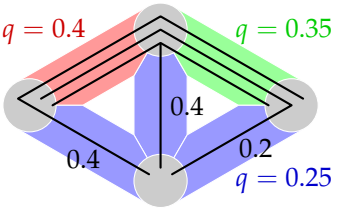
(a) Graph used to define matroid



(b) Capacities p

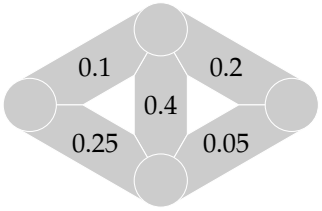


(c) Maximum fractional packing z

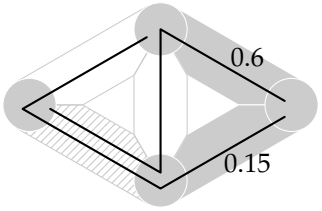


(d) Proportional packing m, q

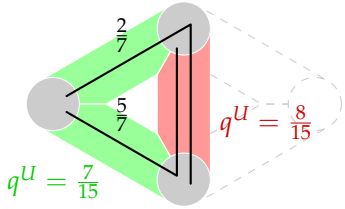
Figure 8.6: Steps in the proportional matroid basis packing algorithm for the matroid game of Example 8.A



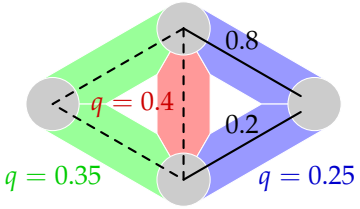
(a) Capacities p



(b) Maximum fractional packing z



(c) Proportional packing m^U, q^U of subproblem



(d) Proportional packing m, q

Figure 8.7: Steps in the algorithm for the matroid game of Example 8.B

Example 8.B. We run the algorithm again on a different input: we use the same matroid as in Example 8.A, but with the capacities given in Figure 8.7a.

The first step proceeds as before: the induced colouring of $\mathcal{Y}$ uses only one colour. The capacities $c_x = 3p_x$ in the maximum fractional packing problem are however different, and this time, we find that multiple optimal packings exist (again with $Z = 0.75$). The solution displayed in Figure 8.7b saturates elements c, d and e (shown grey or striped), but other optimal solutions exist that leave d (the striped element) unsaturated, or that saturate b. Elements c and e are always saturated, so $U = \{a, b, d\}$ (the triangle forming the left part of the graph).

The matroid $(U, \mathcal{Y}|U)$ with capacities $p_{\cdot|U}$ is solved in a recursive call; we omit the details, except to remark that the algorithm requires a further recursive call on the matroid with ground set $\{a\}$. The proportional packing returned by the first recursive call is shown in Figure 8.7c: $q^U_a = 8/15$, $q^U_b = q^U_d = 7/15$, $m^U_{\{a,b\}} = 2/7$, $m^U_{\{a,d\}} = 5/7$ (and $m^U_{\{b,d\}} = 0$).

Finally, a proportional packing for the original problem is computed. This packing has q as shown in Figure 8.7d. The vector $m = m^C$ obeys $\sum_y m_y = 1$, and so can be viewed as a probability distribution. By its construction in the algorithm, this distribution can be sampled from by sampling a basis from $\mathcal{Y}|U$ according to m^U, and extending it to a basis in the original matroid by independently sampling from the contraction $\mathcal{Y}/U = \{\{c\}, \{e\}\}$ according to $z_{\cdot/U}/Z$; here $z_{\cdot/U}$ is the contraction of z, given by $z_{\{c\}/U} = 0.6$ and $z_{\{e\}/U} = 0.15$, and $Z = 0.75$ is a normalizing constant.

This solution puts positive weight on four bases. Other solutions to this proportional packing problem exist, having the same q but different m; two such solutions put positive weight on only three bases.

8.6 Conclusion

In this chapter, we found efficient algorithms for two important classes of probability updating games, namely graph and matroid games. These algorithms can find worst-case optimal strategies exactly, whereas general-purpose algorithms for convex optimization only approximate the solution and require more running time if a greater accuracy is desired. Also, our algorithms can be performed with pencil and paper for small problem instances. This shows that worst-case optimal probability updating is a computationally feasible way of dealing with coarsened data in situations where these algorithms are applicable.

8.6.1 Future work

While the classes of graph and matroid games are important, many games remain for which we do not have algorithms with a similar level of efficiency. Two paths suggest themselves:

- Theorem 7.7 shows that if $\mathcal{Y}$ is neither a graph nor a matroid, then there exist pairs of marginals and loss functions for which the worst-case optimal strategy is not RCAR. However, this may not hold for *all* pairs of marginal and loss function, so an algorithm for finding RCAR strategies that can accept a more general class of message structures as inputs may sometimes return the worst-case optimal strategy, even if the loss function is not logarithmic.

- For games with logarithmic loss, the substitution decomposition from the previous chapter (Lemma 7.2) provides an additional tool for finding worst-case optimal strategies. For what class of message structures can an efficient algorithm be found that combines the algorithms in this chapter with the substitution decomposition?

In situations where the marginal is an estimate of some unknown distribution, it may also be useful to know how sensitive the worst-case optimal strategies are to changes in the marginal. The algorithms in this chapter may be extended to compute such information.

A different direction for future work concerns the proportional flows we used to derive an efficient algorithm for graph games. Some of the properties we found in Section 8.3 may also be of interest to economists, but more research is needed to evaluate proportional flows from that perspective. We refer to Section 8.3.7 for details.

Appendix 8.A Proofs

Proof of Theorem 8.1. Define $q \in \mathbf{R}^n$ as follows: for x even, let q_x equal the slope of the string between A_{x-1} and A_x; for x odd, let q_x equal one minus this slope. For any outcome and message $x \in y_k \in \mathcal{Y}$ such that the string does not touch A_k, we see that $P(x \mid y) = q_x$ since both are determined by the slope of the string as it passes above or below A_k. Thus P is RCAR with vector q.

For any message y_k with $P(y_k) = 0$, we need to verify that the worst-case optimal strategy satisfies $q_k + q_{k+1} \leq 1$. Note that the string does not touch A_1, because all other points are above the line through A_0 and A_1. By the same argument (replacing 'above' by 'below' if n is odd) the string does not touch A_{n-1}. If k is even, the string may be pushed down at A_k, so the slope to the left of that point, which equals q_k, must be smaller than or equal to the slope to the right, which equals $1 - q_{k+1}$. If k is odd, we similarly find $1 - q_k \geq q_{k+1}$. In both cases, we conclude $q_k + q_{k+1} \leq 1$.

What remains is to show that the marginal of P on the outcomes given in the theorem agrees with p. We do this by first deriving from p a formula for the marginal of P on the messages.

Consider two points A_a, A_b with $a < b$ such that the string touches these points but no points in between (thus the string follows a straight line between points A_a and A_b). Using the notation p_S for $\sum_{x \in S} p_x$, the slope of this segment of the string equals

$$\frac{p_{(a,b],\text{even}}}{p_{(a,b]}}.$$

This quantity equals q_x for any even $a < x \leq b$, so we call it q_{even}, and define $q_{\text{odd}} \coloneqq p_{(a,b],\text{odd}} / p_{(a,b]} = 1 - q_{\text{even}}$.

