Royal Holloway University of London

Computer Learning Research Centre

Computer Science Department

Conformal Prediction and Testing under On-line Compression Models

Valentina Fedorova

Supervisors: Prof. Alex Gammerman

Prof. Vladimir Vovk

Dr. Ilia Nouretdinov

June 2014

A Thesis submitted for the degree of Doctor of Philosophy

Declaration of Authorship

I hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

Valentina Fedorova 17/06/2014

Abstract

This thesis addresses and expands research in conformal prediction – a machine learning method for generating prediction sets that are guaranteed to have a prespecified coverage probability.

Introductory Chapter 1 places conformal prediction among other machine learning methods and describes fundamentals of the theory of conformal prediction.

Chapter 2 studies efficiency criteria for conformal prediction. It outlines four established criteria of efficiency and also introduces six new criteria. The conformity measures that optimise each of these ten criteria are described when the data-generating distribution is known. Lastly, the empirical behaviour of two of the outlined criteria is illustrated for standard conformal predictors on a benchmark data set.

Chapter 3 introduces conformal prediction under hypergraphical models. These models express assumptions about relationships between data features. Several conformal predictors under these models are constructed. Their performance is studied empirically using benchmark data sets. Also label conditional conformal predictors under hypergraphical models are described and studied empirically.

Chapter 4 addresses the problem of testing assumptions for complex data, with a particular focus on the exchangeability assumption. Testing is conducted in the online mode, which gives a valid measure of the degree to which this tested assumption has been falsified after observing each data example. Such measures are provided by martingales. Two new techniques for constructing martingales – mixtures of stepped martingales and plug-in martingales – are suggested. It is also proved that, under a

stationarity assumption, plug-in martingales are competitive with commonly used power martingales. Results on testing two benchmark data sets are presented at the end of this chapter.

Chapter 5 discusses conformal prediction under the Gauss linear assumption and online testing is applied to this assumption. The performance of both prediction and testing is studied empirically for synthetic data sets and a benchmark data set.

Acknowledgements

I am very grateful to Prof. Alex Gammerman and Prof. Vladimir Vovk for their supervision over my PhD studies – thank you for teaching me conformal prediction and other interesting topics, and widening my academic interests, and also thank you very much for your care all these years.

My big appreciation goes to Dr. Ilia Nouretdinov for his inspirational ideas and patience in answering all my questions.

I am indebted to Royal Holloway and the Computer Science department for financial support during my PhD. I would like to thank all PhD students and staff in the Computer Science department for their kindness and help.

I give my special gratitude to my parents, my sister, my grandparents, and my uncle for their unlimited love, care, and support whenever I needed. I am also thankful to all my family members for their encouragement and care.

I would like to thank all my friends for their special love and care. They let me know what true friendship can achieve.

Finally, I would like to thank all my teachers from Russia who introduced me to many interesting subjects of Mathematics and Computer Science. Their passion about these subjects greatly inspired me.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

The following abbreviations are commonly used throughout this thesis.

CoP conditional probability conformity measure.

CP conformal predictor.

HOCM hypergraphical on-line compression model.

i.i.d. independent and identically distributed.

KNN k-Nearest Neighbours method.

OCM on-line compression model.

SP signed predictability conformity measure.

Chapter 1.

Introduction

This introductory chapter discusses the area of machine learning within the scope of this thesis with a particular focus on the theory of conformal prediction, and outlines the content of this thesis.

1.1. Machine learning

The general problem in the area of machine learning is to create algorithms that learn from data. Such problems can be found across the spectrum of science: from identifying digits in handwritten postal codes, through filtering out "spam" messages by using individual properties and words, to predicting which disease a patient has based on their test results.

Generally speaking, we need to predict an outcome measurement (the label) based on a set of features (the object). The pair of an object and its label is called the example. Usually, a learning algorithm is initially given a training sequence of examples and the goal is to learn how to predict the label for unseen objects. Problems where the training sequence consists of objects with labels are known as "supervised" learning problems. In "unsupervised" learning problems the training sequence contains objects without labels

and usually the goal is to discover groups within the training data. This thesis focuses on supervised learning problems.

Assumptions

It is traditional in machine learning to impose certain assumptions on data. Usually such assumptions reflect some regularity in a learning environment; it is needed to guarantee that algorithms will be able to learn and produce useful predictions. A standard assumption is that data examples are generated independently by the same (unknown) mechanism – data is independent and identically distributed (i.i.d.). Other assumptions can also represent certain knowledge about the nature of data. For example, assumptions on the structure of data represent information about relationships between data features. In that context a learning algorithm can be designed to exploit the knowledge about data.

Simple prediction and hedged prediction

Different methods have been developed to deal with supervised learning problems.

One of the first learning algorithms was proposed in 1957 by Frank Rosenblatt and called the Perceptron. The Perceptron is a simple linear classifier (it uses a linear combination of features to identify the corresponding label), and it learns by finding the weights for the linear combination that result in a good performance on a training sequence. Then one can combine a large number of the Perceptrons in a network – this is known as neural networks. In this traditional version such a network represents a composition of linear transformations and therefore produces linear solutions. Later this idea was generalised for nonlinear solutions by introducing "activation functions". The activation function is applied to units in the middle of the network and transforms the output of one unit before passing it further to next units. Then by using a nonlinear

activation function (for example, the sigmoid function $\sigma(v) = 1/(1+\exp(-v))$ the neural network can produce powerful nonlinear solutions.

Another approach to supervised learning problems is called decision trees. In this method one constructs a decision function in the form of a tree: each non-leaf node corresponds to a question about features and branches from the node correspond to answers, and each leaf node provides a decision about the label. Then for a given object one can traverse the tree from its root to a leaf node to find the corresponding label.

Yet another method – k-Nearest Neighbours (KNN) – is based on measuring distances between objects. For this purpose a metric on the space of objects is defined (for example, the Euclidean metric). Then for a given object the label is assigned based on the labels of the k nearest neighbours from a training sequence.

One more approach introduced by Vapnik and Chervonenkis – the support vector machine – is based on the idea to transform the object space into a larger space and then built a linear classifier in this transformed space.

There are also other interesting approaches to supervised learning, but a description of all these methods is not the goal of this subsection and next their common disadvantage is explained. Usually all these learning algorithms produce a point prediction (a possible value for the label), but do not provide any information on how reliable the prediction is. Some methods (such as cross-validation) allow us to evaluate the expected probability of wrong predictions, indicating the average quality of prediction, but one cannot estimate confidence for each individual prediction.

This thesis is concerned with algorithms that make "hedged predictions" [Gammerman and Vovk, 2007] – predictions associated with confidence values. Intuitively, the confidence signifies the degree of certitude that the prediction is correct. Based on the confidence values, one can obtain a prediction in the form of a set of labels whose confidence values are above a certain threshold. There are several approaches which enable

us to correlate prediction with confidence; among them are statistical learning theory, Bayesian methods and classical statistical methods. We proceed by discussing these three approaches.

Statistical learning theory

Statistical learning theory [Vapnik, 1998] (including the PAC theory [Valiant, 1984] which stands for Probably Approximately Correct learning) allows us to estimate with respect to some confidence level the upper bound on the probability of error. The theory says that for a member of a certain class of algorithms and for some confidence level, the upper bound on the probability of error can be obtained. The results from this approach are valid (i.e. the methods provide correct upper bounds on the probability of error) under the assumption that data is i.i.d. However, there are three main issues with the practical use of such methods. First, that bounds produced by statistical learning theory may depend on the VC-dimension of a family of algorithms or other numbers that are difficult to attain for methods used in practice. Second, that the bounds usually become informative when the size of the training sequence is large. The example in [Vovk et al., 2005a, p. 249] shows that the bounds calculated for the USPS data set are not useful. The authors calculate the upper bound on the probability of errors for this data set, which results in a roughly assessed value of 1.7 – not a bound for probability. Third, that the same confidence values are attached to all examples independent of their individual properties.

Bayesian methods

Bayesian methods (see, e.g., Bernardo and Smith [2000]) require the assumptions that the data-generating distribution belongs to a certain parametric family of distributions, and that the prior distribution for the parameters is known. The Bayesian theorem can

then be applied to calculate the posterior distribution for the parameters, which provides the basis for calculating the distribution of labels for a given object. The latter is used to obtain prediction sets for new objects. Hedged predictions produced by the Bayesian methods are optimal (in terms of the narrowness of their prediction sets). Nevertheless, when their prior distributions are not correct, there is no theoretical base for validity of these methods (i.e. the probability of error can be higher than it is expected). Melluish et al. [2001] show that inaccurate priors used in Bayesian methods result in invalid prediction sets. In practice, the prior is usually unknown and (arbitrary) chosen to reflect certain knowledge about the parameters. But the correctness of such a prior is the main problem in using this approach.

Classical statistical methods

A classical statistical approach for hedged prediction is constructing statistical tolerance regions (see, e.g., [Guttman, 1970]), which are also known as prediction regions or prediction sets. In this context, given a sample of independent observations from a population, one aims to predict new observations from this population. A statistical tolerance region is a range of values for new observations and can have different guarantees. One type of tolerance regions is β-content tolerance regions at a confidence level γ. Such a region contains at least 100β percent of the population with probability at least γ. Another type is β-expectation tolerance regions. Such a region is constructed to have the expected coverage probability β. The first limitation for many of these results is that they are based on the assumption of a certain parametric statistical model for data. For example, it is typical to assume that data has a normal, or Gaussian, distribution. However, statistical tolerance regions can be constructed under the assumption that data is i.i.d. (and additionally that the data-generating distribution is continuous); in this case it is just an instance of conformal predictors that will be discussed later (see

p. 257 [Vovk et al., 2005a]).

To summarise the discussion of statistical learning theory, Bayesian methods and classical statistical methods, following limitations can be pointed out: the first approach fails to provide useful confidence values in usual settings, the second one requires restrictive assumptions on the data-generating distribution, finally the last one is often based on parametric assumptions about the statistical model for data or represents an instance of more general methods for prediction. Another approach to construct methods for hedged prediction that do not have these limitations was described in Vovk et al. [2005a]. This approach is called conformal prediction and will be studied in this thesis. The next section sets the stage for this study.

1.2. Conformal Prediction

This section outlines the basis of conformal prediction. Subsection 1.2.1 introduces terminology that will be used throughout this thesis and describes the general prediction framework. Subsection 1.2.2 discusses conformal predictors (CPs) under the exchangeability assumption and their main properties. After this Subsection 1.2.3 defines on-line compression models (OCMs) and CPs under these models.

1.2.1. Terminology

We assume that Reality outputs a sequence of pairs

$$((x_1, y_1), (x_2, y_2), \ldots),$$

where each pair (x_i, y_i) consists of an *object* x_i and its *label* y_i. Each object is from a measurable space $\mathbf{X}$, called the *object space*, and each label is from a measurable space

$\mathbf{Y}$, called the *label space*. The Cartesian product

$$\mathbf{Z} := \mathbf{X} \times \mathbf{Y}$$

is called the *example space* and its elements are called *examples* (often in this thesis z_i will be also written for (x_i, y_i) for brevity).

This thesis studies prediction under certain assumptions on the process which Reality uses to generate examples. As mentioned earlier, it is traditional in machine learning to assume that examples are generated independently and from the same (but unknown) probability distribution Q on $\mathbf{Z}$, which is the assumption that data is *independent and identically distributed (i.i.d.)*. This is equivalent to saying that infinite sequences of examples $\left(z_1, z_2, \ldots\right)$ are drawn from the *power probability distribution* Q^∞ on $\mathbf{Z}^\infty$.

Exchangeability

The main assumption for conformal prediction is the exchangeability assumption, which is slightly weaker than assuming that data is i.i.d.

Consider a sequence of random variables $(Z_1, Z_2, \ldots)$ that all take values in the same example space $\mathbf{Z}$. Then the joint probability distribution $P(Z_1, \ldots, Z_N)$ of a finite number of the random variables is *exchangeable* if it is invariant under any permutation of the indices: for any measurable $E \in \mathbf{Z}^N$ and any permutation π of the set $\{1, \ldots, N\}$

$$P(E) = P\{(z_1, \ldots, z_N) : (z_{\pi(1)}, \ldots, z_{\pi(N)}) \in E\}.$$

The joint distribution of an infinite number of random variables $(Z_1, Z_2, \ldots)$ is *exchangeable* if the marginal distribution $P(Z_1, \ldots, Z_N)$ is exchangeable for every N.

Learning environments

There are two main types of learning environments – batch and on-line.

In the context of batch prediction, Predictor is initially given a *training sequence* of examples $((x_1, y_1), \ldots, (x_l, y_l))$ and Predictor's goal is to predict labels for a sequence of new objects, called the *test objects* or the *test sequence*, taking the advantage of knowing the training sequence.

In the on-line prediction setting, the training sequence is initially empty and prediction is performed as follows: for $i = 1, 2, \ldots$, (1) Predictor observes an object x_i and makes a prediction on its label based on a current training sequence; (2) then the true label y_i for the object is revealed; (3) finally, the current training sequence is augmented with the pair (x_i, y_i). On the next prediction trial the process is repeated using the updated training sequence for prediction. To summarise, in this context the prediction for a new object x_n is based on the knowledge of all previous examples $((x_1, y_1), \ldots, (x_{n-1}, y_{n-1}))$.

The strongest results of the theory of conformal prediction are stated for on-line prediction, and this thesis mainly considers on-line learning.

Now we discuss a general notion of confidence predictors and then a special type of confidence predictors – CPs – will be defined.

Confidence predictors

For a given sequence of examples $((x_1, y_1), \ldots, (x_l, y_l))$, a new object x and at each *significance level* $\epsilon \in (0, 1)$ a *deterministic confidence predictor* Γ outputs a subset

$$\Gamma^\epsilon := \Gamma^\epsilon((x_1, y_1), \ldots, (x_l, y_l), x)$$

of $\mathbf{Y}$, that is called the *prediction set*. Naturally in this context, the confidence predictor Γ *makes an error* for an object x at level ϵ if the true label y is not in Γ^ϵ.

Two properties of confidence predictors are their validity and information efficiency.

Consider data generated by an exchangeable distribution on $\mathbf{Z}^\infty$. We say that a deterministic confidence predictor is *valid* if for any significance level $\epsilon \in (0,1)$ at each prediction trial it makes errors independently and with probability ϵ. The primary goal is to develop valid confidence predictors. Information efficiency of such valid predictors can be measured differently. Broadly speaking, it is desirable that a valid prediction set contains a small number of labels: the narrower the prediction set is, the more efficient the predictor. For example, consider the problem of medical diagnosis where one predicts possible diagnoses for a patient based on their test results, in this context prediction sets containing one diagnosis are more informative than those with two or more diagnoses.

It was shown [Vovk et al., 2005a, Theorem 2.1] that no deterministic confidence predictor can be valid under the exchangeability assumption. This theorem explains that in the deterministic case it is impossible to construct the predictor that makes errors independently and with exactly probability ϵ for any significance level $\epsilon \in (0,1)$ due to finite precision. This is the reason to consider randomised confidence predictors.

A *randomised confidence predictor* at each significance level ϵ outputs a prediction set

$$\Gamma^\epsilon := \Gamma^\epsilon((x_1, y_1), \ldots, (x_n, y_n), x, \tau),$$

which additionally depends on a random number τ distributed uniformly in $[0,1]$. The validity for randomised confidence predictors is defined in the same way as for deterministic confidence predictors. Protocol 1.1 summarises randomised confidence prediction in the on-line context: Reality generates examples (x_n, y_n) and random numbers τ_n uniformly distributed in $[0,1]$, at each significance level ϵ Predictor outputs the prediction set $\Gamma^\epsilon((x_1, y_1), \ldots, (x_{n-1}, y_{n-1}), x_n, \tau_n)$.

We proceed with a discussion of special confidence predictors called CPs. Hereinafter discussions are restricted to randomised CPs (also called *smoothed* CPs in [Vovk et al., 2005a]). The adjectives "randomised" or "smoothed" will be omitted for brevity.