For $a < x \leq b$, the marginal constraints $\sum_{y \ni x} P(y) P(x \mid y) = p_x$ are equivalent to $\sum_{y \ni x} P(y) = p_x / q_x$. By defining $P(y_0)$ and $P(y_n)$ as 0 (note that there are no such elements in $\mathcal{Y}$), we can write $\sum_{y \ni x} P(y) = P(y_{x-1}) + P(y_x)$. For $a < k \leq b$, we must have $P(y_k) = p_k / q_k - P(y_{k-1})$ by the marginal constraint on $x = k$. Using $P(y_a) = 0$ and applying this recursion repeatedly, we find that the following choice of marginal on messages satisfies all marginal constraints for $a < x \leq b$:

$$P(y_k) = (-1)^k \left(\frac{p_{(a,k],\text{even}}}{q_{\text{even}}} - \frac{p_{(a,k],\text{odd}}}{q_{\text{odd}}} \right) \qquad \text{for } a < k \leq b.$$

(Note that we get $P(y_b) = 0$ as required.) Meanwhile in string land, the point A_k is at height $p_{(0,k],\text{even}}$, and the string intersects the vertical line through A_k at height $p_{(0,a],\text{even}} + p_{(a,k]} q_{\text{even}}$; the (signed) difference is

$$\begin{aligned} \delta_k \coloneqq p_{(a,k],\text{even}} - p_{(a,k]} q_{\text{even}} &= p_{(a,k],\text{even}} - (p_{(a,k],\text{even}} + p_{(a,k],\text{odd}}) q_{\text{even}} \\ &= p_{(a,k],\text{even}} q_{\text{odd}} - p_{(a,k],\text{odd}} q_{\text{even}}. \end{aligned}$$

This is positive at even k where the string passes below A_k, and negative at odd k. Thus the choice of marginal we found above equals the choice given in the

theorem:

$$P(y_k) = (-1)^k \frac{\delta_k}{q_{\text{even}} q_{\text{odd}}} = \frac{|\delta_k|}{q_{\text{even}} q_{\text{odd}}}.$$

which is positive for all $a < k < b$. Because P also satisfies all marginal constraints, it follows that P is a probability distribution. □

Proof of Theorem 8.2. If f is proportional, multiply the flow in each f-component by the largest factor such that the resulting flow is under $c^{(w)}$. This gives a flow f' with the same components as f, and with supply proportions $q'_v = q_v/(w(1-q_v)+q_v)$. This is strictly increasing as a function of q_v for all $w > 0$, so f' is also proportional.

To show that f' is maximum, we need the following theory (see Schrijver, 2003a). An *augmenting path* for a flow f is a path from an unsaturated source s (i.e. with $u_s < 1$) ending at an unsaturated sink t. This path may travel backwards along arcs, but only along those that carry positive flow. By increasing the flow along the path's (odd-numbered) arcs from S to T and decreasing the flow along (even-numbered) arcs where the path travels backwards from T to S, we increase the utilization of s and t without changing the utilization of the intermediate nodes, and without rendering any flow negative. Conversely, if no augmenting path exists for a flow f, then f is a maximum flow.

An augmenting path for f' would have to start at a source s with $q'_s > 1/2$ and end at a sink t with $q'_t < 1/2$ (such s and t are unsaturated). Along the path, by the definition of proportional flow, q' can decrease only when going from a node in T to one in S, which must happen on an even-numbered arc in the path. However, such an arc where q' changes must cross to a different f'-component, so there was no flow on it. We conclude that there is no augmenting path, so f' is maximum.

If f is not proportional, we are in one of three cases:

- There are arcs $(s,t_1),(s,t_2)$ with positive flow but $u_{t_1} < u_{t_2}$. Then take $w < u_{t_2}/u_s$. Now adapt f by multiplying the flow in each f-component by the largest factor such that the resulting flow f' is under $c^{(w)}$ (this is the only componentwise rescaling that might lead to a maximum flow). To satisfy the capacity constraint on t_2, the modified flow f' must leave s and t_1 both unsaturated. So f' can be increased along (s,t_1), showing that it is not maximum.

- There are arcs $(s_1,t),(s_2,t)$ with positive flow but $u_{s_1} < u_{s_2}$. Taking $w > u_t/u_{s_2}$ and adapting f to G_w as before leaves s_1 and t unsaturated.

- The utilization of f is constant within each f-component, but there is an arc (s,t) with zero flow and $q_s > q_t$. If we take $q_s/(1-q_s) > w > q_t/(1-q_t)$, we get $q'_s > 1/2 > q'_t$ in the flow adapted to G_w: both s and t are unsaturated, so this flow is not maximum. □

Proof of Lemma 8.3. The decomposition can be constructed as follows given an arbitrary maximum flow (showing existence and uniqueness): Start with

$V_<$ consisting of all unsaturated nodes in T and $V_>$ consisting of all unsaturated nodes in S. Then add a node v to $V_<$ ($V_>$) if it is adjacent to a node u in $V_<$ ($V_>$) with either positive flow, or with $u \in T$ ($u \in S$). When no more nodes can be added to $V_<$ or $V_>$, let $V_= = V \setminus (V_< \cup V_>)$. Note that $V_< \cap V_> = \emptyset$: if it is not, the construction shows an augmenting path (as seen in the proof of Theorem 8.2). Because the above construction only adds nodes to $V_<$ or $V_>$ when they must necessarily be there, we see that a decomposition with maximal $V_=$ is obtained this way; because the order of operations does not change the result, this decomposition is unique.

We also need to show that the initial choice of maximum flow will not affect the result. Different maximum flows must be related by a (sequence of) cycles or (even-length) paths that flow could be alternately added/subtracted along. We will show that there are no such *alternating* paths or cycles that would cause $V_<$ to change; the proof for $V_>$ is analogous.

For a new node to be added to $V_<$, an alternating path or cycle would have to add flow to an arc leaving $V_<$, or cause a node in $T \setminus V_<$ to become unsaturated. Either can only be done by an alternating path or cycle with nodes both in $V_<$ and outside $V_<$. However, an alternating path with one endpoint in $V_<$ and the other outside would have to have its endpoints in T (because it needs to increase the flow along the arc leaving $V_<$, but nodes in $V_< \cap S$ are saturated), and could only increase the utilization of nodes in $T \setminus V_<$. Other border-crossing paths or cycles have both odd- and even-numbered arcs with zero flow, so they are not alternating. Hence all alternating paths and cycles that have one node in $V_<$ are completely in $V_<$.

We still need to show that alternating paths and cycles contained in $V_<$ cannot cause a node to disappear from $V_<$. After applying an alternating path within $V_<$, at least one of its endpoints (in T) will be unsaturated, and following the path from there, we find that each of its nodes must still be in $V_<$. Applying an alternating cycle within $V_<$ does not change the set of nodes that are added to $V_<$ due to not being saturated or due to (sequences of) arcs not in the cycle: this set must have been nonempty, and must still be so. Again, by following the cycle from such a node, we find that all its nodes must still be in $V_<$. □

Proof of Lemma 8.4. Let f' be the maximum componentwise rescaling of f (which has the same components and supply proportions as f). An f'-component C has $q_C < 1/2$ if and only if its nodes in T are unsaturated by f', $q_C > 1/2$ if and only if its nodes in S are unsaturated by f', and $q_C = 1/2$ if and only if all its nodes are saturated by f'. By the inequality in the definition of proportional flows, D will have no arcs disallowed by the decomposition. □

Proof of Theorem 8.5. Consider a proportional maximum flow f on (D, c). An isolated node v of D has $q_v = 1$ if it is a source and $q_v = 0$ if it is a sink, regardless of f. For each nonisolated $v \in V$, let $w = q_v/(1 - q_v)$ according to f, and determine $c^{(w)}$ (the modified capacities as in (8.10), with demands multiplied by w), $f^{(w)}$ (the maximum componentwise rescaling of f), $q^{(w)}$ (the supply proportions of $f^{(w)}$), and $V_=^{(w)}$ (the fully saturated part of the capacitated

Edmonds-Gallai decomposition of $(D, c^{(w)})$). Then for node v, $q_v^{(w)} = 1/2$, so $v \in V_=^{(w)}$. Another proportional flow f' on (D, c) has to be a componentwise rescaling of a proportional flow on $(D, c^{(w)})$, but by Lemma 8.4, all proportional flows on $(D, c^{(w)})$ have $q_v^{(w)} = 1/2$. Thus like f, f' has $q'_v/(1 - q'_v) = w$. This determines q'_v uniquely for any proportional flow f' on (D, c). □

Proof of Lemma 8.6. A flow that is not maximum is also not lexicographically maximum: by increasing the flow along an augmenting path, two utilizations (one source, one sink) are increased, so the resulting flow is also lexicographically larger than the original. Thus a lexicographically maximum flow is also a maximum flow.