Protocol 1.1 On-line confidence prediction

for $n = 1, 2, \ldots$ **do**
  Reality outputs $x_n \in \mathbf{X}$ and a random number $\tau_n \sim U[0, 1]$
  Predictor outputs $\Gamma_n^\epsilon \subseteq \mathbf{Y}$ for all $\epsilon \in (0, 1)$
  Reality outputs $y_n \in \mathbf{Y}$
end for

1.2.2. Conformal prediction under the exchangeability assumption

In this subsection we discuss conformal prediction under the exchangeability assumption.

Conformity measure

The general idea of conformal prediction is to test how well a new example fits to previously observed examples. For this purpose a "conformity measure" is defined. Formally, *conformity measure* A is a measurable function that estimates how well one example fits to a bag of others assigning a *conformity score* α to the example:

$$\alpha := A\Big(\{\!|z_1, \ldots, z_l|\!\}, z\Big).$$

A *bag* of examples $\{\!|z_1, \ldots, z_l|\!\}$, in general, is a multiset (the same element may be repeated more than once) rather than a set. Usually conformity measures are based on some prediction method. Numerous machine learning algorithms have been used for designing conformity measures: see, e.g., Vovk et al. [2005a] and Balasubramanian et al. [2013]. Some examples of conformity measures will be given in Chapters 2–5. In particular, on page 68 it will be discussed that conformity measures can represent a "soft model" for data: if this model is incorrect only the efficiency of predictions may suffer, but it does not affect their validity.

Conformal predictors

Consider a training sequence of examples $(z_1, \dots, z_l)$, a new object x and a random number τ distributed uniformly in $[0, 1]$. The *conformal predictor* (CP) Γ determined by a conformity measure A is defined by

$$\Gamma^\epsilon(z_1, \dots, z_l, x, \tau) := \{y \mid p^y > \epsilon\}, \tag{1.1}$$

where $\epsilon \in (0, 1)$ is a given significance level and for each $y \in \mathbf{Y}$ the corresponding *p-value* p^y is defined by

$$p^y := \frac{\left|\left\{i = 1, \dots, l+1 \mid \alpha_i^y < \alpha_{l+1}^y\right\}\right| + \tau \left|\left\{i = 1, \dots, l+1 \mid \alpha_i^y = \alpha_{l+1}^y\right\}\right|}{l+1}, \tag{1.2}$$

and the corresponding conformity scores are defined by

$$\alpha_i^y := A\Big(\wr z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_l, (x, y)\wr, z_i\Big), i = 1, \dots, l;$$
$$\alpha_{l+1}^y := A\Big(\wr z_1, \dots, z_l\wr, (x, y)\Big).$$

Notice that the system of prediction sets (1.1) output by a CP is decreasing in ϵ, or *nested*.

Conformal transducers

The *conformal transducer* determined by a conformity measure A outputs the system of p-values $(p^y \mid y \in \mathbf{Y})$ defined by (1.2) for each training sequence $(z_1, \dots, z_l)$, an object x and a random number $\tau \sim U[0, 1]$. (This is just another representation of CPs.)

Chapter 2 will discuss several criteria of information efficiency for conformal prediction that do not depend on the significance level, these criteria are defined using the output of conformal transducers.

In [Vovk et al., 2005b, Section 2.5] the definition of transducers is slightly different: they process a sequence of examples in the on-line mode and translate each example into the p-value corresponding to the true label of the example. The definition used in this thesis is more general. Having the system of p-values, the sequence of p-values corresponding to examples can be obtained easily. To make it precise, consider a sequence of examples $(z_1, z_2, \ldots)$ and a sequence of independent random numbers $(\tau_1, \tau_2, \ldots)$ uniformly distributed on $[0, 1]$, the corresponding sequence of p-values is defined by

$$p_i := p^{y_i}, i = 1, 2, \ldots,$$

where p^{y_i} is the p-value computed by (1.2) for the training sequence $(z_1, \ldots, z_{i-1})$, the object x_i, the random number τ_i and corresponding to the label y_i. (The process will be further detailed on page 87 (Protocol 4.2).) If these p-values are independent and distributed uniformly on $[0, 1]$ we say that the conformal transducer is *valid*.

The following theorem is a standard result in the theory of conformal prediction (see, e.g., [Vovk et al., 2003, Theorem 1]).

Theorem 1. *If examples $(z_1, z_2, \ldots)$ (resp. $(z_1, z_2, \ldots, z_n)$) satisfy the exchangeability assumption, the corresponding p-values $(p_1, p_2, \ldots)$ (resp. $(p_1, p_2, \ldots, p_n)$), generated in the on-line mode using* (1.2)*, are independent and uniformly distributed on* $[0, 1]$.

In Chapter 4 we will see that this result is the midpoint in constructing martingales for testing exchangeability.

Another implication of Theorem 1 is that in the on-line mode CPs are automatically valid under the exchangeability assumption: the probability to make an error on any step is equal to the significance level and errors on different steps occur independently of each other.

1.2.3. Conformal prediction under OCMs

This subsection defines a class of statistical models – on-line compression models – and then discusses the definition of CPs under these models. (Conformal prediction under the exchangeability assumption discussed earlier is a special case of conformal prediction under OCMs, but the definitions for the exchangeability assumption is more intuitive, and that is why that case has been considered first.)

On-line compression models

The idea of OCMs is to compress data into a more or less compact summary, which is interpreted as the useful information in the data.

The following simple example demonstrates operations within OCMs. Consider the coin tossing game, let us extract the useful information for predicting head or tail from a sequence of observed outcomes. Suppose that at the first trial we observe tail. Let us summarise this event by two numbers – the number of heads and the number of tails, $(\#\text{head}, \#\text{tail}) = (0, 1)$. Then at the second trial we observe head. Our summary is updated as $(\#\text{head}, \#\text{tail}) = (1, 1)$. We proceed as follows: each trial after receiving head or tail we update the summary. Suppose that after 10 trials the following sequence of outcomes has been realised:

$$(\text{tail}, \text{head}, \text{tail}, \text{tail}, \text{tail}, \text{tail}, \text{head}, \text{head}, \text{tail}, \text{head}).$$

The summary of this sequence is $(\#\text{head}, \#\text{tail}) = (4, 6)$. This is the useful information extracted from the sequence of 10 observations. On the other hand, having the summary $(\#\text{head}, \#\text{tail}) = (4, 6)$ one may ask what is the probability of having tail at the last trial. Then, assuming that all sequences of outcomes are equally probable, this probability can be found using the summary: it is the number of sequences of length 9 with 5 tails divided

by the number of sequences of length 10 with 6 tails

$$\frac{\binom{9}{5}}{\binom{10}{6}} = \frac{6}{10}.$$

Analogously, the probability of having head at the last trial is $\frac{4}{10}$. In this simple example we can see two important operations – updating summary and looking back one step. Formally these operations are expressed by OCMs.

For the formal description of OCMs Markov kernels are used. Let Ω and Z be two measurable spaces. A function $Q(\omega, A)$, usually written as $Q(A \mid \omega)$, where ω ranges over Ω and A ranges over the measurable sets in Z, is called a *Markov kernel* if:

- as a function of A, $Q(A \mid \omega)$ is a probability distribution on Z, for each $\omega \in \Omega$;
- as a function of ω, $Q(A \mid \omega)$ is measurable, for each measurable $A \subseteq Z$.

It will be said that Q is a *Markov kernel of the type* $\Omega \hookrightarrow Z$. In other words, a Markov kernel maps $\omega \in \Omega$ to probability distributions on Z. Often in this thesis $Q(\omega)$ is used to denote the probability distribution $A \mapsto Q(A \mid \omega)$ on Z.

An *on-line compression model (OCM)* consists of five elements $(\Sigma, \square, \mathbf{Z}, F, B)$, where:

1. $\square$ is called the *empty summary*;
2. Σ is a measurable space, called the *summary space*; $\square \in \Sigma$ and elements $\sigma \in \Sigma \setminus \{\square\}$ are called *summaries*;
3. $\mathbf{Z}$ is the example space from which examples are drawn;
4. F is called the *forward function*, this is a measurable function of the type $\Sigma \times \mathbf{Z} \to \Sigma$ that describes how to update a summary after observing a new example;

5. B is called the *backward kernel*, this is a Markov kernel of the type $\Sigma \hookrightarrow \Sigma \times \mathbf{Z}$; it is also required that B be an inverse to F in the sense that

$$B(F^{-1}(\sigma) \mid \sigma) = 1$$

(in other words, it is required that $B(\cdot|\sigma)$ gives probability one to the set of pairs (σ', z) such that $F(\sigma', z) = \sigma$).

Let us discuss the intuition behind elements of this model and introduce some further notation.

Consider the on-line setting where data examples arrive one by one. An OCM is a way of summarising statistical information. At the beginning we have no information, which is represented by the empty summary $\sigma_0 := \square$. When the first example z_1 arrives, the summary is updated to $\sigma_1 := F(\sigma_0, z_1)$ and so on. In general, past examples are summarised using statistics that contain all the information useful for predicting future examples. A precise form of this summary is individual for different statistical models, and this is analogous to the notion of sufficient statistics. Let $t_n(z_1, \dots, z_n)$ be the *nth statistics* in the OCM, which maps $z_1, \dots, z_n$ to σ_n:

$$\begin{aligned} &t_1(z_1) := F(\sigma_0, z_1), \\ &t_n(z_1, \dots, z_n) := F(t_{n-1}(z_1, \dots, z_{n-1}), z_n), n = 2, 3, \dots. \end{aligned} \tag{1.3}$$

The value $\sigma_n = t_n(z_1, \dots, z_n)$ is a summary of $z_1, \dots, z_n$; it is required that the summaries should be computable sequentially: the forward function F updates σ_{n-1} to σ_n This on-line character of summarising is reflected in Condition 4 in the definition of OCM. It is also important that when summarising no useful information is lost, which is formally reflected in Condition 5. If σ_n is the summary obtained from (σ_{n-1}, z_n), then the distribution of (σ'_{n-1}, z) given σ_n is known, and therefore the more detailed descrip-

tion (σ_{n-1}, z_n) does not carry any additional information about the statistical model generating examples $z_1, z_2, \ldots$.

An OCM can be used to find the distribution of data sequences $(z_1, \ldots, z_n)$ from summary σ_n. This is done by combining the one-step backward kernels $B(\sigma_n), \ldots, B(\sigma_1)$. Intuitively, it can be understood as a sequence of drawings whose outcomes have the probability distributions specified by the kernels. The drawing from σ_n (specified by probabilities given by $B(\sigma_n)$) gives us z_n and σ_{n-1}, the drawing from σ_{n-1} (specified by probabilities given by $B(\sigma_{n-1})$) gives us z_{n-1} and σ_{n-2}, and so on. Finally this process gives us the conditional distribution P_n of $(z_1, \ldots, z_n)$ given σ_n, which can be described as

$$\begin{aligned} P_n(\,\mathrm{d}z_1, \ldots, \,\mathrm{d}z_n) :=& B(\,\mathrm{d}z_1, \square \mid \sigma_1) B(\,\mathrm{d}z_2, \sigma_1 \mid \sigma_2) \cdots \\ & B(\,\mathrm{d}z_{n-1}, \sigma_{n-2} \mid \sigma_{n-1}) B(\,\mathrm{d}z_n, \sigma_{n-1} \mid \sigma_n) \end{aligned}$$

(see [Vovk et al., 2005a], p. 191).

Consider a probability distribution Q on $\mathbf{Z}^\infty$ and an OCM $M = (\Sigma, \square, \mathbf{Z}, F, B)$. For $n = 2, 3, \ldots$ let $t_n(z_1, \ldots, z_n)$ be the nth statistic defined by (1.3). We say that Q *agrees* with M if for each $n = 2, 3, \ldots$, $B(\cdot \mid \sigma)$ is a version of the conditional distribution, w.r. to Q, of $(t_{n-1}(z_1, \ldots, z_{n-1}), z_n)$ given $t_n(z_1, \ldots, z_n) = \sigma$ and given the values $z_{n+1}, z_{n+2}, \ldots$. In other words, with probability one

$$B(A \mid \sigma) = \mathbb{P}_{(z_1, z_2 \ldots) \sim Q} \left\{ (t_{n-1}(z_1, \ldots, z_{n-1}), z_n) \in A \mid t_n(z_1, \ldots, z_n) = \sigma, z_{n+1}, z_{n+2}, \ldots \right\}$$

for each $A \subseteq \Sigma \times \mathbf{Z}$. Intuitively, this agreement has a similar sense as the "agreement" of a sufficient statistic with the statistical model for which it was constructed (in Statistics it is expressed by saying "a statistic is sufficient with respect to a statistical model").

For more information about OCMs and the intuition behind them see, e.g., [Vovk et al., 2005a, Chapter 8.1] and [Shafer and Vovk, 2008, Section 5.1]. In Chapters 3

and 5 two important classes of OCMs – hypergraphical and the Gauss linear OCMs – will be described.

Conformal predictors under OCMs

The definition of CPs can be easily extended to OCMs [Vovk et al., 2005a, Chapter 8.2].

A *conformity measure for an OCM* $(\Sigma, \square, \mathbf{Z}, F, B)$ is a measurable function

$$A : \Sigma \times \mathbf{Z} \to \mathbb{R}.$$

The function assigns a conformity score $A(\sigma, z)$ to an example z w.r. to a summary σ. Intuitively, the score reflects how typical it is to observe z having the summary σ.

Consider a training sequence $(z_1, \ldots, z_l)$, generated by a distribution that agrees with an OCM $M = (\Sigma, \square, \mathbf{Z}, F, B)$, an object x and a random number $\tau \sim U[0,1]$. Let σ_l be the summary of the training sequence. For each $y \in \mathbf{Y}$ denote $\sigma^* := F(\sigma_l, (x, y))$ (the dependence of σ^* on y is important although not reflected in this notation). For each $y \in \mathbf{Y}$ the p-value p^y is defined by

$$\begin{aligned} p^y :=& B(\{(\sigma, z) \in \Sigma \times \mathbf{Z} : A(\sigma, z) < A(\sigma_l, (x, y))\} \mid \sigma^*) \\ &+ \tau \cdot B(\{(\sigma, z) \in \Sigma \times \mathbf{Z} : A(\sigma, z) = A(\sigma_l, (x, y))\} \mid \sigma^*). \end{aligned} \tag{1.4}$$

The CP and the conformal transducer determined by a conformity measure for an OCM are defined analogously to those under the exchangeability assumption (see Subsection 1.2.2) but using (1.4) rather than (1.2) for calculating p-values.

Again, using (1.4) for an OCM we can generate the sequence of p-values under the OCM and corresponding to a sequence of examples (as explained on page 26 and substituting (1.4) instead of (1.2)). The following theorem is a generalisation of Theorem 1 (see [Vovk et al., 2005a, Theorem 8.1]).

Theorem 2. *If examples* $(z_1, z_2, \ldots)$ *(resp.* $(z_1, z_2, \ldots, z_n)$*) are generated by a probability distribution that agrees with an OCM, the corresponding p-values* $(p_1, p_2, \ldots)$ *(resp.* $(p_1, p_2, \ldots, p_n)$*), generated in the on-line mode using* (1.4) *for the OCM, are independent and uniformly distributed on* $[0, 1]$*.*

Consider $(z_1, z_2, \ldots)$ generated by a distribution that agrees with an OCM. A CP is *valid* w.r. to the OCM if for any significance level $\epsilon \in (0, 1)$ at each prediction trial it makes errors independently and with probability ϵ. Analogously, a conformal transducer is *valid* w.r. to the OCM if the corresponding p-values $(p_1, p_2, \ldots)$ are independent and uniformly distributed on $[0, 1]$. Theorem 2 implies a remarkable result of the theory of conformal prediction, that in the on-line mode CPs and conformal transducers are automatically valid under their models.

Having reviewed the preceding studies, the motivation for this work was to consider conformal prediction under different OCMs with a particular focus on their efficiency properties. The following list summarises the problems that motivates this research:

1. to find useful measures for information efficiency of conformal predictions;
2. to study efficiency of conformal prediction under stronger assumptions;
3. to develop methods for testing assumptions used for prediction.

1.3. Main contributions

The following list summarises the original results that were obtained during the course of work on this thesis. The corresponding chapters for these findings are given following each bullet point.