Let $(V_<, V_=, V_>)$ be the capacitated Edmonds-Gallai decomposition of V. All nodes in $V_=$ have $u_v = 1$ for any maximum flow. Within $V_<$, all sources have $u_s = 1$ and within $V_>$, all sinks have $u_t = 1$. Hence $u_S^{(i)}$ depends only on the flow in $V_>$ and $u_T^{(i)}$ only on the flow in $V_<$, so that both can be optimized independently. Thus a lexicographically maximum flow always exists. □

Proof of Theorem 8.7. Let $(V_<, V_=, V_>)$ be the capacitated Edmonds-Gallai decomposition of V. For lexicographically maximum f, consider nodes $s \in S \cap V_<$ and $t, t' \in T \cap V_<$ with $u_t < u_{t'}$ and $(s, t), (s, t') \in A$. If $f_{(s,t')} > 0$, there is an alternating path that increases u_t without decreasing the utilization of any sink with utilization smaller that u_t; this contradicts the lexicographic maximality, so $f_{(s,t')} = 0$. So each source in $V_<$ is connected by f only to sinks whose utilization is the same (and is connected to at least one sink, because all sources participate in the flow). Thus $V_<$ decomposes into f-components with constant q_C within each f-component, and any arcs with zero flow between the components of the capacitated Edmonds-Gallai decomposition go from $s \in S \cap V_<$ to $t \in T \cap V_<$ with $q_s < q_t < 1/2$.

Analogously, $V_>$ decomposes into f-components with constant $q_C > 1/2$, and all f-components of $V_=$ have $q_C = 1/2$. By its definition, there are no arcs across the components of the capacitated Edmonds-Gallai decomposition with $q_s > q_t$. So the decomposition of V into f-components obtained this way satisfies all properties of a proportional flow. With Lemma 8.6, this shows that any lexicographically maximum flow is a proportional maximum flow.

For the converse: Lemma 8.6 tells us that a lexicographically maximum flow exists, and that it is a maximum flow. Other maximum flows with the same utilizations on each node are also lexicographically maximum. As we just saw, these flows are proportional. There are no other proportional maximum flows, because all proportional maximum flows have these utilizations by Theorem 8.5. Thus all proportional maximum flows are lexicographically maximum. □

Proof of Theorem 8.9. It is immediate that P is a distribution over pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ with $x \in y$, and that it satisfies the equality on q'_x imposed by the

RCAR condition. For the inequality we have, for each arc $(s,t) \in A$,

$$q_s \leq q_t \iff q'_s \leq 1 - q'_t \iff q'_s + q'_t \leq 1.$$

Finally we show that P agrees with the marginal p for a source $x \in S$ (the case for sinks is analogous):

$$\begin{aligned}\sum_{y \ni x} P(x \mid y)P(y) &= \frac{\beta_C}{\alpha_C + \beta_C} \sum_{y \ni x} P(y) = \frac{\beta_C}{\alpha_C + \beta_C} \frac{\alpha_C + \beta_C}{\alpha_C \beta_C} f_x \\ &= \frac{1}{\alpha_C} \alpha_C \cdot c_x = p_x\end{aligned}$$

using $\alpha_C^{-1} + \beta_C^{-1} = (\alpha_C + \beta_C)/(\alpha_C \beta_C)$. Thus P is a strategy for the quizmaster, and is RCAR with vector q'. It follows from Theorem 7.5 that P is worst-case optimal. □

Proof of Lemma 8.10. First note that the network is *skew symmetric*: we get the same network when we rename each s_x to t_x and vice versa and reverse the direction of all arcs.

Now suppose the claim of the lemma is false and there exists a proportional maximum flow with $u_{s_x} \neq u_{t_x}$ for some outcome x. By the skew symmetry of the network, this gives another proportional maximum flow on the original network which has utilizations $u'_{s_x} = u_{t_x} \neq u_{s_x}$. If two maximum flows on the same network have different utilizations, this implies that they have different supply proportions (by the invertibility demonstrated in (8.9)). But this contradicts the uniqueness result of Theorem 8.5. □

Proof of Theorem 8.11. We show that this satisfies the marginal constraint. We take C to be the f-component containing s_x (a proof centred around t_x would have been analogous):

$$\begin{aligned}\sum_{y \ni x} P(x \mid y)P(y) &= \frac{\beta_C}{\alpha_C + \beta_C} \sum_{y \ni x} P(y) = \frac{\beta_C}{\alpha_C + \beta_C} \frac{\alpha_C + \beta_C}{2\alpha_C \beta_C} (f_{s_x} + f_{t_x}) \\ &= \frac{1}{2\alpha_C} 2 f_{s_x} = \frac{1}{\alpha_C} \alpha_C c_{s_x} = p_x.\end{aligned}$$

The other properties of worst-case optimal strategies in our game are verified as for Theorem 8.9. □

Proof of Theorem 8.12. We first show the correctness of a construction we will use several times to define a packing on a matroid $(\mathcal{X}, \mathcal{Y})$ from packings on its restriction and contraction. Given a matroid $(\mathcal{X}, \mathcal{Y})$ and a set $S \subseteq \mathcal{X}$, each basis $y \in \mathcal{Y}$ with $|y \cap S| = r(S)$ has representatives in the restriction and contraction: $y \cap S \in \mathcal{Y}|S$ and $y \setminus S \in \mathcal{Y}/S$. We now show that the converse also holds: for each basis $y_1 \in \mathcal{Y}|S$ and each basis $y_2 \in \mathcal{Y}/S$, we have $y_1 \cup y_2 \in \mathcal{Y}$.

To prove this claim, let $M = (\mathcal{X}, \mathcal{Y})$ and $M' = M|S \oplus M/S$ (the direct sum of the restriction to S and the contraction from S: its bases are exactly the sets $y = y_1 \cup y_2$ for $y_1 \in \mathcal{Y}|S$ and $y_2 \in \mathcal{Y}/S$). For any set $T \subseteq \mathcal{X}$,

$$\begin{aligned} r_{M'}(T) &= r_{M|S}(T \cap S) + r_{M/S}(T \setminus S) \\ &= r_M(T \cap S) + r_M(T \cup S) - r_M(S) \leq r_M(T), \end{aligned}$$

where we used the rank function of a contraction as given in Oxley (2011, Proposition 3.1.6), and the submodularity of the rank function. Because $r_M(\mathcal{X}) = r_{M'}(\mathcal{X})$, it follows that any basis of M' is also a basis of M, proving our claim.

Consider now the construction that takes vectors $z^S \in \mathbf{R}_{\geq 0}^{\mathcal{Y}|S}$ and $z \in \mathbf{R}_{\geq 0}^{\mathcal{Y}}$ to define a vector $z' \in \mathbf{R}_{\geq 0}^{\mathcal{Y}}$ as follows: $z'_y = z^S_{y \cap S} \cdot z_{(y \setminus S)/S}$ for $y \in \mathcal{Y}$ with $y \cap S \in \mathcal{Y}|S$ and $z'_y = 0$ for other $y \in \mathcal{Y}$. If z obeys $z_y = 0$ for $y \cap S \notin \mathcal{Y}|S$ (equivalently, for $|y \cap S| \neq r(S)$), z' has the following properties: For $x \in S$,

$$\sum_{y \in \mathcal{Y}, y \ni x} z'_y = \left[\sum_{y_1 \in \mathcal{Y}|S, y_1 \ni x} z^S_{y_1} \right] \cdot \left[\sum_{y_2 \in \mathcal{Y}/S} z_{y_2/S} \right] = \left[\sum_{y_1 \in \mathcal{Y}|S, y_1 \ni x} z^S_{y_1} \right] \cdot Z \tag{8.13}$$

where $Z = \sum_{y \in \mathcal{Y}} z_y$, and for $x \in \mathcal{X} \setminus S$,

$$\sum_{y \in \mathcal{Y}, y \ni x} z'_y = \left[\sum_{y_1 \in \mathcal{Y}|S} z^S_{y_1} \right] \cdot \left[\sum_{y_2 \in \mathcal{Y}/S, y_2 \ni x} z_{y_2/S} \right] = Z^S \cdot \left[\sum_{y \in \mathcal{Y}, y \ni x} z_y \right] \tag{8.14}$$

where $Z^S = \sum_{y \in \mathcal{Y}|S} z^S_y$.

Pick a colour class C of the input matroid, and consider the algorithm described in Schrijver (2003b, Theorem 42.7) (this algorithm does the real work in determining a maximum fractional basis packing). Let U be the last set found during the execution of this algorithm for which $r(U) - x(U) < r(C) - x(C)$ (where x equals the capacity vector multiplied by some constant). It follows from the operation of the algorithm that this U minimizes

$$\frac{r(C)}{\lambda} \cdot p_{C \setminus S|C} + r(S) \tag{8.15}$$

over all $S \subseteq C$ for λ equal to the optimal value Z as well as for some other $Z' > Z$. Furthermore, the optimal Z found by the algorithm satisfies

$$\frac{r(C)}{Z} \cdot p_{C \setminus U|C} + r(U) = \frac{r(C)}{Z} \cdot p_{C \setminus C|C} + r(C),$$

so $Z = p_{C \setminus U|C} \cdot r(C)/(r(C) - r(U))$.