- (Chapter 2) The development of new efficiency criteria for CPs, including the investigation of empirical counterparts of two optimal conformity measures for the most interesting criteria.

- (Chapter 3) Several new CPs under hypergraphical models are described; these CPs are studied empirically, demonstrating that hypergraphical models are useful for conformal prediction; for the unconditional case, the best (or almost the best) performance among these CPs is achieved when hypergraphical models only used as "soft models".

- (Chapter 4) Two new techniques for constructing martingales are suggested; plug-in martingales are constructed by looking at data and it is proved that, under a stationarity assumption, these plug-in martingales are competitive with previously introduced martingales.

- (Chapter 5) The extension of on-line testing to the Gauss linear assumption.

1.4. Publications

The following is a list of papers published during work on this thesis:

- A paper [Fedorova et al., 2012b] describing plug-in martingales for testing exchangeability.

- A paper [Fedorova et al., 2012a] applying the plug-in martingales for testing the Gaussian linear assumption and also empirically studying conformal prediction under this assumption.

- A working paper [Fedorova et al., 2013a] describing CPs under hypergraphical models and presenting an empirical study of the CPs; the papers [Fedorova et al., 2013b] and [Fedorova et al., 2013c] are shorter versions of this paper.

- A working paper [Vovk et al., 2014] describing efficiency criteria for conformal prediction and including an empirical study of their properties.

1.5. Thesis summary

The remainder of this thesis is organised as follows:

The main part of this thesis starts (in Chapter 2) with a study of efficiency criteria for conformal prediction. Some of these criteria are used to study the performance of different CPs in Chapter 3. Chapter 3 defines hypergraphical models and CPs under these models, and presents an empirical study of the CPs. After this, the problem of testing assumptions is addressed. Chapter 4 describes the framework for on-line testing, defines the exchangeability martingales and describes the construction of the martingales. Then (in Chapter 5) the testing method is applied to the Gauss linear assumption. Also Chapter 5 presents an empirically study of conformal prediction under this assumption. Chapter 6 concludes and also outlines some future directions for this field of research.

There are also two separate appendices: Appendix A describes used data sets and Appendix B presents an R implementation of conformal prediction and martingales for on-line testing.

Chapter 2.

Efficiency criteria for conformal prediction

As mentioned for confidence predictors in general, two main desiderata for conformal predictors (CPs) are their validity and efficiency. Since (as explained in Section 1.2) CPs are automatically valid under their models, the most important property for CPs is their efficiency. The efficiency reflects the narrowness, in some sense, of prediction sets producer by CPs; but there is no precise definition for this notion. This chapter studies different criteria of efficiency for conformal prediction. It considers the classification problem only. Section 2.1 outlines four established criteria of efficiency and Section 2.2 introduces six new criteria. Section 2.3 discusses the conformity measures that optimise each of these ten criteria in an idealised setting when the data-generating distribution is known. Section 2.4 moves to a more realistic setting and studies empirical behaviour of two important criteria. Section 2.5 concludes.

2.1. Introduction

As discussed in the previous chapter CPs have guaranteed validity. However, it is easy to construct a valid but not useful CP: for each significance level ϵ the predictor outputs the whole label set with probability $1 - \epsilon$ and the empty set with probability ϵ. Other non-trivial CPs may differ in their efficiency (which evaluates the narrowness, in some sense, of their prediction sets). Empirical investigation of the efficiency of various CPs is becoming a popular area of research: see, e.g., Vovk et al. [2005a], Gammerman [2012], Balasubramanian et al. [2013], and Papadopoulos et al. [2014]. This chapter points out that the standard criteria of efficiency used in literature have a serious disadvantage; such criteria of efficiency will be called "improper". In two recent papers two proper criteria have been introduced, and this chapter introduces two more and argues that proper criteria should be used in place of more standard ones. This chapter is concentrated on the case of classification only (the label space is finite).

Surprisingly few criteria of efficiency have been used in literature, and even fewer have been studied theoretically. One can speak of the efficiency of individual predictions or of the overall efficiency of predictions on a test sequence; the latter is usually (in particular, in this chapter) defined by averaging the efficiency over the individual test examples, and so this introductory section only discusses the former.

The two criteria for efficiency of a prediction that have been used most often in literature (see, e.g., the references given above) are:

- The confidence and credibility of the prediction. This criterion does not depend on the choice of a significance level ϵ.
- Whether the prediction is a singleton (the ideal case), multiple (an inefficient prediction), or empty (a superefficient prediction) at a given significance level ϵ.

The other two criteria that have been used are the sum of the p-values for all potential

labels (this does not depend on the significance level) and the size of the prediction set at a given significance level: see the recent papers by Fedorova et al. [2013a] and Johansson et al. [2013].

As before, this chapter only considers the case of smoothed CPs: the case of deterministic predictors may lead to packing problems without an explicit solution. For example, this is the case for the N criterion defined below: for this criterion the goal is to form a valid prediction set (i.e., with coverage probability $1-\epsilon$, where ϵ is the significance level) containing the minimal number of labels, but in the deterministic case it may happen that the optimal prediction set for level ϵ_1 includes labels that are not in the optimal prediction set for level $\epsilon_2 < \epsilon_1$, so this problem would require a separate solution for each ϵ. This situation is analogous to the Neyman–Pearson lemma: cf. Lehmann [1986], Section 3.2.

2.2. Criteria of efficiency for conformal predictors and transducers

This chapter uses the notation introduced in Section 1.2 and the label space $\mathbf{Y}$ is finite. Also the main assumption in this chapter is the exchangeability assumption, but all discussions will be useful for conformal prediction under other assumptions as well. The definitions of CPs and conformal transducers, and their validity were discussed in Subsection 1.2.2 (see page 24). Next we discuss measures of information efficiency for CPs and conformal transducers. The efficiency of CPs means that the prediction sets they output tend to be small, and the efficiency of conformal transducers means that their p-values tend to be small.

Suppose we are given a test sequence $(z_{l+1}, \dots, z_{l+k})$ and would like to use it to measure the efficiency of the predictions derived from the training sequence $(z_1, \dots, z_l)$. For each

test example $z_i = (x_i, y_i)$, $i = l+1, \ldots, l+k$, we have a nested family $(\Gamma_i^\epsilon \mid \epsilon \in (0,1))$ of subsets of $\mathbf{Y}$ and a system of p-values $(p_i^y \mid y \in \mathbf{Y})$. This chapter will discuss ten criteria of efficiency for such a family or a system, but some of them will depend, additionally, on the observed labels y_i of the test examples. Let us start from the *prior* criteria, which do not depend on the observed test labels.

Basic prior criteria

We will have two kinds of criteria: those applicable to the prediction sets Γ_i^ϵ and so depending on the significance level ϵ and those applicable to systems of p-values $(p_i^y \mid y \in \mathbf{Y})$ and so independent of ϵ. The simplest criteria of efficiency are:

- The *S criterion* (with "S" standing for "sum") measures efficiency by the average sum $\frac{1}{k}\sum_{i=l+1}^{l+k}\sum_{y\in\mathbf{Y}} p_i^y$ of the p-values. It is ϵ-free.
- The *N criterion* uses the average size $\frac{1}{k}\sum_{i=l+1}^{l+k}|\Gamma_i^\epsilon|$ of the prediction sets ("N" stands for "number": the size of a prediction set is the number of labels in it). Under this criterion the efficiency is a function of the significance level ϵ.

For both these criteria small values are preferable. The S criterion was introduced in Fedorova et al. [2013a] and the N criterion was introduced independently in Johansson et al. [2013] and Fedorova et al. [2013a], although the analogue of the N criterion for regression (where the size of a prediction set is defined to be its Lebesgue measure) had been used earlier in Lei and Wasserman [2014].

Other prior criteria

A disadvantage of the basic criteria is that they look too stringent. Even for a good conformal transducer, we cannot expect all p-values p^y to be small: the p-value corresponding to the true label will not be small with high probability; and even for a good

CP we cannot expect the size of its prediction set to be zero: with high probability it will contain the true label. The other prior criteria are less stringent. The ones that do not depend on the significance level are:

- The *U criterion* (with "U" standing for "unconfidence") uses the average unconfidence over the test sequence, where the *unconfidence* for a test object x_i is the second largest p-value $\min_y \max_{y' \neq y} p_i^{y'}$; small values are preferable. The U criterion in this form was introduced in Fedorova et al. [2013a], but it is equivalent to using the average confidence (one minus unconfidence), which is very common (used, e.g., in Vovk et al. [2005a]). If two conformal transducers have the same average unconfidence (which is presumably a rare event), the criterion compares the average credibilities, where the *credibility* for a test object x_i is the largest p-value $\max_y p_i^y$; smaller values are preferable. (Intuitively, a small credibility is a warning that the test object is unusual, and since such a warning presents useful information and the number of warnings is guaranteed to be small, we want to be warned as often as possible.)

- The *F criterion* uses the average fuzziness, where the *fuzziness* for a test object x_i is defined as the sum of all p-values apart from a largest one, i.e., as $\sum_y p_i^y - \max_y p_i^y$; smaller values are preferable. If two conformal transducers lead to the same average fuzziness, the criterion compares the average credibilities, with smaller values preferable.

Their counterparts depending on the significance level are:

- The *M criterion* uses the percentage of objects x_i in the test sequence for which the prediction set Γ_i^ϵ at level ϵ is *multiple*, i.e., contains more than one label. Smaller values are preferable. When the percentage of multiple predictions is the same for two CPs (which is a common situation: the percentage can well be zero),

the M criterion compares the percentages of empty predictions (larger values are preferable). This is a widely used criterion. In particular, it was used in Vovk et al. [2005a] and earlier papers by the authors.

- The *E criterion* (where "E" stands for "excess") uses the average (over the test sequence, as usual) amount the size of the prediction set exceeds 1. In other words, the criterion gives the average number of excess labels in the prediction sets as compared with the ideal situation of one-element prediction sets. Smaller values are preferable for this criterion. When these averages coincide for two CPs, this criterion compares the percentages of empty predictions (larger values are preferable).

Observed criteria

The prior criteria discussed in the previous subsection treat the largest p-value, or prediction sets of size 1, in a special way. The corresponding criteria of this subsection attempt to achieve the same goal by using the observed label.

These are the observed counterparts of the non-basic prior ϵ-free criteria:

- The *OU* ("observed unconfidence") *criterion* uses the average observed unconfidence over the test sequence, where the *observed unconfidence* for a test example (x_i, y_i) is the largest p-value p_i^y for the *false labels* $y \neq y_i$.

- The *OF criterion* uses the average sum of the p-values for the false labels, i.e., $\frac{1}{k}\sum_{i=l+1}^{l+k}\sum_{y\neq y_i} p_i^y$.

The counterparts of the last group depending on the significance level are:

- The *OM criterion* uses the percentage of observed multiple predictions in the test sequence, where an *observed multiple* prediction is defined to be a prediction set including a false label.

Table 2.1.: The ten criteria studied in Chapter 2: the two basic prior ones in the upper section; the four other prior ones in the middle section; and the four observed ones in the lower section

ϵ-free	**ϵ-dependent**
S (sum of p-values)	*N (number of labels)*
U (unconfidence)	M (multiple)
F (fuzziness)	E (excess)
OU (observed unconfidence)	OM (observed multiple)
OF (observed fuzziness)	*OE (observed excess)*

- The *OE criterion* (OE standing for "observed excess") uses the average number of false labels included in the prediction sets at level ϵ.

For all these four observed criteria smaller values are preferable.

The ten criteria investigated in this chapter are given in Table 2.1. Half of the criteria depend on the significance level ϵ, and the other half are the respective ϵ-free versions.

In the case of binary classification problems, $|Y| = 2$, the number of different criteria of efficiency in Table 2.1 reduces to six: the criteria not separated by a vertical or horizontal line (namely, U and F, OU and OF, M and E, and OM and OE) coincide.

2.3. Optimal idealised conformity measures

This section considers the limiting case of infinitely long training and test sequences. To formalise this setting, we assume that the prediction algorithm is directly given the data-generating probability distribution Q on $\mathbf{Z}$ instead of being given a training sequence and an intuition behind an infinitely long test sequence will be discussed below.

In this setting, instead of conformity measures *idealised conformity measures* will be used. These are functions $A(z, Q)$ of $z \in \mathbf{Z}$ and $Q \in \mathcal{P}(\mathbf{Z})$ (where $\mathcal{P}(\mathbf{Z})$ is the set of all probability measures on $\mathbf{Z}$). The data-generating distribution Q will be fixed for the

rest of this section, and so the corresponding conformity scores are written as $A(z)$.

The *idealised smoothed conformal predictor* corresponding to A outputs the following prediction set $\Gamma^\epsilon(x)$ for each object $x \in \mathbf{X}$ and each significance level $\epsilon \in (0,1)$. For each potential label $y \in \mathbf{Y}$ for x define the corresponding *p-value* as

$$p^y = p(x,y) := Q\{z \in \mathbf{Z} \mid A(z) < A(x,y)\} + \tau Q\{z \in \mathbf{Z} \mid A(z) = A(x,y)\} \tag{2.1}$$

(it would be more correct to write $A((x,y))$, but one pair of parentheses is omitted), where τ is a random number distributed uniformly on $[0,1]$. (The same random number τ is used in (2.1) for all (x,y)). The prediction set is

$$\Gamma^\epsilon(x) := \{y \in \mathbf{Y} \mid p(x,y) > \epsilon\}\,. \tag{2.2}$$

The *idealised smoothed conformal transducer* corresponding to A outputs for each object $x \in \mathbf{X}$ the system of p-values $(p^y \mid y \in \mathbf{Y})$ defined by (2.1); in the idealised case the alternative notation $p(x,y)$ for p^y usually will be used.

The standard properties of validity for conformal transducers and predictors (see Theorem 1 on page 26) simplify in this idealised case as follows:

- If (x,y) is generated from Q, $p(x,y)$ is distributed uniformly on $[0,1]$.
- Therefore, the idealised smoothed conformal predictor makes an error with probability ϵ.

The test sequence being infinitely long is formalised by replacing the use of a test sequence in the criteria of efficiency by averaging with respect to the data-generating probability distribution Q. In the case of the top two and bottom two criteria in Table 2.1 (the ones set in italics) this is done as follows. Let us write $\Gamma^\epsilon_A(x)$ for the $\Gamma^\epsilon(x)$ in (2.2) and $p_A(x,y)$ for the $p(x,y)$ in (2.1) to indicate the dependence on the choice of the idealised conformity measure A. An idealised conformity measure A is:

- *S-optimal* if, for any idealised conformity measure B,

$$\mathbb{E}_{x,\tau} \sum_{y \in \mathbf{Y}} p_A(x, y) \leq \mathbb{E}_{x,\tau} \sum_{y \in \mathbf{Y}} p_B(x, y), \tag{2.3}$$

  where the notation $\mathbb{E}_{x,\tau}$ refers to the expected value when $x \sim Q_{\mathbf{X}}$, $Q_{\mathbf{X}}$ being the marginal distribution of Q on $\mathbf{X}$, and $\tau \sim U[0, 1]$;

- *N-optimal* if, for any idealised conformity measure B and any significance level ϵ,

$$\mathbb{E}_{x,\tau} \left|\Gamma^\epsilon_A(x)\right| \leq \mathbb{E}_{x,\tau} \left|\Gamma^\epsilon_B(x)\right|;$$

- *OF-optimal* if, for any idealised conformity measure B,

$$\mathbb{E}_{(x,y),\tau} \sum_{y' \neq y} p_A(x, y') \leq \mathbb{E}_{(x,y),\tau} \sum_{y' \neq y} p_B(x, y'),$$

  where the lower index (x, y) in $\mathbb{E}_{(x,y),\tau}$ refers to averaging over $(x, y) \sim Q$;

- *OE-optimal* if, for any idealised conformity measure B and any significance level ϵ,

$$\mathbb{E}_{(x,y),\tau} \left|\Gamma^\epsilon_A(x) \setminus \{y\}\right| \leq \mathbb{E}_{(x,y),\tau} \left|\Gamma^\epsilon_B(x) \setminus \{y\}\right|.$$

The idealised versions of the other six criteria listed in Table 2.1 will be defined in Subsection 2.3.2.

2.3.1. Proper criteria of efficiency

The goal in this subsection is to characterise the optimal idealised conformity measures for the four criteria of efficiency that are set in italics in Table 2.1. In the rest of this chapter it is assumed that the object space $\mathbf{X}$ is finite (from the practical point of view,

this is not a restriction); since we consider the case of classification, $|\mathbf{Y}| < \infty$, this implies that the whole example space $\mathbf{Z}$ is finite. To avoid trivialities, it is also assumed that the data-generating probability distribution Q satisfies $Q_{\mathbf{X}}(x) > 0$ for all $x \in \mathbf{X}$ (often curly braces in expressions such as $Q_{\mathbf{X}}(\{x\})$ are omitted).

The *conditional probability (CoP) idealised conformity measure* is

$$A(x, y) := Q(y \mid x). \tag{2.4}$$

This idealised conformity measure was introduced by an anonymous referee of the conference version of Fedorova et al. [2013b], but its non-idealised analogue in the case of regression had been used in Lei and Wasserman [2014] (following Lei et al. [2013] and literature on minimum volume prediction). We say that an idealised conformity measure A is a *refinement* of an idealised conformity measure B if

$$B(z_1) < B(z_2) \Longrightarrow A(z_1) < A(z_2) \tag{2.5}$$

for all $z_1, z_2 \in \mathbf{Z}$. Let $\mathcal{R}(\mathrm{CoP})$ be the set of all refinements of the CoP idealised conformity measure. If C is a criterion of efficiency (one of the ten criteria in Table 2.1), $\mathcal{O}(C)$ stands for the set of all C-optimal idealised conformity measures. The following theorem has been proved by Vladimir Vovk [Vovk et al., 2014].

Theorem 3. $\mathcal{O}(\mathrm{S}) = \mathcal{O}(\mathrm{OF}) = \mathcal{O}(\mathrm{N}) = \mathcal{O}(\mathrm{OE}) = \mathcal{R}(\mathrm{CoP})$.

We say that an efficiency criterion is *proper* if the CoP idealised conformity measure is optimal for it. Theorem 3 shows that four of the ten outlined criteria are proper, namely S, N, OF, and OE (they are set in italic in Table 2.1). In the following section we will see that in general the other six criteria are not proper (or *improper*, as will be said). This terminology will be discussed in the next section.

2.3.2. Improper criteria of efficiency

Next the idealised analogues of the six criteria that are not set in italics in Table 2.1 are defined. An idealised conformity measure A is:

- *U-optimal* if, for any idealised conformity measure B, we have either

$$\mathbb{E}_{x,\tau} \min_y \max_{y' \neq y} p_A(x, y') < \mathbb{E}_{x,\tau} \min_y \max_{y' \neq y} p_B(x, y')$$

  or both

$$\mathbb{E}_{x,\tau} \min_y \max_{y' \neq y} p_A(x, y') = \mathbb{E}_{x,\tau} \min_y \max_{y' \neq y} p_B(x, y')$$

  and

$$\mathbb{E}_{x,\tau} \max_y p_A(x, y) \leq \mathbb{E}_{x,\tau} \max_y p_B(x, y);$$

- *M-optimal* if, for any idealised conformity measure B and any significance level ϵ, we have either

$$\mathbb{P}_{x,\tau}(|\Gamma^\epsilon_A(x)| > 1) < \mathbb{P}_{x,\tau}(|\Gamma^\epsilon_B(x)| > 1) \tag{2.6}$$

  or both

$$\mathbb{P}_{x,\tau}(|\Gamma^\epsilon_A(x)| > 1) = \mathbb{P}_{x,\tau}(|\Gamma^\epsilon_B(x)| > 1)$$

  and

$$\mathbb{P}_{x,\tau}(|\Gamma^\epsilon_A(x)| = 0) \geq \mathbb{P}_{x,\tau}(|\Gamma^\epsilon_B(x)| = 0);$$

- *F-optimal* if, for any idealised conformity measure B, we have either

$$\mathbb{E}_{x,\tau}\Big(\sum_y p_A(x, y) - \max_y p_A(x, y)\Big) < \mathbb{E}_{x,\tau}\Big(\sum_y p_B(x, y) - \max_y p_B(x, y)\Big)$$

or both

$$\mathbb{E}_{x,\tau}\Big(\sum_y p_A(x,y) - \max_y p_A(x,y)\Big) = \mathbb{E}_{x,\tau}\Big(\sum_y p_B(x,y) - \max_y p_B(x,y)\Big)$$

and

$$\mathbb{E}_{x,\tau} \max_y p_A(x,y) \le \mathbb{E}_{x,\tau} \max_y p_B(x,y);$$

- *E-optimal* if, for any idealised conformity measure B and any significance level ϵ, we have either

$$\mathbb{E}_{x,\tau}\big((|\Gamma^\epsilon_A(x)| - 1)^+\big) < \mathbb{E}_{x,\tau}\big((|\Gamma^\epsilon_B(x)| - 1)^+\big)$$

or both

$$\mathbb{E}_{x,\tau}\big((|\Gamma^\epsilon_A(x)| - 1)^+\big) = \mathbb{E}_{x,\tau}\big((|\Gamma^\epsilon_B(x)| - 1)^+\big)$$

and

$$\mathbb{P}_{x,\tau}(|\Gamma^\epsilon_A(x)| = 0) \ge \mathbb{P}_{x,\tau}(|\Gamma^\epsilon_B(x)| = 0);$$

- *OU-optimal* if, for any idealised conformity measure B,

$$\mathbb{E}_{(x,y),\tau} \max_{y' \ne y} p_A(x,y') \le \mathbb{E}_{(x,y),\tau} \max_{y' \ne y} p_B(x,y');$$

- *OM-optimal* if for any idealised conformity measure B and any significance level ϵ,

$$\mathbb{P}_{(x,y),\tau}(\Gamma^\epsilon_A(x) \setminus \{y\} \ne \emptyset) \le \mathbb{P}_{(x,y),\tau}(\Gamma^\epsilon_B(x) \setminus \{y\} \ne \emptyset).$$

The next three definitions follow Vovk et al. [2005a]. The *predictability* of $x \in \mathbf{X}$ is

$$f(x) := \max_{y \in \mathbf{Y}} Q(y \mid x).$$

A *choice function* $\hat{y} : \mathbf{X} \to \mathbf{Y}$ is defined by the condition

$$\forall x \in \mathbf{X} : f(x) = Q(\hat{y}(x) \mid x).$$

Define the *signed predictability (SP) idealised conformity measure* corresponding to $\hat{y}$ by

$$A(x, y) := \begin{cases} f(x) & \text{if } y = \hat{y}(x) \\ -f(x) & \text{if not;} \end{cases} \tag{2.7}$$

an *SP idealised conformity measure* is the SP idealised conformity measure corresponding to some choice function.

To state the following two theorems, the notion of refinement needs to be modified. Let $\mathcal{R}'(\mathrm{SP})$ be the set of all idealised conformity measures A such that there exists an SP idealised conformity measure B that satisfies both (2.5) and

$$B(x, y_1) = B(x, y_2) \implies A(x, y_1) = A(x, y_2) \tag{2.8}$$

for all $x \in \mathbf{X}$ and $y_1, y_2 \in \mathbf{Y}$.

Theorems 4–6 have been proved by Vladimir Vovk [Vovk et al., 2014].

Theorem 4. $\mathcal{O}(\mathrm{U}) = \mathcal{O}(\mathrm{M}) = \mathcal{R}'(\mathrm{SP})$.

Define the *modified conditional probability (MCoP) idealised conformity measure* corresponding to a choice function $\hat{y}$ by

$$A(x, y) := \begin{cases} P(y \mid x) & \text{if } y = \hat{y}(x) \\ P(y \mid x) - 1 & \text{if not;} \end{cases} \tag{2.9}$$

an *MCoP idealised conformity measure* is an idealised conformity measure corresponding to some choice function; $\mathcal{R}'(\mathrm{MCoP})$ is defined analogously to $\mathcal{R}'(\mathrm{SP})$ but using the MCoP

rather than SP idealised conformity measure.

Theorem 5. $\mathcal{O}(\mathrm{F}) = \mathcal{O}(\mathrm{E}) = \mathcal{R}'(\mathrm{MCoP})$.

The *modified signed predictability (MSP) idealised conformity measure* is defined by

$$A(x,y) := \begin{cases} f(x) & \text{if } f(x) > 1/2 \text{ and } y = \hat{y}(x) \\ 0 & \text{if } f(x) \le 1/2 \\ -f(x) & \text{if } f(x) > 1/2 \text{ and } y \neq \hat{y}(x), \end{cases}$$

where f is the predictability function; notice that this definition is unaffected by the choice of the choice function. (When the predictability $f(x) > 1/2$, the value $\hat{y}(x)$ is defined unambiguously.)

Define a set $\mathcal{R}''(\mathrm{MSP})$ in the same way as $\mathcal{R}'(\mathrm{SP})$ but using the MSP idealised conformity measure and except that for $A \in \mathcal{R}''(\mathrm{MSP})$, $f(x) = 1/2$, and $y \neq \hat{y}(x)$ it is allowed $A(x,y) < A(x,\hat{y}(x))$.

Theorem 6. *If* $|\mathbf{Y}| > 2$, $\mathcal{O}(\mathrm{OU}) = \mathcal{O}(\mathrm{OM}) = \mathcal{R}''(\mathrm{MSP})$.

Let us show that from Theorems 4–6 it follows that the six criteria that are not set in italics in Table 2.1 are improper (except for OU and OM when $|\mathbf{Y}| = 2$, of course), i.e. that the CoP idealised conformity measure is not optimal for them.

First, show that the CoP idealised conformity measure is not in the set $\mathcal{R}'(\mathrm{SP})$.

Proof. Assume that $|\mathbf{X}| > 1$ and $|\mathbf{Y}| > 2$. Let the data-generating probability distribution Q on $\mathbf{Z}$ be concentrated on the set

$$\{(x, y^{(1)}), (x, y^{(2)}), (x, y^{(3)})\} \subseteq \mathbf{Z},$$

for some arbitrary fixed $x \in \mathbf{X}$ and $y^{(1)}, y^{(2)}, y^{(3)} \in \mathbf{Y}$. Therefore, without loss of

generality, it may be assumed that $\mathbf{Z} = \{1, 2, 3\}$. And also let

$$Q(1) < Q(2) < Q(3).$$

Let $A(z)$ denote the CoP idealised conformity measure for Q. According to (2.4) this conformity measure specifies the following order on $\mathbf{Z}$:

$$A(1) < A(2) < A(3).$$

For chosen distribution Q the predictability is $Q(3)$, and it is attained only at label 3. Therefore the choice function is unique and the SP idealised conformity measure is defined unambiguously. Denote this conformity measure as $B(z)$; according to (2.7) it specifies the following order on $\mathbf{Z}$:

$$B(1) = B(2) < B(3).$$

Then condition (2.8) implies that for all refinements $B' \in \mathcal{R}'(\mathrm{SP})$ we should have $B'(1) = B'(2)$, but $A(1) < A(2)$, hence $A \notin \mathcal{R}'(\mathrm{SP})$. □

Let us show that the CoP idealised conformity measure is not in the set $\mathcal{R}'(\mathrm{MCoP})$.

Proof. As explained above, it may be assumed that $\mathbf{Z} = \{1, 2, 3\}$. Also let

$$Q(1) < Q(2) = Q(3).$$

As before, $A(z)$ is the CoP idealised conformity measure for Q, and it specifies the following order on $\mathbf{Z}$:

$$A(1) < A(2) = A(3).$$

In this case, the predictability is attained at labels 2 and 3, therefore we have two

different MCoP idealised conformity measures for this Q (the first is corresponding to the choice function $\hat{y} = 2$, and the second is corresponding to $\hat{y} = 3$). Denote these conformity measures as $B_1(z)$ and $B_2(z)$; according to (2.9) the first one (for $\hat{y} = 2$) specifies the ordering:

$$B_1(1) < B_1(3) < B_1(2),$$

and the second one (for $\hat{y} = 3$) specifies the ordering:

$$B_2(1) < B_2(2) < B_2(3),$$

Then condition (2.5) implies that for all refinements $B' \in \mathcal{R}'(\text{MCoP})$ we have $B'(2) \neq B'(3)$, but $A(2) = A(3)$, hence $A \notin \mathcal{R}'(\text{MCoP})$. □

Finally, to show that the CoP idealised conformity measure is not in the set $\mathcal{R}''(\text{MSP})$ we can use the same example as for $\mathcal{R}'(\text{SP})$ additionally assuming that $Q(1) > 1/2$.

Therefore the six criteria that are not set in italics in Table 2.1 are improper as the CoP idealised conformity measure is not optimal for them. This disparaging terminology (in analogy with "improper scoring rules": see, e.g., Gneiting and Raftery 2004) is used since the optimal idealised conformity measures for those criteria have clear disadvantages, such as:

- They may depend on the arbitrary choice of a choice function. In many cases there is a unique choice function, but the possibility of non-uniqueness is still awkward.
- They may encourage "strategic behaviour" (such as ignoring the differences, which may be very substantial, between potential labels other than $\hat{y}(x)$ for a test object x when using the M criterion).

2.4. Empirical Study

This section considers CPs in the realistic setting when the data-generating distribution is unknown. Behaviour of two of the outlined ϵ-free criteria – OF (proper) and U (standard but improper) – is studied empirically. Namely, we define empirical counterparts (based on the KNN method) for the optimal idealised conformity measures for these two criteria, and evaluate their performance using these criteria.

This empirical study uses the benchmark USPS data set described in Appendix A.1 and the original division of this data set into the training and test sequences. The programs implementing prediction methods are written in R, and the results presented in the figures below are for the seed 0 of the R random number generator; however, similar results have been observed in experiments with other seeds.

Based on the KNN method, three conformity measures are derived. Fix a metric on the object space $\mathbb{R}^{256}$; in these experiments two metrics are used – tangent distance (implementation by Daniel Keyser) and Euclidean distance. Given a sequence of examples $(z_1, \dots, z_n)$, $z_i = (x_i, y_i)$, the following three ways of computing conformity scores are considered: for $i = 1, \dots, n$,

- the *KNN-ratio conformity scores* are

$$\alpha_i := \frac{\sum_{j=1}^{K} d_j^{\neq}}{\sum_{j=1}^{K} d_j^{=}}, \tag{2.10}$$

  where $d_j^{\neq}$ are the distances to the objects in $(z_1, \dots, z_n)$ with labels different from y_i and sorted in the increasing order (so that $d_1^{\neq}$ is the smallest distance from x_i to an object x_j with $y_j \neq y_i$), and $d_i^{=}$ are the distances to the objects in $(z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_n)$ labelled as y_i (so that $d_1^{=}$ is the smallest distance from x_i to an object x_j with $j \neq i$ and $y_j = y_i$). This conformity measure has one parameter, K, whose range is $\{1, \dots, 50\}$ in the experiments.

- the *KNN-CoP conformity scores* are

$$\alpha_i := \frac{N_i}{K}, \tag{2.11}$$

  where N_i is the number of objects labelled as y_i among the K nearest neighbours of x_i. This conformity measure is a KNN counterpart of the CoP idealised conformity measure (cf. (2.4)), its parameter K is in the range $\{2, \dots, 50\}$ in the experiments.

- finally, define

$$f_i := \max_y (N_i^y / K),$$

  where N_i^y is the number of objects labelled as y among the K nearest neighbours of x_i, fix some

$$\hat{y}_i \in \arg\max_y (N_i^y / K)$$

  (chosen randomly if $|\arg\max_y (N_i^y/K)| > 1$), and define the *KNN-SP conformity scores* by

$$\alpha_i := \begin{cases} f_i & \text{if } y = \hat{y}_i \\ -f_i & \text{otherwise.} \end{cases} \tag{2.12}$$

  This conformity measure is a KNN counterpart of the SP idealised conformity measure (cf. (2.7)), its parameter K is in the range $\{2, \dots, 50\}$ in the experiments.