Any z satisfying the basis packing constraint (8.12) obeys

$$r(C) \cdot \sum_y z_y = \sum_{x \in C \setminus U} \sum_{y \ni x} z_y + \sum_{x \in U} \sum_{y \ni x} z_y \leq \sum_{x \in C \setminus U} r(C) \cdot p_{x|C} + r(U) \cdot \sum_y z_y,$$

where the first sum is bounded by the capacities and the second by the fact that every basis $y \in \mathcal{Y}|C$ contains at most $r(U)$ elements from U. To achieve the optimal value $\sum_y z_y = Z$, both bounds must hold with equality, meaning that z fills all elements in $C \setminus U$ to capacity, and $z_y > 0$ only for $y \in \mathcal{Y}|C$ with $|y \cap U| = r(U)$ (and thus $|y \setminus U| = r(C) - r(U)$).

We claim that the set U considered here is the set of possibly unsaturated elements; to prove this, we still need to show that for each $x \in U$, a maximum fractional basis packing z exists that leaves x unsaturated. In fact, we will show a single z'' leaving all of U unsaturated. Because U minimizes (8.15) at $\lambda = Z'$, by Schrijver (2003b, Corollary 40.2f), the vector $(1/Z')(c)_{x\in U}$ is in the spanning set polytope of the matroid $(U, \mathcal{Y}|U)$. Then by Schrijver, Corollary 42.7a, a vector $z' \in \mathbf{R}_{\geq 0}^{\mathcal{Y}|U}$ with $\sum_{y\in\mathcal{Y}|U} z'_y = Z' > Z$ exists that is a fractional basis packing for the matroid $(U, \mathcal{Y}|U)$ with capacities $(c)_{x\in U}$. Now given this z' and an arbitrary maximum fractional basis packing z on $(C, \mathcal{Y}|C)$, let z'' be given by $z''_y = z'_{y\cap U} \cdot z_{(y\setminus U)/U} / Z'$ for $|y \cap U| = r(U)$, and $z''_y = 0$ otherwise. We have $\sum_{y\ni x} z''_y \leq c_x \cdot Z/Z' < c_x$ for $x \in U$ by (8.13) and $\sum_{y\ni x} z''_y = Z' \cdot c_x / Z' = c_x$ for $x \in C \setminus U$ by (8.14); also,

$$\sum_{y\in\mathcal{Y}|C} z''_y = \left(\sum_{y\in\mathcal{Y}|U} z'_y\right) \cdot \left(\sum_{y\in\mathcal{Y}|C/U} z_y\right) \Big/ Z' = Z,$$

so z'' is a maximum fractional basis packing that leaves all elements of U unsaturated, proving the claim.

Because U minimizes (8.15) at $\lambda = Z$, we have for any $S \subseteq C$

$$\frac{r(C)}{Z} \cdot p_{C\setminus U|C} + r(U) \;\leq\; \frac{r(C)}{Z} \cdot p_{C\setminus S|C} + r(S);$$

filling in Z, this becomes

$$r(C) \leq \frac{r(C) - r(U)}{p_{C\setminus U}} \cdot p_{C\setminus S} + r(S). \tag{8.16}$$

(It follows that $r(U) < r(C)$, otherwise $S = \emptyset$ would violate this inequality.)

Using the above, we show that q_x for $x \in C\setminus$ (q_x is the same for all such x) equals $\min_{x'\in C} q_{x'}$. We do this by induction. The base case when a fractional basis packing with $Z = 1$ exists in C: then q_x is the same for all $x \in C$. For the inductive step, let $U \neq \emptyset$ be the possibly unsaturated part of C, let C' be a colour class within U found in the recursive call (possibly $C' = U$), and let U' be its possibly unsaturated part ($U' = \emptyset$ if C' is a base case). Then for any $x \in C \setminus U$ and any $x' \in C' \setminus U'$,

$$\begin{aligned}
q_x \;&=\; \frac{p_{C\setminus U}}{r(C) - r(U)} \;=\; \frac{p_{C\setminus U}}{r(C) - r(C') - r(U \setminus C')} \\
&\leq\; \frac{p_{C\setminus U} + p_{C'\setminus U'}}{r(C) - r\left((U \setminus C') \cup U'\right)} \;=\; \frac{p_{C\setminus U} + p_{C'\setminus U'}}{r(C) - r(U \setminus C') - r(U')} \\
&\leq\; \frac{p_{C'\setminus U'}}{r(C') - r(U')} \;=\; \frac{p_{C'\setminus U'|U}}{r(C') - r(U')} \cdot p_U \;=\; q^U_{x'} \cdot p_U \;=\; q_{x'}.
\end{aligned}$$

The second and third equalities use that for a 2-connected component C' of the matroid $(U, \mathcal{Y}|U)$, $r(U) = r(C') + r(U \setminus C')$ (Oxley, 2011, Proposition 4.2.1). The first inequality is a rewriting of (8.16), using $r((U \setminus C') \cup U') \leq r(U) < r(C)$. The other inequality follows because the quantity on the middle line is a weighted average of the quantities on the first and third lines.

We are now ready to show that the output of the algorithm satisfies (8.11). We will again proceed by induction, but first show that for each colour class C, $\sum_{x \in y} q_x \leq p_C$ for all $y \in \mathcal{Y}|C$ with equality if $m^C_y > 0$, and that m^C satisfies $q_x \cdot \sum_{y \ni x} m^C_y = p_x$ for all $x \in C$.

For the base case, $\sum_{x \in y} q_x = p_C$ for all $y \in \mathcal{Y}|C$, and $Z = 1$ implies that all elements are saturated by z, so that $q_x \cdot \sum_{y \ni x} m^C_y = q_x \cdot c_x = p_C \cdot p_{x|C} = p_x$.

For the induction step, first consider $y \in \mathcal{Y}|C$ with $m^C_y > 0$. For such y, $y \cap U \in \mathcal{Y}|U$, so

$$\sum_{x \in y} q_x = \sum_{x \in y \cap U} q^U_x \cdot p_U + (r(C) - r(U)) \cdot p_{C \setminus U} / (r(C) - r(U)) = p_U + p_{C \setminus U} = p_C,$$

where we also used that $m^U_{y \cap U} > 0$, so that $\sum_{x \in y \cap U} q^U_x = 1$ by the induction hypothesis. For other $y \in \mathcal{Y}|C$, we have $\sum_{x \in y \cap U} q^U_x \leq 1$, and such y may have more than $r(C) - r(U)$ elements in $C \setminus U$. As we saw, these elements have the smallest q_x among $x \in C$, so $\sum_{x \in y} q_x \leq p_C$.

Finally, m^C satisfies the marginal constraints: For $x \in U$,

$$q_x \cdot \sum_{y \ni x} m^C_y \;=\; p_U \cdot q^U_x \cdot \sum_{y' \in \mathcal{Y}|U, y_1 \ni x} m^U_{y'} \cdot Z/Z \;=\; p_U \cdot p_{x|U} \;=\; p_x$$

using (8.13), and for $x \in C \setminus U$

$$q_x \cdot \sum_{y \ni x} m^C_y = \frac{p_{C \setminus U}}{r(C) - r(U)} \cdot 1 \cdot c_x / Z = p_{C \setminus U} \cdot \frac{p_{x|C}}{p_{C \setminus U|C}} = p_x$$

using (8.14).

Having established this for each colour class C, we conclude that $\sum_{x \in y} q_x \leq 1$ for all $y \in \mathcal{Y}$, with equality if $m^C_{y \cap C} > 0$ for all C. For all colour classes C and all $x \in C$, we have $q_x \cdot \sum_{y \ni x} m_y = q_x \cdot \sum_{y' \in \mathcal{Y}|C, y' \ni x} m^C_{y'} = p_x$ using that the colour classes are 2-connected components of $(\mathcal{X}, \mathcal{Y})$ by Lemma 7.8, and thus the matroid is the direct sum of its colour classes. This completes the proof of correctness.

To see that the algorithm terminates, observe that the sum of ranks of the matroids appearing as arguments in recursive calls is strictly smaller than the rank of the original matroid (as $r(U) < r(C)$ for each colour class C). Eventually, we must encounter the case $r(U) = 0$, meaning that all elements are saturated by z, so $Z = 1$: the base case.