These three kinds of conformity measures combined with the two metrics give six CPs. The top panel in Figure 2.1 gives the average unconfidence

$$\text{unconfidence}(K) := \frac{1}{k} \sum_{i=l+1}^{l+k} \min_{y \in \mathbf{Y}} \max_{y' \neq y} p_i^{y'}$$

over the test sequence (so that $k = 2007$) for a range of the values of the parameter K.

The bottom panel in this figure is for the average observed fuzziness

$$\text{observed fuzziness}(K) := \frac{1}{k} \sum_{i=l+1}^{l+k} \sum_{y \in \mathbf{Y} \setminus \{y_i\}} p_i^y$$

over the test sequence for a range of K. The best results in this figure are for KNN-ratio combined with tangent distance for small values of parameter K. For the two other kinds of conformity measures – KNN-CoP and KNN-SP – their relative evaluation changes depending on the kind of a criterion used to measure efficiency. As expected, the KNN-CoP CPs are better under the OF criterion, whereas the KNN-SP CPs are better under the U criterion (cf. Theorems 3 and 4), if we ignore small values of K (when the probability estimates N_i^y/K are very unreliable).

2.5. Conclusion

This chapter has discussed ten criteria of efficiency for conformal prediction. Idealised conformity measures that optimise each of the criteria have been described when the data-generating distribution is known. This sheds light upon the kind of behaviour implicitly encouraged by the criteria even in the realistic case where the data-generating distribution is unknown. Following these ideas, empirical counterparts for the CoP and SP conformity measures have been constructed. Experiments with a benchmark data set have shown that different criteria rank these conformity measures differently. In addition, the results confirm the statements of Theorems 3 and 4 for the non-idealised setting.

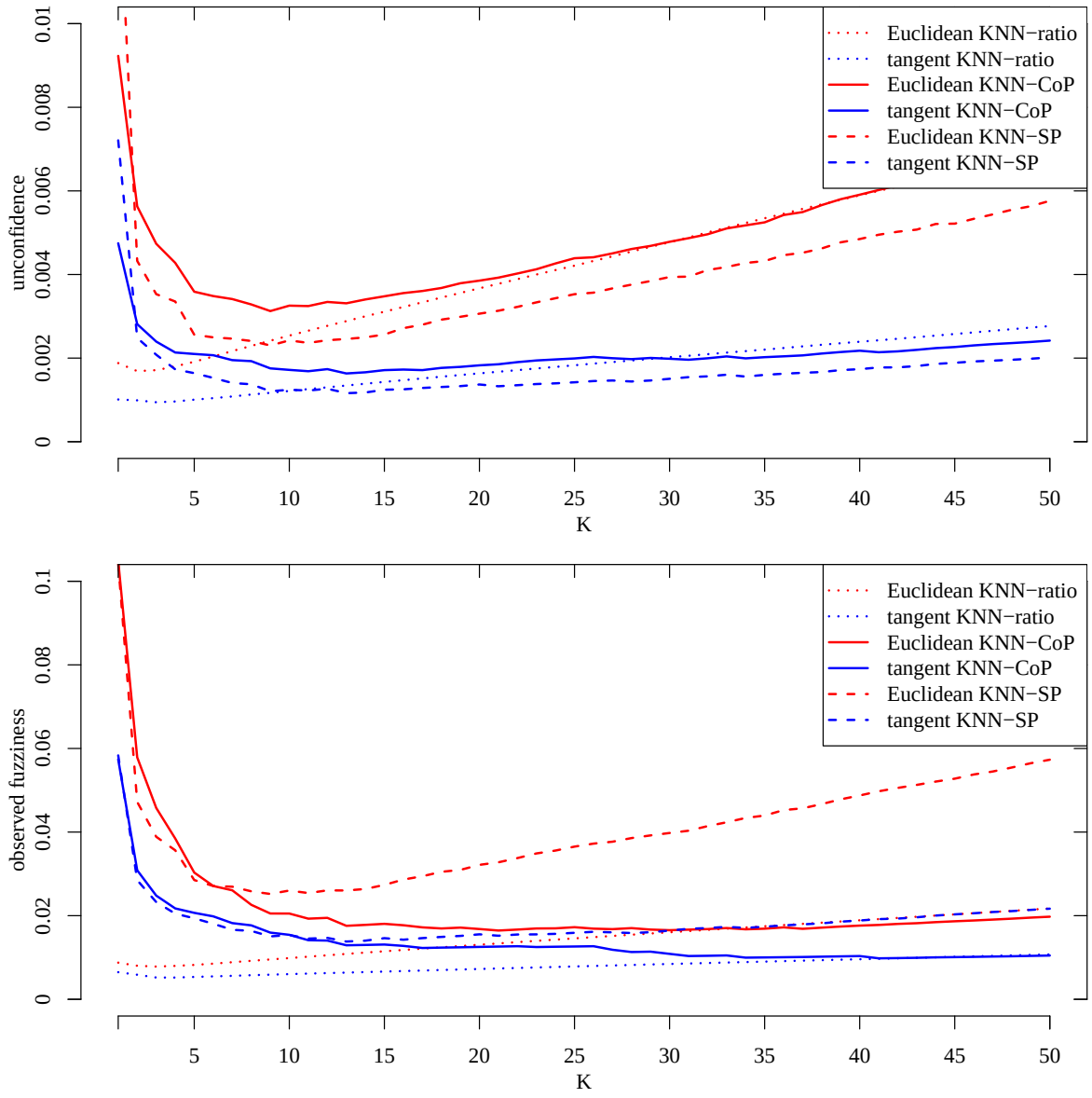

Figure 2.1.: Top plot: average unconfidence for the USPS data set (for different values of parameters). Bottom plot: average observed fuzziness for the USPS data set.

Chapter 3.

Conformal prediction under hypergraphical models

Usually conformal prediction is studied under the exchangeability assumption, since this is one of the weakest assumptions on data. Conversely, other statistical models are useful to describe certain knowledge about the nature of data. The general definition of conformal predictors (CPs) given in Subsection 1.2.3 allows us to construct valid predictors under different models represented as on-line compression models (OCMs). This chapter studies CPs under special OCMs – hypergraphical models. These models are more specific than the exchangeability model and represent assumptions about relationships between data features. Section 3.1 summarises previous similar studies. Section 3.2 formally defines hypergraphical models and reviews their basic properties. Section 3.3 describes conformal prediction in the context of hypergraphical models and introduces two conformity measures for these models. Section 3.4 empirically studies the performance of several CPs defined by these two conformity measures. Section 3.5 describes label conditional CPs under hypergraphical models and also empirically studies their performance. Section 3.6 concludes.

3.1. Introduction

To the best of my knowledge, conformal prediction has been studied, apart from the exchangeability model and its variations, only for the Gauss linear model and Markov model (see [Vovk et al., 2005a, Chapter 8] and [Fedorova et al., 2012a], which is similar to Chapter 5 in this thesis). Hypergraphical OCMs have been used only in the context of Venn rather than conformal prediction (see [Vovk et al., 2005a, Chapter 9]).

This chapter studies conformal prediction under the OCMs known as hypergraphical models (Vovk et al. [2005a], Section 9.2). Such models describe relationships between data features. In the case where every feature is allowed to depend in any way on the rest of the features, the hypergraphical model becomes the exchangeability model. More specific hypergraphical models restrict the dependence in some way. Such restrictions are typical of many real-world problems: for example, different symptoms can be conditionally independent given the disease. A popular approach to such problems is to use Bayesian networks (see, e.g., Cowell et al. [1999]). The definition of Bayesian networks requires a specification of both the pattern of dependence between features and the distribution of the features. Usual methods guarantee a valid probabilistic outcome if the used distributions of features are correct. Several algorithms (see, e.g., Cowell et al. [1999], Chapter 9) are known for estimating the distribution of features; however, the accuracy of such approximations is a major concern in applying Bayesian networks. CPs constructed from hypergraphical OCMs use only the pattern of dependence between the features but do not involve their distribution. This makes conformal prediction based on hypergraphical models more robust and realistic than Bayesian networks. (The notion of a hypergraphical model can be regarded as more general than that of a Bayesian network: the standard algorithms in this area transform Bayesian networks into hypergraphical models by "marrying parents", forgetting the direction of the arrows, triangulation, and regarding the cliques of the resulting graph as the hyperedges; see, e.g., Cowell et al.

[1999], Section 3.2.)

3.2. Definitions

This chapter uses the notation introduced in Section 1.2, and focuses on the case where both an object space $\mathbf{X}$ and a label space $\mathbf{Y}$ are finite. Also, it is assumed that examples are structured, consisting of variables. Hypergraphical structures describe relationships between the variables. Next the hypergraphical structures are defined and after this the definition of hypergraphical models is given.

Hypergraphical structures

A *hypergraphical structure*[1] consists of three elements $(V, \mathcal{E}, \Xi)$:

1. V is a finite set; its elements are called *variables*.
2. $\mathcal{E}$ is a finite collection of subsets of V whose union covers all variables: $\bigcup_{E \in \mathcal{E}} E = V$. Elements of $\mathcal{E}$ are called *clusters*.
3. Ξ is a function that maps each variable $v \in V$ into a finite set (of the values that v can take).

A *configuration* on a set $E \subseteq V$ (we are usually interested in the case where E is a cluster) is an assignment of values to the variables from E; let $\Xi(E)$ be the set of all configurations on E. A *table*[2] on a set E is an assignment of natural numbers or zero to the configurations on E. The *size* of the table is the sum of values that it assigns to different configurations. A *table set* is a collection of tables on the clusters $\mathcal{E}$, one for

[1] The name reflects the fact that the components $(V, \mathcal{E})$ form a hypergraph, where a hyperedge $E \in \mathcal{E}$ can connect more than two vertices.

[2] Generally, a table assigns real numbers to configurations. This thesis only considers *natural tables*, which assign natural numbers or zero to configurations, and "natural" is omitted for brevity.

each cluster $E \in \mathcal{E}$. The number assigned by a table set σ to a configuration on E is called its *σ-count*.

3.2.1. Hypergraphical OCMs

The example space $\mathbf{Z}$ associated with the hypergraphical structure is the set of all configurations on V. One of the variables in V is singled out as the *label variable*, and the configurations on the label variable are denoted $\mathbf{Y}$. All other variables are *object variables*, and the configurations on the object variables are denoted $\mathbf{X}$. Since $\mathbf{Z} = \mathbf{X} \times \mathbf{Y}$, this is a special case of the prediction setting described at the beginning of Section 1.2.

An example $z \in \mathbf{Z}$ *agrees* with a configuration on a set $E \subseteq V$ (or the configuration agrees with the example) if the restriction $z|_E$ of z to the variables in E coincides with the configuration. A table set σ *generated* by a sequence of examples $(z_1, \ldots, z_n)$ assigns to each configuration on each cluster the number of examples in the sequence that agree with the configuration; the size of each table in σ will be equal to the number of examples in the sequence, and this number is called the *size* of the table set. Different sequences of examples can generate the same table set σ, and $\#\sigma$ denotes the number of different sequences generating σ.

Next a special class of OCMs is defined (the definition of an OCM was given on page 28). The *hypergraphical on-line compression model* (HOCM) associated with the hypergraphical structure $(V, \mathcal{E}, \Xi)$ consists of five elements $(\Sigma, \square, \mathbf{Z}, F, B)$, where:

1. The *empty table set* $\square$ is the table set assigning 0 to each configuration.

2. The set Σ is defined by the conditions that $\square \in \Sigma$ and $\Sigma \setminus \{\square\}$ is the set of all table sets σ with $\#\sigma > 0$. The elements $\sigma \in \Sigma$ are called *summaries*.

3. The *forward function* $F(\sigma, z)$, where σ ranges over Σ and z over $\mathbf{Z}$, updates σ by

adding 1 to the σ-count of each configuration which agrees with z.

4. The *backward kernel* B maps each $\sigma \in \Sigma \setminus \{\square\}$ to a probability distribution $B(\sigma)$ on $\Sigma \times \mathbf{Z}$. Example z *agrees* with summary σ if the σ-count for each configuration which agrees with z is positive; if so, obtain a table set, denoted $\sigma \downarrow z$, from σ by subtracting 1 from the σ-count of any configuration that agrees with z. $B(\sigma)$ is defined by
$$B(\{(\sigma \downarrow z, z) \mid \sigma\}) := \frac{\#(\sigma \downarrow z)}{\#\sigma}.$$
Notice that $B(\sigma)$ is indeed a probability distribution, and it is concentrated on the pairs $(\sigma \downarrow z, z)$ such that $F(\sigma \downarrow z, z) = \sigma$.

For the rest of this chapter "hypergraphical models" will be used as a general term for hypergraphical structures and HOCMs when no precision is required.

When discussing hypergraphical models it will be always assumed that the distribution generating examples agrees with the HOCM (as defined on page 30 for an arbitrary OCM). For this special class of OCMs, it is equivalent to assuming that the examples $z_1, z_2, \ldots$ are produced independently by a probability distribution Q on $\mathbf{Z}$ that has a decomposition
$$Q(\{z\}) = \prod_{E \in \mathcal{E}} f_E(z|_E) \tag{3.1}$$
for some functions $f_E : \Xi(E) \to [0, 1]$, $E \in \mathcal{E}$, where z is an example and $z|_E$ its restriction to the variables in E. (For proof see [Vovk et al., 2005a, Corollary 9.6].)

Junction tree structures

An important type of hypergraphical structures is where clusters can be arranged into a "junction tree". For the corresponding HOCMs it will be possible to describe efficient calculations of the backward kernels. If a hypergraphical structure is not of this type it

can be replaced by a more general junction-tree structure (each cluster in the former is a subset of a cluster in the latter) before defining the HOCM.

Let (U, S) denote an undirected tree with U the set of vertices and S the set of edges. Then (U, S) is a *junction tree* for a hypergraphical structure $(V, \mathcal{E}, \Xi)$ if there exists a bijective mapping C from the set of vertices U of the tree to the set $\mathcal{E}$ of clusters of the hypergraphical structure that has the following property: $C_u \cap C_w \subseteq C_v$ whenever a vertex v lies on the path from a vertex u to a vertex w in the tree (let C_x stand for $C(x)$). Not every hypergraphical structure has a junction tree, of course: an example is a hypergraphical structure with three clusters whose intersection is empty but whose pairwise intersections are not. See, e.g., [Cowell et al., 1999, Section 4.3], for further information on junction trees; intuitive examples of junction trees will be given in Section 3.4.

If $s = \{u, v\} \in S$ is an edge of the junction tree connecting vertices u and v then C_s stands for $C_u \cap C_v$. It is convenient to identify vertices u and edges s of the junction tree with the corresponding clusters C_u and sets C_s, respectively.

If $E_1 \subseteq E_2 \subseteq V$ and f is a table on E_2, the *marginalisation* of f to E_1 is the table f^* on E_1 assigning to each $a \in \Xi(E_1)$ the number $f^*(a) = \sum_b f(b)$, where b ranges over the configurations on E_2 such that $b|_{E_1} = a$. If σ is a summary then for $u \in U$ denote σ_u the table that σ assigns to C_u, and for $s = \{u, v\} \in S$ denote σ_s the marginalisation of σ_u (or σ_v) to C_s. The shorthand $\sigma_u(z)$ will be used for the number assigned to the restriction $z|_{C_u}$ by the table for the vertex u and $\sigma_s(z)$ for the number assigned to $z|_{C_s}$ by the marginal table for the edge s. Consider the HOCM corresponding to the junction tree (U, S). Denote the weight assigned by $B(\sigma)$ to $(\sigma \downarrow z, z)$ as $P_\sigma(z)$:

$$P_\sigma(z) := B(\{(\sigma \downarrow z, z) \mid \sigma\}).$$

It has been proved (Vovk et al. [2005a], Lemma 9.5) that

$$P_\sigma(z) = \frac{\prod_{u \in U} \sigma_u(z)}{n \prod_{s \in S} \sigma_s(z)}, \tag{3.2}$$

where n is the size of σ. If any of the factors in (3.2) is zero then the whole ratio is set to zero.

3.3. Conformal prediction under HOCMs

Consider a training sequence $(z_1, \ldots, z_l)$ generated by a distribution that agrees with an HOCM $M = (\Sigma, \square, \mathbf{Z}, F, B)$. The goal is to predict the label for a new object x.

The definition of a conformity measure for an HOCMs is the same as that for an arbitrary OCM (see page 31). Next the definition of the CP determined by such a conformity measure is restated in the context of HOCMs.

Consider a conformity measure A for the HOCM M. For each $y \in \mathbf{Y}$ denote $\sigma^* \in \Sigma$ the table set generated by the sequence $(z_1, \ldots, z_l, (x, y))$. For $z \in \mathbf{Z}$ such that $\sigma^* \downarrow z$ is defined denote the conformity scores as

$$\alpha_z := A\left(\sigma^* \downarrow z, z\right) \tag{3.3}$$

(notice that $\alpha_{(x,y)}$ is always defined). The *p-value* for y, denoted p^y, can be written as

$$p^y = \sum_{z : \alpha_z < \alpha_{(x,y)}} P_{\sigma^*}(z) + \tau \sum_{z : \alpha_z = \alpha_{(x,y)}} P_{\sigma^*}(z) \tag{3.4}$$

(cf. (1.4) on page 31), where $\tau \sim \mathbf{U}[0, 1]$ is a random number from the uniform distribution on $[0, 1]$, $P_{\sigma^*}(z)$ is the backward kernel, as defined above, and the sums involve only those $z \in \mathbf{Z}$ for which α_z is defined. Then for a significance level ϵ the *conformal*

predictor Γ based on A outputs the prediction set

$$\Gamma^{\epsilon}(z_1, \ldots, z_l, x, \tau) := \{y \in \mathbf{Y} : p^y > \epsilon\}. \tag{3.5}$$

Next two examples of conformity measures for HOCMs will be described.

3.3.1. Conformity measures for HOCMs

Earlier (in Section 2.3) idealised conformity measures that optimise different criteria of efficiency for conformal prediction were discussed. This section defines conformity measures for HOCMs that are hypergraphical counterparts of two of those idealised conformity measures. (In the current setting, the empirical distribution can be estimated using a hypergraphical model, and there is no need to involve the KNN approximation as it was done on page 52.)

Consider a summary σ and an example (x, y). The *hypergraphical conditional probability conformity measure* is defined by

$$A(\sigma, (x, y)) := P_{\sigma^*}(y \mid x) := \frac{P_{\sigma^*}((x, y))}{\sum_{y' \in \mathbf{Y}} P_{\sigma^*}((x, y'))}, \tag{3.6}$$

where $\sigma^* := F(\sigma, (x, y))$ and $P_{\sigma^*}((x, y))$ is the backward kernel. In other words, $A(\sigma, (x, y))$ is the conditional probability $P_{\sigma^*}(y \mid x)$ of y given x under P_{σ^*}. The conditional probability $P_{\sigma^*}(y \mid x)$ can be easily computed using (3.2).

Define the *predictability* of an object $x \in \mathbf{X}$ as

$$f(x) := \max_{y \in \mathbf{Y}} P_{\sigma^*}(y \mid x), \tag{3.7}$$

the maximum of conditional probabilities. (If the predictability of an object is close to 1 then the object is "easily predictable".) Fix a *choice function* $\hat{y} : \mathbf{X} \to \mathbf{Y}$ defined by

the condition

$$\forall x \in \mathbf{X} : f(x) = P_{\sigma^*}(\hat{y}(x) \mid x).$$

(The function maps each object x to one of the labels at which the maximum in (3.7) is attained.) The *hypergraphical signed predictability conformity measure* is defined by

$$A(\sigma, (x, y)) := \begin{cases} f(x) & \text{if } y = \hat{y}(x) \\ -f(x) & \text{otherwise.} \end{cases} \tag{3.8}$$

3.4. Empirical study

This section empirically studies the performance of CPs corresponding to conformity measures (3.6) and (3.8) using LED data sets. First the experimental setting is discussed, then hypergraphical structures for the data sets are described, and after that the experimental results are presented.

Set-up for experiments

Let us consider conformal prediction in the on-line mode (Protocol 1.1 on page 24). Reality generates examples (x_n, y_n) from a probability distribution Q satisfying (3.1) for some hypergraphical structure. Predictor uses a CP Γ under the corresponding hypergraphical model to output the prediction set

$$\Gamma^\epsilon_n := \Gamma^\epsilon(x_1, y_1, \ldots, x_{n-1}, y_{n-1}, x_n, \tau_n)$$

at each significance level ϵ.

Considerations in this section are restricted to the problems where the hypergraphical model used for computing the p-values is known to be correct; therefore, the predictions will always be valid, and there is no need to test validity experimentally.

In Chapter 2 we discussed ten different criteria of efficiency for conformal prediction. For this chapter two ϵ-free and two ϵ-dependent criteria are considered.

The ϵ-free criteria are: the cumulative observed fuzziness OF_n over the first n steps

$$\mathrm{OF}_n := \sum_{i=1}^{n} \sum_{\substack{y \in \mathbf{Y} \\ y \neq y_i}} p_i^y$$

(see page 40), and the cumulative unconfidence U_n over the first n steps

$$\mathrm{U}_n := \sum_{i=1}^{n} \min_{y \in \mathbf{Y}} \max_{\substack{y \in \mathbf{Y} \\ y \neq y_i}} p_i^y$$

(see page 39). As discussed before, the criteria work in the same direction: the smaller the better.

The ϵ-dependent criteria are: the average observed excess over the first n steps defined by

$$\mathrm{OE}_n^\epsilon := \frac{1}{n} \sum_{i=1}^{n} |\Gamma_i^\epsilon \setminus \{y_i\}|$$

at each significance level $\epsilon \in (0, 1)$ (see page 41), and the percentage of multiple predictions defined by

$$\mathrm{mult}_n^\epsilon := \begin{cases} 1 & \text{if } |\Gamma_n^\epsilon| > 1 \\ 0 & \text{otherwise} \end{cases} \qquad \text{and} \qquad \mathrm{M}_n^\epsilon := \frac{1}{n} \sum_{i=1}^{n} \mathrm{mult}_i^\epsilon$$

at each significance level $\epsilon \in (0, 1)$ (see page 39). It is desirable for both these values to be close to 0.

Chapter 2 pointed out that if the data distribution is known:

- the idealised conditional probability conformity measure (2.4) is optimal in the sense of OE_n^ϵ and in the sense of OF_n (see Theorem 3 on page 44);

- the idealised signed predictability conformity measure (2.7) is optimal in the sense of M_n^{ϵ} and in the sense of U_n (see Theorem 4 on page 47).

This empirical study will demonstrate that the conclusions hold for the hypergraphical counterparts of the two idealised conformity measures.

LED data

LED data sets used for the experiments are described in Appendix A.2. First the data-generating distribution is discussed and then hypergraphical structures used for the data are defined.

Following the notation in Appendix A.2, let $(S_0, \ldots, S_6, C)$ be the vector of random variables corresponding to the seven LED segments and the label, and let $(s_0, \ldots, s_6, c)$ be an example. Denote the probability of noise for each LED as p_{noise} (it is the same for all LEDs). According to the data-generating mechanism the probability of the example decomposes as

$$Q\left(\{(s_0, \ldots, s_6, c)\}\right) = Q_7\left(C = c\right) \cdot \prod_{i=0}^{6} Q_i\left(S_i = s_i \mid C = c\right), \tag{3.9}$$

where Q_7 is the uniform distribution on the decimal digits and

$$Q_i\left(S_i = s_i \mid C = c\right) := \begin{cases} 1 - p_{\text{noise}} & \text{if } s_i = s_i^c \\ p_{\text{noise}} & \text{otherwise,} \end{cases} \qquad i = 0, \ldots, 6, \tag{3.10}$$

$(s_0^c, \ldots, s_6^c, c)$ representing the ideal image for the label c.

Hypergraphical assumptions for LED data

Consider two hypergraphical models that agree with the decomposition (3.9). These models make different assumptions about the pattern of dependence between the features

and the label; they do not depend on a particular probability of noise p_{noise} or the fact that the same value of p_{noise} is used for all LEDs. For both hypergraphical structures the set of variables is $V := \{s_0, \dots, s_6, c\}$.

Nontrivial hypergraphical model. Consider the hypergraphical structure with the clusters $\mathcal{E} := \{\{s_i, c\} : i = 0, \dots, 6\}$. A junction tree for this hypergraphical structure can be defined as a chain with vertices $U := \{u_i : i = 0, \dots, 6\}$ and the bijection $C_{u_i} := \{s_i, c\}$. By saying that U is a chain it is implied that there are edges connecting vertices u_0 and u_1, u_1 and u_2, u_2 and u_3, u_3 and u_4, u_4 and u_5, and u_5 and u_6 (and these are the only edges). It is clear that this is a junction tree and that $C_s = \{c\}$ for each edge s. It is also clear from (3.9) that the assumption (3.1) is satisfied; e.g., one can set

$$f_{\{s_0,c\}}(s_0, c) := Q_7(C = c) \cdot Q_0(S_0 = s_0 \mid C = c);$$
$$f_{\{s_i,c\}}(s_i, c) := Q_i(S_i = s_i \mid C = c), \quad i = 1, \dots, 6.$$

Exchangeability model. The hypergraphical model with no information about the pattern of dependence between the attributes and the label is the exchangeability model. The corresponding hypergraphical structure has one cluster, $\mathcal{E} := \{V\}$. The junction tree is the one vertex associated with V and no edges.

Experiments

Each LED data set for the experiments consists of $10,000$ examples generated with the probability of noise $p_{\text{noise}} = 1\%$ in the data-generating program (see Appendix A.2). The text below assumes that the reader can see Figures 3.1–3.4 in colour (the web-links for coloured pictures are available); the colours become different shades of grey in black-and-white. Hopefully, the descriptions will be detailed enough for the reader to identify the most important graphs unambiguously.

Each of the figures corresponds to an efficiency criterion for conformal prediction; namely, Figure 3.1 [3] plots the cumulative observed fuzziness OF_n versus $n = 1, \ldots, 10000$ in the on-line prediction protocol, Figure 3.2 [4] plots the cumulative unconfidence U_n versus $n = 1, \ldots, 10000$, Figure 3.3 [5] plots the final average observed excess of predictions $\mathrm{OE}^{\epsilon}_{10000}$ versus $\epsilon \in [0, 0.05]$ and Figure 3.4 [6] plots the final percentage of multiple predictions $\mathrm{M}^{\epsilon}_{10000}$ versus $\epsilon \in [0, 0.05]$. Two conformity measures are considered: the hypergraphical conditional probability (CoP) conformity measure (3.6) and the hypergraphical signed predictability (SP) conformity measure (3.8). The graphs corresponding to the former are represented in the plots as solid lines, and the graphs corresponding to the latter are represented as dashed lines.

Two of the graphs in each figure correspond to idealised predictors and are drawn only for comparison, representing an unachievable ideal goal. In the idealised case we know the true distribution for the data (given by (3.9), (3.10), and $p_{\mathrm{noise}} = 1\%$). The true distribution is used instead of the backward kernel P_{σ^*} in both (3.4) and (3.6) for the hypergraphical CoP conformity measure and in both (3.4) and (3.8) for the hypergraphical SP conformity measure. It yields the ideal results (the two red lines in the plots) for the two conformity measures, CoP and SP. At least one of them gives the best results in each of the figures (remember that for these four criteria the lower the better).

In addition, for each of the two conformity measures four realistic predictors are constructed (which are CPs, unlike the idealised ones). The *pure hypergraphical CP* (represented by blue lines in the plots) is obtained using the nontrivial hypergraphical model both when computing p-values (see (3.4)) and when computing the conformity measure ((3.6) in the case of hypergraphical CoP and (3.8) in the case of hypergraphical

[3] `http://www.cs.rhul.ac.uk/~valentina/thesis_pics/of_hm.pdf`
[4] `http://www.cs.rhul.ac.uk/~valentina/thesis_pics/u_hm.pdf`
[5] `http://www.cs.rhul.ac.uk/~valentina/thesis_pics/oe_hm.pdf`
[6] `http://www.cs.rhul.ac.uk/~valentina/thesis_pics/m_hm.pdf`

SP). Analogously the exchangeability model is used to obtain the *pure exchangeability CP* (green lines in the plots). The two *mixed CPs* (black and yellow lines) are obtained when different models are used to compute the p-values and the conformity scores.

The intuition behind the pure and mixed CPs can be explained using the distinction between hard and soft models made in Vovk et al. [2005b]. The model used when computing the p-values (see (3.4)) is the hard model; the validity of the CP depends on it. The model used when computing conformity scores (see (3.6) and (3.8)) is the soft model (as mentioned on page 24); when it is violated, validity is not affected, although efficiency can suffer. The true probability distribution (3.9) conforms to both the exchangeability model and the nontrivial hypergraphical model; therefore, all four CPs are automatically valid, and only their efficiency is investigated. (In the current context, it is obvious that the exchangeability model is more general than the nontrivial hypergraphical model, but we can also apply the criterion given in Vovk et al. [2005a], Proposition 9.2.)

In the legends of Figures 3.1–3.4, the hard model used is indicated after "pv" (the way of computing the p-values), and the soft model used is indicated after "CM" (the conformity measure); "exch" refers to the exchangeability model, and "hgr" refers to the nontrivial hypergraphical model.

The most interesting graphs in Figures 3.1–3.4 are the black ones, corresponding to the exchangeability model as the hard model and the nontrivial hypergraphical model as the soft model. The performance of the corresponding CPs is typically better than, or at least close to, the performance of any of the remaining realistic predictors. The fact that the validity of these CPs only depends on the exchangeability assumption makes them particularly valuable. The yellow graphs correspond to the nontrivial hypergraphical model as the hard model and the exchangeability model as the soft model; the performance of the corresponding CPs is very poor in these experiments.

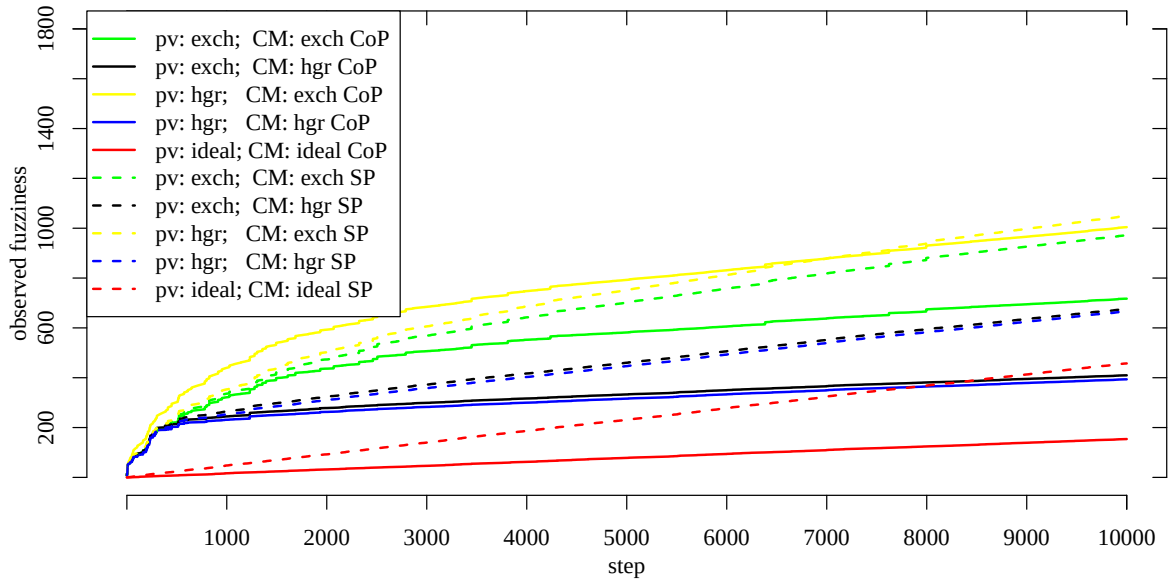

Figure 3.1.: Cumulative observed fuzziness for on-line predictions. The results are for the LED data set with 1% of noise and 10,000 examples. (In the legend "pv" stands for the model for calculating p-values, "CM" stands for the model for constructing conformity measure.)