It remains to prove our claim that the algorithm runs in strongly polynomial time given an independence testing oracle. Krogdahl (1977) shows how

the colour classes of the matroid can be determined in (strongly) polynomial time using an independence testing oracle. An algorithm that finds a maximum fractional packing z in strongly polynomial time using an independence testing oracle is given in Schrijver (2003b, Corollary 42.7a); the set of possibly unsaturated elements U is found by the same algorithm as we described earlier.

The independence testing oracle of $(\mathcal{X}, \mathcal{Y})$ also gives an independence testing oracle on the restrictions of $(\mathcal{X}, \mathcal{Y})$ considered in the algorithm. The remaining operations of the algorithm can also be performed in strongly polynomial time, assuming that m_y^C and m_y are not stored explicitly as sequences of numbers, but using a suitable data structure that can represent them in terms of other vectors. □

Bibliography

R. K. Ahuja, J. B. Orlin, C. Stein, and R. E. Tarjan. Improved algorithms for bipartite network flow. *SIAM Journal on Computing*, 23(5):906–933, 1994.

H. Akaike. Information theory and an extension of the maximum likelihood principle. In B. N. Petrov and F. Csáki, editors, *2nd International Symposium on Information Theory*, pages 267–281, 1973.

H. Akaike. A Bayesian extension of the minimum AIC procedure of autoregressive model fitting. *Biometrika*, 66(2):237–242, 1979.

T. Ando. Bayesian predictive information criterion for the evaluation of hierarchical Bayesian and empirical Bayes models. *Biometrika*, 94(2):443–458, 2007.

J. Y. Audibert. *PAC-Bayesian statistical learning theory*. PhD thesis, Université Paris VI, 2004.

J. Y. Audibert. Progressive mixture rules are deviation suboptimal. In *NIPS*, 2007.

R. E. Barlow, D. J. Bartholomew, J. M. Bremmer, and H. D. Brunk. *Statistical Inference under Order Restrictions: The Theory and Application of Isotonic Regression*. Wiley, New York, 1972.

A. R. Barron. Are Bayes rules consistent in information? In *Open Problems in Communication and Computation*, pages 85–91. Springer, 1987.

A. R. Barron. Information-theoretic characterization of Bayes performance and the choice of priors in parametric and nonparametric problems. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics*, volume 6, pages 27–52. Oxford University Press, Oxford, 1998.

A. R. Barron and T. M. Cover. Minimum complexity density estimation. *IEEE Transactions on Information Theory*, 37(4):1034–1054, 1991.

A. R. Barron and N. Hengartner. Information theory and superefficiency. *Annals of Statistics*, 26(5):1800–1825, 1998.

A. R. Barron, J. Rissanen, and B. Yu. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, 1998. Special Commemorative Issue: Information Theory: 1948-1998.

A. R. Barron, M. J. Schervish, and L. Wasserman. The consistency of posterior distributions in nonparametric problems. *The Annals of Statistics*, 27(2):536–561, 1999.

S. Ben-David, T. Lu, T. Luu, and D. Pál. Impossibility theorems for domain adaptation. In *Artificial Intelligence and Statistics (AISTATS)*, pages 129–136, 2010.

R. Berk. Limiting behavior of posterior distributions when the model is incorrect. *Annals of Mathematical Statistics*, 37:51–58, 1966.

J. M. Bernardo. Expected information as expected utility. *The Annals of Statistics*, 7(3):686–690, 1979.

J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. Wiley, Chichester, 1994.

J. Besag. Statistical analysis of non-lattice data. *The statistician*, pages 179–195, 1975.

P. Bissiri, C. Holmes, and S. Walker. A general framework for updating belief distributions. *arXiv preprint arXiv:1306.6430*, 2013.

O. Bochet, R. İlkılıç, H. Moulin, and J. Sethuraman. Balancing supply and demand under bilateral constraints. *Theoretical Economics*, 7:395–423, 2012.

O. Bochet, R. İlkılıç, and H. Moulin. Egalitarianism under earmark constraints. *Journal of Economic Theory*, 148:535–562, 2013.

G. E. P. Box. Robustness in the strategy of scientific model building. In R. L. Launer and G. N. Wilkinson, editors, *Robustness in Statistics*, New York, 1979. Academic Press.

G. E. P. Box. Sampling and Bayes' inference in scientific modelling and robustness. *Journal of the Royal Statistical Society. Series A (General)*, pages 383–430, 1980.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

L. Breiman. Statistical modeling: The two cultures (with discussion). *Statistical Science*, 16(3):199 –215, 2001.

K. P. Burnham and D. R. Anderson. *Model selection and multimodel inference: A practical information-theoretic approach*. Springer, New York, second edition, 2002.

O. Catoni. A mixture approach to universal model selection. Preprint LMENS-97-30, 1997. Available from `http://www.math.ens.fr/edition/publis/Index.97.html`.

O. Catoni. *PAC-Bayesian Supervised Classification*. Lecture Notes-Monograph Series. IMS, 2007.

O. Catoni. Discussion on the paper 'Catching up Faster by Switching Sooner' by Van Erven, Grünwald and De Rooij. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(3):399–400, 2012.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, Cambridge, UK, 2006.

G. Claeskens and N. L. Hjort. The focused information criterion. *Journal of the American Statistical Association*, 98:900–916, 2003. With discussion.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, 1991.

I. Csiszár and P. Shields. The consistency of the BIC Markov order estimator. *Annals of Statistics*, 28:1601–1619, 2000.

N. V. Cuong, W. S. Lee, N. Ye, K. M. A. Chai, and H. L. Chieu. Active learning for probabilistic hypotheses using the maximum Gibbs error criterion. In *Advances in Neural Information Processing Systems 26*, 2013.

A. S. Dalalyan and A. B. Tsybakov. Mirror averaging with sparsity priors. *Bernoulli*, 18(3):914–944, 2012.

P. L. Davies and A. Kovac. Local extremes, runs, strings and multiresolution. *The Annals of Statistics*, 29:1–65, 2001.

A. P. Dawid. The well-calibrated Bayesian. *Journal of the American Statistical Association*, 77:605–611, 1982. Discussion: pages 611–613.

A. P. Dawid. Present position and potential developments: Some personal views, statistical theory, the prequential approach. *Journal of the Royal Statistical Society, Series A*, 147(2):278–292, 1984.

A. P. Dawid. Probability, causality and the empirical world: A Bayes–de Finetti–Popper–Borel synthesis. *Statistical Science*, 19:44–57, 2004.

P. De Blasi and S. G. Walker. Bayesian asymptotics with misspecified models. *Statistica Sinica*, 23:169–187, 2013.

J. L. Doob. Application of the theory of martingales. In *Le Calcul de Probabilités et ses Applications. Colloques Internationaux du Centre National de la Recherche Scientifique*, pages 23–27, Paris, 1949.

A. Doucet and N. Shephard. Robust inference on parameters via particle filters and sandwich covariance matrices. Technical Report 606, University of Oxford, Department of Economics, 2012.

L. Dümbgen, R. Samworth, and D. Schuhmacher. Approximation by log-concave distributions, with applications to regression. *The Annals of Statistics*, 39(2):702–730, 2011.

D. B. Dunson and J. A. Taylor. Approximate Bayesian inference for quantiles. *Nonparametric Statistics*, 17(3):385–400, 2005.

T. van Erven and P. Harremoës. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

T. van Erven, P. D. Grünwald, and S. de Rooij. Catching up faster in Bayesian model selection and model averaging. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

T. van Erven, P. D. Grünwald, W. M. Koolen, and S. de Rooij. Adaptive hedge. In *Advances in Neural Information Processing Systems 24 (NIPS-11)*, 2011.

T. van Erven, P. D. Grünwald, and S. de Rooij. Catching up faster by switching sooner: A predictive approach to adaptive estimation with an application to the AIC-BIC dilemma. *Journal of the Royal Statistical Society, Series B*, 74(3): 361–417, 2012. With discussion.

T. E. Feenstra. Conditional prediction without a coarsening at random condition. Master's thesis, Leiden University, 2012. Thesis adviser: P. D. Grünwald.

T. S. Ferguson. *Mathematical Statistics: A Decision Theoretic Approach*. Academic Press, New York, 1967.

Y. Freund, Y. Mansour, and R. E. Schapire. Generalization bounds for averaged classifiers (how to be a Bayesian without believing). *Annals of Statistics*, 32 (4):1698–1722, 2004.