The data-generating program is written in C, and the data-processing programs are written in R. For results presented in the figures, in both cases the seed of the pseudo-random number generator is set to 0, but the corresponding tables and experiments not included in this chapter confirm that conclusions given below apply to other seeds as well. Now each of the figures and the corresponding tables will be discussed separately.

Table 3.1.: The final values of the cumulative observed fuzziness in Figure 3.1 for the black and blue graphs.

Seed (10^4)	0	1	...	99	Average	St. dev.
pv: exch; CM: hgr CoP	409.6	422.2	...	402.7	407.0	24.75
pv: hgr; CM: hgr CoP	393.6	402.2	...	385.2	384.4	23.99
pv: exch; CM: hgr SP	675.4	747.0	...	717.1	676.1	55.87
pv: hgr; CM: hgr SP	666.1	729.8	...	701.2	657.2	53.90

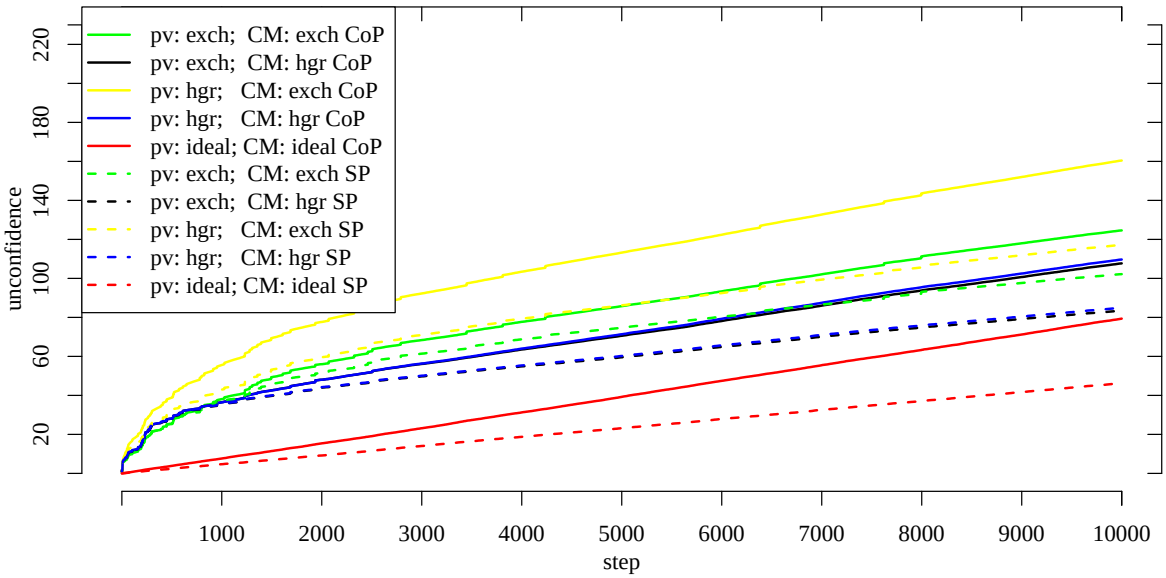

Figure 3.2.: Cumulative unconfidence for on-line predictions. The results are for the LED data set with 1% of noise and 10,000 examples. (In the legend "pv" stands for the model for calculating p-values, "CM" stands for the model for constructing conformity measure.)

Figure 3.1 shows the cumulative observed fuzziness OF_n. As mentioned earlier, the optimal conformity measure to use is CoP. Indeed, for this criterion the predictors based on the CoP conformity measure outperform the predictors based on the SP conformity measure (the solid lines are below the dashed lines of the same colour), as expected. The bottom graph corresponds to the idealised CoP predictor; the idealised SP predictor is

Table 3.2.: The final values of the cumulative unconfidence in Figure 3.2 for the black and blue graphs.

Seed (10^4)	0	1	...	99	Average	St. dev.
pv: exch; CM: hgr CoP	107.69	108.03	...	106.96	106.23	9.852
pv: hgr; CM: hgr CoP	109.68	107.80	...	107.80	105.83	9.821
pv: exch; CM: hgr SP	83.40	90.26	...	89.09	82.19	7.066
pv: hgr; CM: hgr SP	84.89	90.56	...	89.45	82.39	6.809

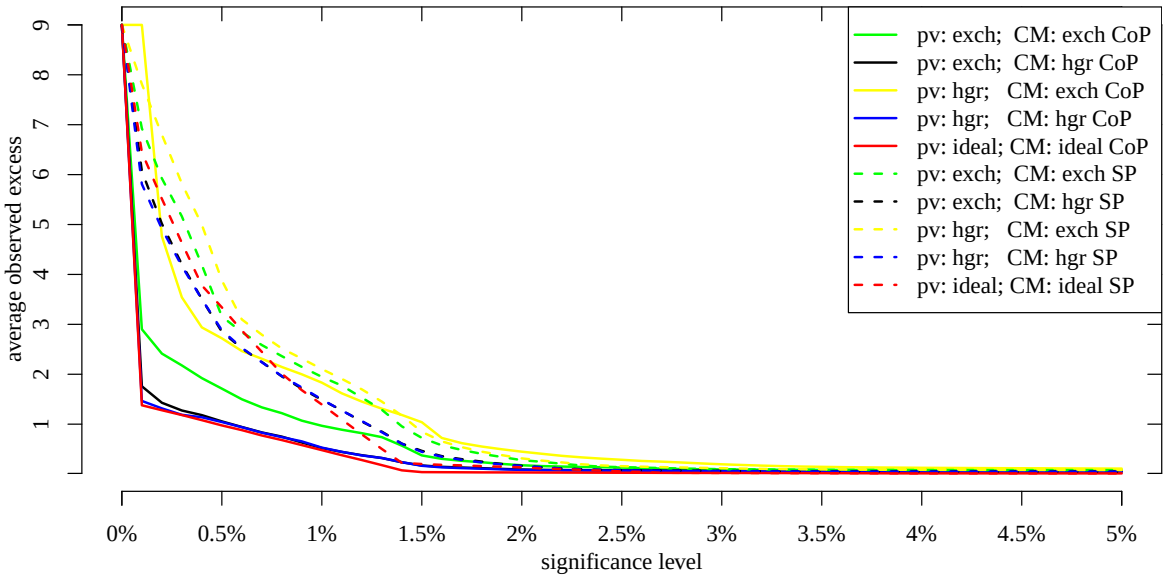

Figure 3.3.: The final average observed excess for significance levels between 0% and 5%. The results are for the LED data set with 1% of noise and 10,000 examples. (In the legend "pv" stands for the model for calculating p-values, "CM" stands for the model for constructing conformity measure.)

the second best most of the time, but at the end it is overtaken by the black and blue graphs corresponding to the CPs based on the CoP conformity measure using the nontrivial hypergraphical model. The black and blue graphs are very close; the blue one is slightly lower but the CP corresponding to the black one still appears preferable as its validity only depends on the weaker exchangeability assumption.

Table 3.3.: The final average observed excess in Figure 3.3 for the significance level 1% and for the black and blue graphs.

Seed (10^4)	0	1	...	99	Average	St. dev.
pv: exch; CM: hgr CoP	0.5218	0.5115	...	0.5197	0.5451	0.1237
pv: hgr; CM: hgr CoP	0.5299	0.4868	...	0.5025	0.5228	0.1216
pv: exch; CM: hgr SP	1.4870	1.7030	...	1.6648	1.4147	0.3598
pv: hgr; CM: hgr SP	1.4959	1.6327	...	1.6359	1.3806	0.3432

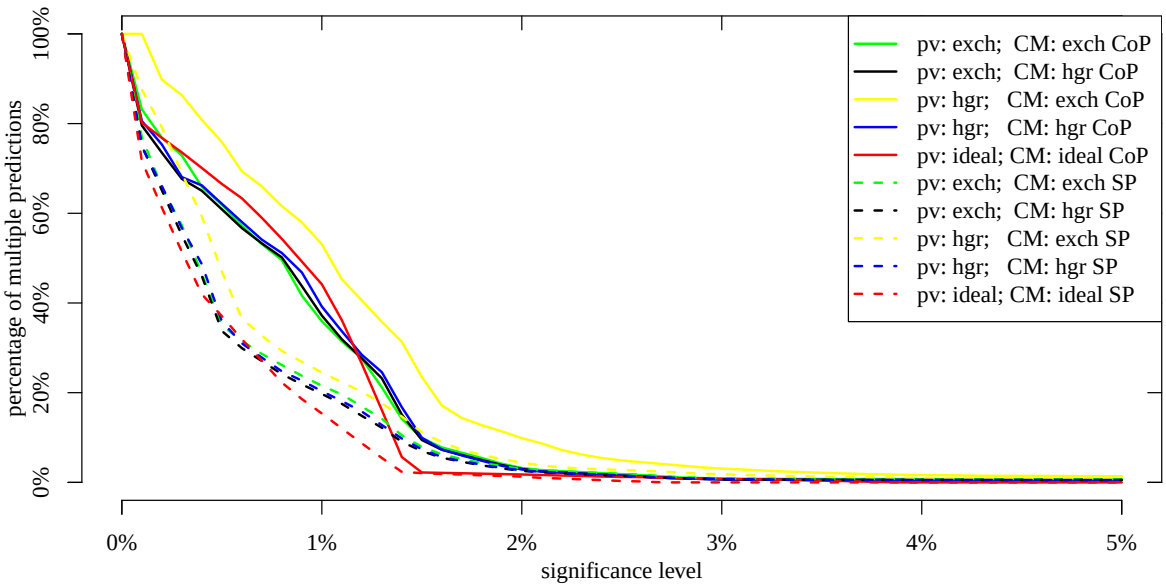

Figure 3.4.: The final percentage of multiple predictions for significance levels between 0% and 5%. The results are for the LED data set with 1% of noise and 10,000 examples. (In the legend "pv" stands for the model for calculating p-values, "CM" stands for the model for constructing conformity measure.)

Table 3.1 shows the final values of the cumulative observed fuzziness in Figure 3.1 for the four most important graphs (two black and two blue) for several seeds. The values of the seed are given in the units of 10,000 (so that 0 stands for 0, 1 for 10,000, 2 for 20,000, etc.), which is the minimal step to ensure that different experiments are based on completely different pseudorandom numbers (when the seed is initialised to n, the

Table 3.4.: The final percentage of multiple predictions in Figure 3.4 for the significance level 1% and for the black and blue graphs.

Seed (10^4)	0	1	...	99	Average	St. dev.
pv: exch; CM: hgr CoP	0.3720	0.4046	...	0.4109	0.3812	0.0905
pv: hgr; CM: hgr CoP	0.3920	0.4047	...	0.4128	0.3815	0.0896
pv: exch; CM: hgr SP	0.1972	0.2425	...	0.2478	0.1918	0.0515
pv: hgr; CM: hgr SP	0.2034	0.2437	...	0.2502	0.1962	0.0489

successive calls to the R pseudorandom number generator produce the pseudorandom numbers corresponding to the seeds n, $n+1$, $n+2$, etc.); the "10^4" in parentheses serves as a reminder of this. The last two columns of this and other tables give aggregate values: column "Average" gives the average of all the 100 values for the seeds 0–99, and column "St. dev." gives the standard estimate of the standard deviation computed from those 100 values (namely, the square root of the standard unbiased estimate of the variance). The table confirms that the black and blue graphs are close to each other on average (see the penultimate column), although there is a clear tendency for the blue ones to be lower: see the last column (to obtain an estimate of the standard deviation of the average, the value given in the last column should be divided by 10).

Figure 3.2 shows the cumulative unconfidence U_n, and so the right conformity measure to use is SP; and indeed, all SP graphs lie below their CoP counterparts. The two bottom graphs are the ones corresponding to idealised predictors; the graph corresponding to the CoP idealised predictor, however, has a suboptimal slope. Of the realistic predictors, the lowest graph is the black SP one (but the blue SP graph, corresponding to the pure hypergraphical CP, is very close). Table 3.2 confirms that each black graph is very close to the corresponding blue graph on average, but the accuracy of the experiments is insufficient to say which tends to be lower.

Figure 3.3 shows the average observed excess of predictions after $10,000$ prediction steps as function of the significance level. For small significance levels the predictors based on the CoP conformity measure perform better, again confirming the theoretical results mentioned earlier. The black CoP graph is very close to the blue CoP graph, corresponding to the pure hypergraphical predictor, except for very low significance levels when the average excess exceeds 1. According to Table 3.3, the accuracy of the experiments is insufficient to tell whether the two blue graphs tend to be lower than the corresponding black ones at the significance level 1% for these data.

Figure 3.4 shows the percentage of multiple predictions after observing $10,000$ examples as function of the significance level. For small significance levels the percentage of the multiple predictions is smaller for the predictors based on the SP conformity measure, again as expected. The performance of the CP corresponding to the black SP graph is again remarkably good, better than that of any other realistic predictor, although very close to the blue SP graph. The closeness at the significance level 1% is confirmed by Table 3.4.

To summarise, the experiments have shown that the best performance of the realistic predictors is achieved when the soft model is the nontrivial hypergraphical model; the choice of the stronger hard model does not change the performance too much, therefore one can continue to use the exchangeability model as the hard model. In the presented experiments both the exchangeability and hypergraphical models are correct for data, therefore using any of the models for computing p-values results in valid predictions. However, in practice (for example, in medical applications) it is often the situation when exchangeability is satisfied but a hypergraphical model is "almost correct". In this case, if the hypergraphical model is used for computing p-values, predictions may be not valid. To avoid such problems it is suggested to use the weaker exchangeability assumption for computing p-values and the stronger hypergraphical assumption for constructing conformity measures. In addition, in the experiments we have seen the examples confirming the results of Theorem 3 (page 44) and Theorem 4 (page 47) in the non-idealised setting: the hypergraphical conditional probability conformity measure (3.6) is optimal in terms of OF and OE criteria (Figures 3.1 and 3.3), and the hypergraphical signed predictability conformity measure (3.8) is optimal in terms of U and M criteria (Figures 3.2 and 3.4).

3.5. Label conditional conformal prediction under HOCMs

The usual notion of validity for CPs (see page 32) is unconditional; the overall probability of error being equal to the significance level ϵ does not prevent the probability of error for different classes (such as 0s, 1s, etc. in the case of LED data sets) being different from ϵ, as long as the average probability over all classes remains ϵ. This section describes label conditional conformal prediction under hypergraphical models, which achieves class-wise validity. It starts with the formal definition of the method and follows by an empirical study of these CPs using LED data sets.

3.5.1. Definition and properties of label conditional conformal prediction

In general, examples can be divided in a natural way into a finite number of categories (for example, each category corresponds to a label, or to a kind of objects). We say that a CP is *category-wise valid* if for any significance level $\epsilon \in (0, 1)$ the probability of errors within each category is ϵ. And analogously, a conformal transducer is *category-wise valid* if p-values (calculated in the on-line mode and corresponding to the true labels) are independent and distributed uniformly on $[0, 1]$ for each of the categories. The automatic validity of CPs (Theorem 2 on page 31) does not guarantee their validity within individual categories: for some categories error rates can be higher than the significance level and it is balanced by lower error rates for other categories. For conformal transducers it means that p-values corresponding to different labels are concentrated in different parts of the unit interval, but altogether their distribution is uniform. A modification of CPs that achieve the category-wise validity, called Mondrian CPs, were introduced in [Vovk et al., 2005a, Section 4.5] under the exchangeability assumption. This section studies

Mondrian CPs and transducers under hypergraphical models focusing on the categories corresponding to labels; the corresponding categories are called classes, as usual, and the corresponding Mondrian CPs are called label conditional CPs.