G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18(1):30–55, 1989.

A. E. Gelfand and S. K. Ghosh. Model choice: A minimum posterior predictive loss approach. *Biometrika*, 85(1):1–11, 1998.

A. Gelman. Bayes and Popper. Entry in A. Gelman's blog on Statistical Modeling, Causal Inference, and Social Science, 2004.

A. Gelman and C. Shalizi. Philosophy and the practice of Bayesian statistics. *British Journal of Mathematical and Statistical Psychology*, 2012.

A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. CRC Press, Boca Raton, FL, third edition, 2013.

A. Gelman, J. Hwang, and A. Vehtari. Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, 24:997–1016, 2014.

S. Ghosal, J. Ghosh, and A. van der Vaart. Convergence rates of posterior distributions. *Annals of Statistics*, 28(2):500–531, 2000.

R. D. Gill. The Monty Hall problem is not a probability puzzle (It's a challenge in mathematical modelling). *Statistica Neerlandica*, 65:58–71, 2011.

R. D. Gill and P. D. Grünwald. An algorithmic and a geometric characterization of coarsening at random. *The Annals of Statistics*, 36:2409–2422, 2008.

R. D. Gill, M. J. van der Laan, and J. M. Robins. Coarsening at random: Characterizations, conjectures, counter-examples. In D. Y. Lin and T. R. Fleming, editors, *Proc. First Seattle Symposium in Biostatistics: Survival Analysis*, pages 255–294, 1997.

A. V. Gnedin. The Monty Hall problem in the game theory class. *arXiv preprint arXiv:1107.0326*, 2011.

T. Gneiting and A. E. Raftery. Strictly proper scoring rules, predictions, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

A. V. Goldberg and R. E. Tarjan. A new appoach to the maximum-flow problem. *Journal of the Association for Computing Machinery*, 35(4):921–940, 1988.

G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215–223, 1979.

I. J. Good. *Good thinking: The foundations of probability and its applications*. University of Minnesota Press, 1983.

P. D. Grünwald. *The Minimum Description Length Principle and Reasoning under Uncertainty*. PhD thesis, University of Amsterdam, The Netherlands, 1998. Available as ILLC Dissertation Series 1998-03; see `www.grunwald.nl`.

P. D. Grünwald. Viewing all models as "probabilistic". In *Proceedings of the Twelfth ACM Conference on Computational Learning Theory (COLT' 99)*, pages 171–182, 1999.

P. D. Grünwald. *The Minimum Description Length Principle*. MIT Press, Cambridge, MA, 2007.

P. D. Grünwald. That simple device already used by Gauss. In P. D. Grünwald, P. Myllymäki, I. Tabus, M. Weinberger, and B. Yu, editors, *Festschrift in Honor of Jorma Rissanen on the Occasion of his 75th Birthday*, pages 293–304. Tampere University Press, Tampere, Finland, 2008.

P. D. Grünwald. Safe learning: Bridging the gap between Bayes, MDL and statistical learning theory via empirical convexity. In *Proceedings of the Twenty-Fourth Conference on Learning Theory (COLT' 11)*, 2011.

P. D. Grünwald. The safe Bayesian: Learning the learning rate via the mixability gap. In *Proceedings 23rd International Conference on Algorithmic Learning Theory (ALT '12)*. Springer, 2012.

P. D. Grünwald. Safe Bayesian learning theory. Manuscript in Preparation, 2014.

P. D. Grünwald and A. P. Dawid. Game theory, maximum entropy, minimum discrepancy and robust Bayesian decision theory. *The Annals of Statistics*, 32 (4):1367–1433, 2004.

P. D. Grünwald and J. Y. Halpern. Updating probabilities. *Journal of Artificial Intelligence Research*, 19:243–278, 2003.

P. D. Grünwald and J. Y. Halpern. When ignorance is bliss. In *Proceedings of the Twentieth Annual Conference on Uncertainty in Artificial Intelligence (UAI 2004)*, Banff, Canada, 2004.

P. D. Grünwald and J. Langford. Suboptimality of MDL and Bayes in classification under misspecification. In *Proceedings of the Seventeenth Conference on Learning Theory (COLT' 04)*, New York, 2004. Springer-Verlag.

P. D. Grünwald and J. Langford. Suboptimal behavior of Bayes and MDL in classification under misspecification. *Machine Learning*, 66(2-3):119–149, 2007. DOI 10.1007/s10994-007-0716-7.

J. Gu and S. Ghosal. Bayesian ROC curve estimation under binormality using a rank likelihood. *Journal of Statistical Planning and Inference*, 139(6):2076–2083, 2009.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer Verlag, 2001.

D. F. Heitjan and D. B. Rubin. Ignorability and coarse data. *The Annals of Statistics*, 19:2244–2253, 1991.

D. P. Helmbold and M. K. Warmuth. Some weak learning results. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 399–412. ACM, 1992.

U. Hjorth. Model selection and forward validation. *Scandinavian Journal of Statistics*, 9:95–105, 1982.

P. Hoff and J. Wakefield. Bayesian sandwich posteriors for pseudo-true parameters. *arXiv preprint arXiv:1211.0087*, 2012.

C. M. Hurvich and C.-L. Tsai. Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307, 1989.

R. İlkılıç and Ç. Kayı. Allocation rules on networks. *Social Choice and Welfare*, 43:877–892, 2014.

M. Jaeger. Ignorability for categorical data. *The Annals of Statistics*, 33:1964–1981, 2005a.

M. Jaeger. Ignorability in statistical and probabilistic inference. *Journal of Artificial Intelligence Research*, 24:889–917, 2005b.

L. Jansen. Robust Bayesian inference under model misspecification. Master's thesis, Leiden University, 2013.

H. Jeffreys. *Theory of Probability*. Oxford University Press, London, 3rd edition, 1961.

A. Juditsky, P. Rigollet, and A. B. Tsybakov. Learning by mirror averaging. *The Annals of Statistics*, 36(5):2183–2206, 2008.

S. Jukna. *Extremal Combinatorics: With Applications in Computer Science*. Springer, Berlin, 2001.

B. Kleijn and A. van der Vaart. Misspecification in infinite-dimensional Bayesian statistics. *Annals of Statistics*, 34(2), 2006.

S. Konishi and G. Kitagawa. Generalised information criteria in model selection. *Biometrika*, 83(4):875–890, 1996.

W. Kotłowski, P. D. Grünwald, and S. de Rooij. Following the flattened leader. In *Conference on Learning Theory (COLT)*, pages 106–118, 2010.

S. Krogdahl. The dependence graph for bases in matroids. *Discrete Mathematics*, 19:47–59, 1977.

S. Lacoste-Julien, F. Huszár, and Z. Ghahramani. Approximate inference for the loss-calibrated Bayesian. *AISTATS 2011 - Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 15:416–424, 2011.

H. Leeb. Evaluation and selection of models for out-of-sample prediction when the sample size is small relative to the complexity of the data-generating process. *Bernoulli*, 14(3):661–690, 2008.

F. B. Lempers. *Posterior Probabilities of Alternative Linear Models*. University Press, Rotterdam, 1971.

J. Li, Y. Liu, L. Huang, and P. Tang. Egalitarian pairwise kidney exchange: Fast algorithms via linear programming and parametric flow. In A. Lomuscio, P. Scerri, A. Bazzan, and M. Huhns, editors, *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems*, pages 445–452, 2014.

J. Q. Li. *Estimation of Mixture Models*. PhD thesis, Yale University, New Haven, CT, 1999.

F. Liang, R. Paulo, G. Molina, M. A. Clyde, and J. O. Berger. Mixtures of *g* priors for Bayesian variable selection. *Journal of the American Statistical Association*, 103(481), 2008.

E. Mammen and S. van de Geer. Locally adaptive regression splines. *The Annals of Statistics*, 25:387–413, 1997.

D. McAllester. PAC-Bayesian stochastic model selection. *Machine Learning*, 51 (1):5–21, 2003.

N. Megiddo. Optimal flows in networks with multiple sources and sinks. *Mathematical Programming*, 7:97–107, 1974.

C. W. Misner, K. S. Thorne, and J. A. Wheeler. *Gravitation*. W. H. Freeman, San Fransisco, 1973.

R. H. Möhring and F. J. Radermacher. Substitution decomposition for discrete structures and connections with combinatorial optimization. *Annals of Discrete Mathematics*, 19:257–356, 1984.

H. Moulin and J. Sethuraman. The bipartite rationing problem. *Operations Research*, 61:1087–1100, 2013.

U. K. Müller. Risk of Bayesian inference in misspecified models, and the sandwich covariance matrix. *Econometrica*, 81(5):1805–1849, 2013.

N. Murata, S. Yoshizawa, and S. Amari. Network Information Criterion — Determining the number of hidden units for an artificial neural network model. *IEEE Transactions on Neural Networks*, 5(6):865–872, 1994.

J. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

A. O'Hagan. Fractional Bayes factors for model comparison. *Journal of the Royal Statistical Society, Series B*, 57(1):99–138, 1995. With discussion.

H. Owhadi and C. Scovel. Brittleness of Bayesian inference and new Selberg formulas. *arXiv preprint arXiv:1304.7046*, 2013.

J. Oxley. *Matroid Theory*. Oxford University Press, New York, second edition, 2011.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

T. Park and G. Casella. The Bayesian Lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

N. Quadrianto and Z. Ghahramani. A very simple safe-Bayesian random forest. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2014. in press.

A. E. Raftery, D. Madigan, and J. A. Hoeting. Bayesian model averaging for linear regression models. *Journal of the American Statistical Association*, 92 (437):179–191, 1997.

R. V. Ramamoorthi, K. Sriram, and R. Martin. On posterior concentration in misspecified models. *arXiv preprint arXiv:1312.4620*, 2013.

J. Rissanen. Universal coding, information, prediction and estimation. *IEEE Transactions on Information Theory*, 30:629–636, 1984.

J. Robins and L. Wasserman. The foundations of statistics: A vignette. *Journal of the American Statistical Association*, 2000.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, New Jersey, 1970.

S. de Rooij, T. van Erven, P. D. Grünwald, and W. M. Koolen. Follow the leader if you can, Hedge if you must. *Journal of Machine Learning Research*, 2014.

A. E. Roth, T. Sönmez, and M. U. Ünver. Pairwise kidney exchange. *Journal of Economic Theory*, 125:151–188, 2005.

R. Royall and T.-S. Tsou. Interpreting statistical evidence by using imperfect models: Robust adjusted likelihood functions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(2):391–404, 2003.

A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume A. Springer, Berlin, 2003a.

A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume B. Springer, Berlin, 2003b.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2): 461–464, 1978.

M. Seeger. PAC-Bayesian generalization error bounds for Gaussian process classification. *Journal of Machine Learning Research*, 3:233–269, 2002.

S. Selvin. A problem in probability. *The American Statistician*, 29:67, 1975. Letter to the editor.

C. Shalizi. Dynamics of Bayesian updating with dependent data and misspecified models. *Electronic Journal of Statistics*, 3:1039–1074, 2009.

R. Shibata. Statistical aspects of model selection. In J. C. Willems, editor, *From data to model*, pages 215–240. Springer-Verlag, 1989.

D. D. Sleator and R. E. Tarjan. A data structure for dynamic trees. *Journal of Computer and System Sciences*, 26:362–391, 1983.

D. J. Spiegelhalter, N. G. Best, B. P. Carlin, and A. van der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society, Series B*, 64(4):583–639, 2002. With discussion.

J. P. Spinrad. *Efficient Graph Representations*. Number 19 in Fields Institute monographs. American Mathematical Society, Providence, RI, 2003.

V. Spokoiny. Parametric estimation. Finite sample theory. *The Annals of Statistics*, 40(6):2877–2909, 2012.

K. Sriram, R. V. Ramamoorthi, and P. Ghosh. Posterior consistency of Bayesian quantile regression based on the misspecified asymmetric Laplace density. *Bayesian Analysis*, 8(2):479–504, 2013.

M. Sugiyama and M. Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT Press, Cambridge (MA), 2012.

M. Sugiyama and H. Ogawa. Subspace information criterion for model selection. *Neural Computation*, 13(8):1863–1889, 2001.

M. Tooley. *Electronic Circuits: Fundamentals and Applications*. Routledge, London, third edition, 2006.

V. N. Vapnik. *Statistical learning theory*. Wiley, New York, 1998.

J. von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100:295–320, 1928.

M. vos Savant. Ask Marilyn. *Parade Magazine*, 15:15, 1990.

V. G. Vovk. Aggregating strategies. In *Proc. COLT' 90*, pages 371–383, 1990.

S. Walker and N. L. Hjort. On Bayesian consistency. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(4):811–821, 2002.

S. G. Walker. Bayesian inference with misspecified models. *Journal of Statistical Planning and Inference*, 143(10):1621–1633, 2013.

S. Watanabe. Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 11:3571–3591, 2010.

H. White. Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the Econometric Society*, pages 1–25, 1982.

H. Whitney. On the abstract properties of linear dependence. *American Journal of Mathematics*, 57(3):509–533, 1935.

F. Willems, Y. Shtarkov, and T. Tjalkens. The context-tree weighting method: basic properties. *IEEE Transactions on Information Theory*, 41:653–664, 1995.

P. M. Williams. Bayesian conditionalisation and the principle of minimum information. *British Journal for the Philosophy of Science*, 31(2):131–144, 1980.

H. Wong and B. Clarke. Improvement over Bayes prediction in small samples in the presence of model uncertainty. *Canadian Journal of Statistics*, 32(3): 269–283, 2004.

Y. Yang. Combining different procedures for adaptive regression. *Journal of multivariate analysis*, 74:135–161, 2000.

Y. Yang. Can the strenghts of AIC and BIC be shared? *Biometrika*, 92(4):937–950, 2005.

Y. Yang. Prediction/estimation with simple linear models: is it really that simple? *Econometric Theory*, 23:1–36, 2007a.

Y. Yang and A. R. Barron. Information-theoretic determination of minimax rates of convergence. *Annals of Statistics*, 27:1564–1599, 1999.

Z. Yang. Fair-balance paradox, star-tree paradox, and Bayesian phylogenetics. *Journal of Molecular Biology and Evolution*, 24(8):1639–1655, 2007b.

A. Zellner. On assessing prior distributions and Bayesian regression analysis with g-prior distributions. In P. K. Goel and A. Zellner, editors, *Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti*, pages 223–243. North–Holland, Amsterdam, 1986.

T. Zhang. From ϵ-entropy to KL entropy: Analysis of minimum information complexity density estimation. *Annals of Statistics*, 34(5):2180–2210, 2006a.

T. Zhang. Information theoretical upper and lower bounds for statistical estimation. *IEEE Transactions on Information Theory*, 52(4):1307–1321, 2006b.

Index

Samenvatting

Betere voorspellingen uit verkeerde en ondergespecificeerde modellen

Statistiek en machine learning behandelen de vraag wat we te weten kunnen komen over een onbekend proces, door te kijken naar de *data* (gegevens) die door het proces zijn voortgebracht. Zo wil een onderzoeker bijvoorbeeld iets te weten komen over de werking van een nieuw medicijn, op basis van gegevens over patiënten die dit medicijn eerder gebruikten. Met zulke kennis kunnen we dan *voorspellen* hoe het proces zich in de toekomst zal gedragen.

Voor de analyse van data worden vaak statistische *modellen* gebruikt. Een model is een verzameling *hypothesen*: mogelijke beschrijvingen van het onbekende proces. In veel realistische situaties is ieder beschikbaar model echter:

- *verkeerd* — geen enkele beschrijving in het model klopt precies met de werkelijkheid; of
- *ondergespecificeerd* — de beschrijvingen zijn niet volledig.

Toch willen we van data leren en goede voorspellingen doen, óók als er geen beter model voorhanden is. In dit proefschrift presenteren we methoden waarmee dit gedaan kan worden.

In hoofdstuk 2 kijken we naar *Akaike's informatiecriterium (AIC)*, in 1973 geïntroduceerd door de Japanse statisticus Hirotugu Akaike. Dit is een methode voor modelselectie, waarbij met een eenvoudige formule wordt bepaald welk van een aantal modellen naar verwachting het best in staat zal zijn om nieuwe data te voorspellen. Deze methode wordt in de praktijk vaak toegepast op een bepaald type ondergespecificeerde modellen: de datapunten bestaan uit twee delen, x en y, maar de modellen beschrijven alleen hoe y zich gedraagt als x al bekend is, en niet waar x vandaan komt. Toegepast op zulke modellen, blijkt AIC echter niet precies te doen wat we zouden willen. De formule geeft dan namelijk niet een inschatting van de voorspelfout op een níéuw datapunt, maar op een datapunt dat slechts gedeeltelijk nieuw is: een oude x met een nieuwe y. In hoofdstuk 2 laten we zien dat het ook mogelijk is om de voorspelfout voor een volledig nieuw datapunt in te schatten, namelijk met de nieuwe methode XAIC. Ook zien we dat door XAIC te gebruiken in plaats van AIC, de kwaliteit van voorspellingen in veel gevallen aanzienlijk beter wordt.

Een andere manier om een voorspeltaak aan te pakken, is *Bayesiaanse statistiek*, vernoemd naar dominee Thomas Bayes, een Engelse wiskundige die

leefde van 1702 tot 1761. In deze tak van statistiek wordt het begrip kans niet alleen gebruikt voor toevallige gebeurtenissen (zoals de kans om zes te gooien met een dobbelsteen), maar ook om onze onzekerheid uit te drukken (zoals de kans dat een bepaald model correct is). Als we onze onzekerheid over een onbekend proces in een kansverdeling hebben uitgedrukt en daarna nieuwe data uit dat proces observeren, dan volgt uit de wetten van de kansrekening een nieuwe kansverdeling, die uitdrukt hoe onze onzekerheid over het onbekende proces is veranderd nu we de data gezien hebben. Op basis van deze kansverdeling kunnen we vervolgens voorspellingen doen.

Wat gebeurt er echter als geen van de modellen correct is? In veel gevallen blijkt deze methode dan nog steeds goede voorspellingen op te leveren, hoewel ook bekend is dat het in zulke gevallen mis kan gaan. In hoofdstukken 3 tot en met 5 zien we een voorbeeld van een situatie waar de modellen allemaal verkeerd zijn (hoewel het verschil op het eerste gezicht niet problematisch lijkt), en waar Bayesiaanse methoden tot heel slechte voorspellingen leiden.

Deze problemen treden niet op als we een lagere *leersnelheid* kiezen. Een lagere leersnelheid betekent dat we onze kansverdeling minder sterk aanpassen wanneer we nieuwe data zien. Het nadeel hiervan is, dat we minder efficiënt gebruikmaken van de data. Daarom willen we de grootste leersnelheid gebruiken die nog veilig is voor onze data. De juiste snelheid is echter moeilijk te bepalen: in onze experimenten blijkt, dat veel voor de hand liggende methoden hier niet in slagen. We zien echter dat onze nieuwe methode *SafeBayes* er wel in slaagt een goede leersnelheid te kiezen, en daardoor betrouwbare voorspellingen blijft doen.

De laatste drie hoofdstukken van dit proefschrift behandelen een andere voorspeltaak, namelijk een generalisatie van het *Monty Hall-probleem* (in het Nederlands ook bekend als het *driedeurenprobleem*). In het spelprogramma *Let's make a deal*, met presentator Monty Hall, krijgt de deelnemer de keus uit drie gesloten deuren. Achter één van deze deuren staat een dure auto, achter de twee andere staan (relatief) waardeloze geiten. Nadat de deelnemer een keuze gemaakt heeft, wordt hij onderbroken door Monty, die één van de andere twee deuren opent en een geit laat zien. Verrassend genoeg kan de deelnemer zijn kans op de auto vergroten door nu van deur te veranderen!

Een algemener probleem is het volgende. Een uitkomst wordt willekeurig getrokken; we kennen de kansverdeling waarmee dit gebeurt, maar krijgen de uitkomst nog niet te zien. Vervolgens ontvangen we nieuwe informatie, waardoor sommige mogelijke uitkomsten afvallen. We weten dat deze informatie klopt, maar we weten niet door wat voor mechanisme de informatie is gekozen (zoals de deelnemer in het Monty Hall-probleem niet weet op wat voor manier de presentator kiest welke deur hij openmaakt).

De Bayesiaanse manier om kansverdelingen bij te werken, vertelt ons niet hoe we met zulke informatie om moeten gaan. Daarom pakken we dit probleem op een andere manier aan. We willen goede voorspellingen over de ware uitkomst kunnen doen, wát het mechanisme ook is dat ons de informatie gaf. We willen het dus zelfs goed doen als het mechanisme ontworpen is om het ons zo moeilijk mogelijk te maken. Met andere woorden: we kijken naar de

worst case (het slechtste geval).

Om verschillende voorspelstrategieën te vergelijken, hebben we getallen nodig die beschrijven hoe 'ver' een voorspelling ernaast zat. Zo'n toekenning van getallen aan voorspellingen heet een *verliesfunctie*. Omdat er oneindig veel verschillende manieren zijn om die getallen toe te kennen, moeten we een keuze maken.

In hoofdstuk 6 bekijken we hoe worst-case optimale strategieën voor allerlei verliesfuncties eruit zien. In het algemeen blijkt dat een voorspelstrategie die volgens de ene verliesfunctie optimaal is, dit volgens andere verliesfuncties niet hoeft te zijn. Het belangrijkste theoretische resultaat in hoofdstuk 7 is, dat er ook situaties zijn waarin de optimale strategie níét op deze manier afhankelijk is van de keuze van een verliesfunctie. Dit geldt in twee gevallen: als door iedere boodschap die het mechanisme ons zou kunnen geven, precies twee van de uitkomsten niet afvallen (de boodschappen vormen dan een *graaf*); of als de boodschappen op een specifieke manier met elkaar samenhangen (ze vormen een *matroïde*). Het Monty Hall-probleem heeft deze eigenschappen, en daarom is de optimale strategie daar niet afhankelijk van de verliesfunctie.

De resultaten van hoofdstuk 6 beschrijven hoe worst-case optimale strategieën voor allerlei verliesfuncties eruit zien, maar niet hoe je ze kunt vinden. Dit probleem wordt gedeeltelijk opgelost in de rest van hoofdstuk 7 en in hoofdstuk 8. In het bijzonder worden in dit laatste hoofdstuk efficiënte *algoritmen* voor grafen en matroïden gegeven: precieze stappenplannen die geheel automatisch uitgevoerd kunnen worden en na een eindig aantal stappen een worst-case optimale strategie hebben berekend.

Acknowledgements

This thesis could not have been written without the help and support of many people around me.

First of all, I thank my promotor Peter Grünwald. When he supervised my master's thesis, his vision and guidance first showed me how many fascinating questions in mathematics are still waiting to be answered. Supervising the PhD thesis you are reading now, he gave me great freedom to pursue the topics I came up with, even when it was not so clear if this would lead to interesting results.

Among my colleagues at the Centrum Wiskunde & Informatica, I especially thank my forerunners: Wouter Koolen, Tim van Erven and Steven de Rooij. The many thought-provoking discussions we had while solving problems together stimulated me to always think deeper. You also helped me improve my skill in the hardest part of research: writing it all down. You were examples to me and I learned a lot from you.

I would also like to thank Teddy Seidenfeld, Erik Quaeghebeur, Lex Schrijver and Guido Schäfer for their motivating discussions on different aspects of worst-case optimal probability updating.

Many other people contributed to the creation of this thesis in various ways. I would like to thank my colleagues at CWI (especially my office-mates Tom Sterkenburg and Jean-Bernard Salomond), for providing such a pleasant atmosphere; the new friends I made in Amsterdam, who quickly made me feel at home here; and my family, including my newly acquired in-laws, for their continuing support. A special thank-you goes to Gracia Murris, for making the cover of this thesis much more fun to look at.

Finally, I thank my wife Elsje for her love and the wonderful change she brought to my life. Her confidence in me has given me energy without which I could not have done this. And in addition to that, she also caught some errors in different parts of this thesis! Above all, I thank God for his blessings in my life, and for giving me inspiration whenever a piece of the puzzle was missing.

Thijs van Ommen
Amsterdam, April 2015.

Curriculum Vitae

Thijs van Ommen was born in 1984 in Rotterdam, the Netherlands. He started studying mathematics and computer science at Leiden University in 2002, obtaining a bachelor's degree in mathematics (2005), a master's degree in mathematics (2011), and a bachelor's degree in computer science (2012). The research presented in this dissertation was performed while working as a PhD student at the Centrum Wiskunde & Informatica (CWI) in Amsterdam, under the supervision of prof. dr. Peter Grünwald.