Formally, *hypergraphical label conditional conformal predictors* are defined in the same way as hypergraphical CPs in Section 3.3 (see (3.3) – (3.5)) except that the definition of p-values (3.4) is modified as follows:

$$p^y := \frac{\sum_{\substack{(x',y')\in\mathbf{Z}:y'=y,\\ \alpha_{(x',y')}<\alpha_{(x,y)}}} P_{\sigma^*}((x',y')) + \tau \sum_{\substack{(x',y')\in\mathbf{Z}:y'=y,\\ \alpha_{(x',y')}=\alpha_{(x,y)}}} P_{\sigma^*}((x',y'))}{\sum_{(x',y')\in\mathbf{Z}:y'=y} P_{\sigma^*}((x',y'))}, \tag{3.11}$$

where, as usual, the sums involve only those $(x',y') \in \mathbf{Z}$ for which $\alpha_{(x',y')}$ is defined. In this thesis sometimes CPs defined in Section 3.3 will be referred to as "unconditional" CPs. To summarise, for the hypergraphical label conditional CP the conformity scores are defined by (3.3), the p-values are defined by (3.11), and the prediction sets are defined by (3.5).

As in the unconditional case, one can use both the hypergraphical conditional probability (CoP) conformity measure (3.6) and the hypergraphical signed predictability conformity measure (3.8) when computing the conformity scores (3.3).

3.5.2. Empirical study

This subsection studies the performance of unconditional and label conditional CPs under hypergraphical models. First we look at the category-wise validity of these predictors and then we compare their efficiency.

As before, the used LED data set consists of 10000 examples generated with $p_{\text{noise}} = 1\%$. (The results presented in this section are for the seed 0 of the pseudorandom generator, for both the data-generating and data-processing programs.) The two hypergraphical models have been described in Section 3.4. Predictors for these experiments are based on the hypergraphical CoP conformity measure (3.6).

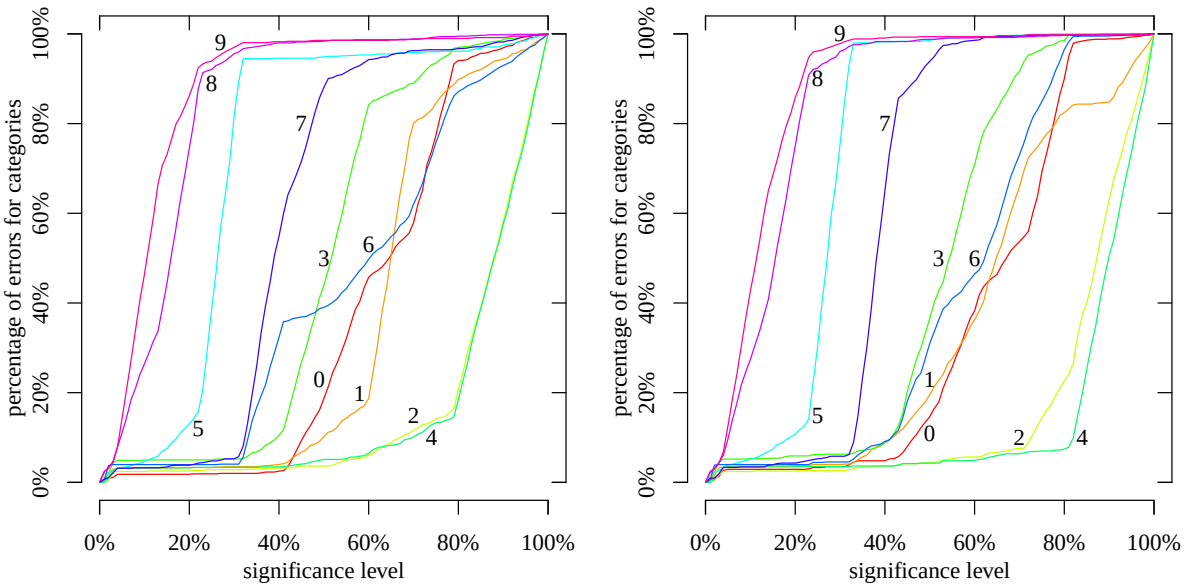

Figure 3.5.: The final percentage of errors for categories for (unconditional) conformal prediction under hypergraphical models; colours are for categories corresponding to labels. The predictions are not category-wise valid. The left plot is for the pure exchangeability conformal predictor and the right plot is for the pure hypergraphical conformal predictor. The results are for the LED data set of 10000 examples with 1% of noise.

Category-wise validity

In the first experiment the final percentage of errors within different categories is assessed. For each of ten categories corresponding to labels $y \in \{0, \ldots, 9\}$ and at each significance level $\epsilon \in (0, 1)$ the final percentage of errors is calculated by

$$\mathrm{Err}^{y,\epsilon} := \frac{|\{i = 1, \ldots, 10000 : y_i = y, y_i \notin \Gamma_i^\epsilon\}|}{|\{i = 1, \ldots, 10000 : y_i = y\}|}. \tag{3.12}$$

Four CPs are constructed: the *pure exchangeability CP* (the exchangeability model is used for p-values (3.4) and for conformity scores (3.6)), the *pure hypergraphical CP* (the nontrivial hypergraphical model is used in (3.4) and (3.6)), the *pure exchangeability label conditional CP* (the exchangeability model is used for (3.11) and for (3.6)), and the *pure hypergraphical label conditional CP* (the nontrivial hypergraphical model is used in

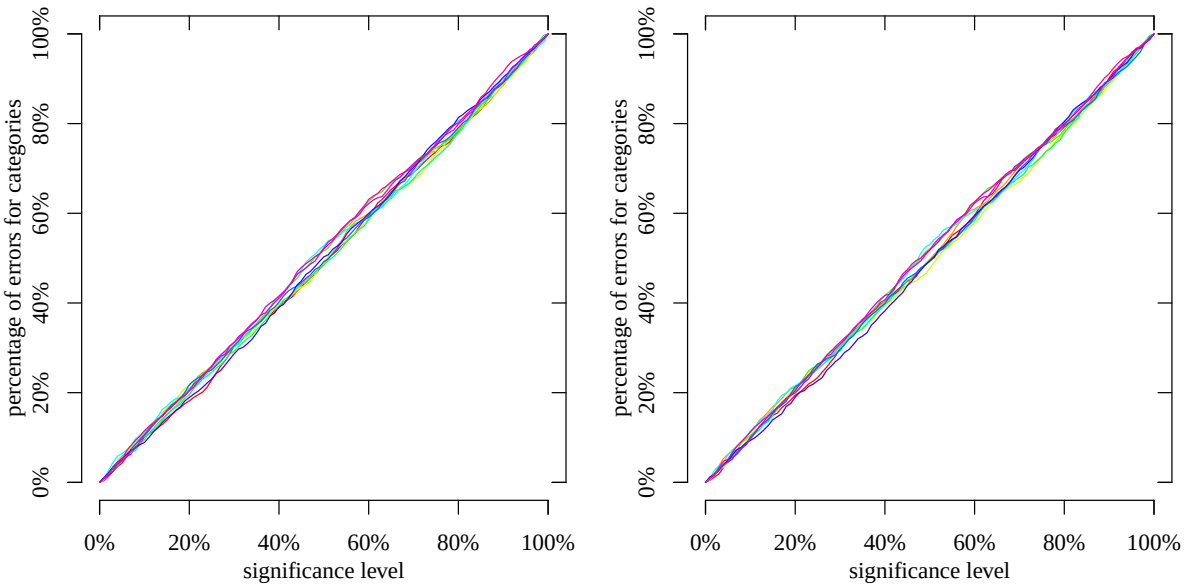

Figure 3.6.: The same as Figure 3.5 for label conditional conformal prediction under hypergraphical models. The predictions are category-wise valid.

(3.11) and (3.6)).

All these four predictors make predictions in the on-line mode, and the final percentage of errors (3.12) is calculated for these predictions and significance levels between 0% and 100%. The percentage of errors plotted against the significance level will be called the *calibration graph*. For valid predictions the calibration graph is the diagonal extending from the bottom left corner (no errors at the significance level 0%, which is achieved when all prediction sets are the whole label set) to the top right corner (errors on each prediction trial at the significance level 100%, which is the result of all predictions being the empty predictions).

Figure 3.5 shows the calibration graphs for the two unconditional CPs. The horizontal axis is for the significance level from 0% to 100%, and the vertical axis is for the final percentage of errors. In each plot, ten calibration graphs of different colours correspond to labels $\{0, 1, \dots, 9\}$. As expected, these unconditional conformal predictions are not category-wise valid. In these plots, calibration graphs that are below the diagonal correspond to easy labels that have been predicted better (the number of errors for these

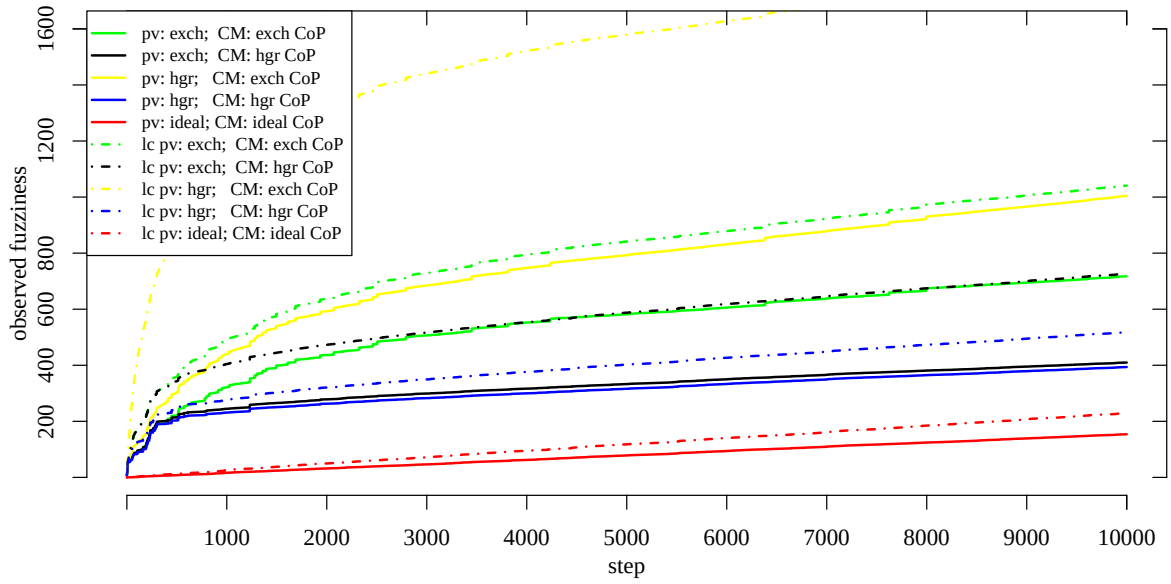

Figure 3.7.: Observed fuzziness for (unconditional) conformal prediction and label conditional (lc) conformal prediction. The results are for the LED data set of 10000 examples with 1% of noise. (In the legend "pv"(or "lc pv") stands for the model for calculating unconditional (or label conditional) p-values, "CM" stands for the model for constructing conformity measure.)

labels is less than expected), and calibration graphs above the diagonal are for difficult labels (the number of errors is greater than expected).

Figure 3.6 shows the results for the two label conditional CPs under hypergraphical models. These predictors are constructed in order to produce category-wise valid predictions, and the experiments just confirm that this property is true for these label conditional predictors.

Efficiency

Let us compare the efficiency of conformal prediction and label conditional conformal prediction under hypergraphical models. This is done by calculating the cumulative observed fuzziness OF_n (defined earlier on page 64) for these predictions.

Figure 3.7 [7] shows the cumulative observed fuzziness for on-line predictions. The solid lines are for unconditional predictors whose p-values are defined by (3.4); the dot-dashed lines show performance for label conditional predictors whose p-values are defined by (3.11). Again, two assumptions are considered: the exchangeability model and the nontrivial hypergraphical model; each model can be used for calculating the hypergraphical CoP conformity scores (3.6) or for p-values ((3.4) and (3.11)). These combinations give four unconditional CPs and four label conditional CPs. Also, as before, two idealised predictors are constructed: the unconditional *idealised predictor* is obtained using the true distribution for the data instead of the backward kernel P_{σ^*} in both (3.4) and (3.6), and analogously the *label conditional idealised predictor* is obtained using the true distribution in both (3.11) and (3.6).

The notation in the legend is similar to that in the previous set of experiments (see page 68), except that "lc" stands for "label conditional".

As expected, the price to pay for the category-wise validity of the label conditional CPs is that they are less efficient than the corresponding unconditional CPs. But the performance of the pure hypergraphical label conditional CP (the blue dot-dashed line) is almost as good as that for the corresponding unconditional one. The performance of other label conditional predictors (including the predictor corresponding to the black line, which was recommended in the unconditional setting of the previous section) is noticeably worse than that for the corresponding unconditional ones.

From the computational point of view, the label conditional version of computing p-values (3.11) is cheaper: for the label conditional p-values one only needs to look at the conformity scores of configurations with the same label, that is several times less (depending on the number of possible labels) than the whole set of configurations.

[7] `http://www.cs.rhul.ac.uk/~valentina/thesis_pics/of_hm_lc.pdf`

3.6. Conclusion

This chapter has discussed conformal prediction under hypergraphical models. Performance of several hypergraphical CPs has been studied as well as the performance of similar label conditional CPs.

The first finding of this chapter is that nontrivial hypergraphical models can be useful for conformal prediction when they are true. In the experiments with usual (unconditional) CPs these models only need to be used as soft models; the performance does not suffer much if the exchangeability model continues to be used as the hard model. This interesting phenomenon deserves a further theoretical investigation. First, this result can be empirically checked for other data sets. Second, it would be of interest to investigate it theoretically and in general describe the expected difference in the performance of CPs based on different hard and soft models.

The empirical study of label conditional CPs under hypergraphical models has demonstrated that they are essential for the category-wise validity. Finally, we have seen that the performance of label conditional CPs is close to that for unconditional ones if the hypergraphical models are used as both the hard and soft models.

Chapter 4.

On-line testing

As pointed out in the introductory chapter, it is traditional in machine learning to impose certain assumptions on data in order to state and prove useful properties of learning algorithms. Therefore it is important to test these assumptions in applications, otherwise results of the algorithms may be misleading. This chapter particulary focuses on testing the exchangeability assumption, although this approach to testing can be used for a more general assumption that the data-generating distribution agrees with an on-line compression model (such an example will be studied in Chapter 5). The main tools for testing exchangeability are exchangeability martingales. Section 4.1 summarises previous works that used exchangeability martingales and motivates this study. Section 4.2 formally defines exchangeability martingales and explains the theory behind their construction. Section 4.3 outlines established techniques for constructing exchangeability martingales and proposes two new types of martingales: mixtures of stepped martingales and plug-in martingales. Also it is proved that, under a stationarity assumption, the plug-in martingales are competitive with the power martingales that have been used in previous studies. Section 4.4 presents testing exchangeability for two benchmark data sets. Section 4.5 summarises this chapter.

4.1. Introduction

It follows from Theorem 1 on page 26 (and its generalisation Theorem 2 on page 31) that under certain assumptions on the data-generating distribution conformal predictors (CPs) achieve validity automatically. Therefore when applying the algorithms to real data it is important to check that these assumptions hold for the data, otherwise the validity of the algorithms may be violated. As mentioned in Section 1.2, a usual assumption for conformal prediction is exchangeability of data.

Exchangeability and i.i.d. It is important to note that the assumption that data is independent and identically distributed (i.i.d.) has the same meaning for testing as the exchangeability assumption. A joint distribution of a sequence of examples is exchangeable if it is invariant w.r. to any permutation of examples. Hence, if the data is i.i.d., its distribution is exchangeable. On the other hand, assuming that examples are from a Borel space (which is a weak assumption and correct for practical settings), by de Finetti's theorem [see, e.g., Schervish, 1995, p. 28] any exchangeable distribution on the data (a potentially infinite sequence of examples) is a mixture of distributions under which the data is i.i.d. Therefore, testing for exchangeability is equivalent to testing for being i.i.d.

In Statistics several distribution-free tests are designed to test the hypothesis that data is i.i.d. Often in these tests data is split into two samples and an attempt is made to detect differences between them. For example, one can compare the medians of the samples, or the distance between the cumulative distribution functions of the samples. And when the differences are above a certain threshold, the hypothesis that data is i.i.d. is rejected. (See, for example, [Cox and Hinkley, 1974, Chapter 6].) Unfortunately, such batch two-sample methods are not suitable for the purpose of this chapter. In the current context, the goal is to test the assumption for a single sequence of data, and it is

not clear how to split the data into two samples in order to obtain the most significant result.

This chapter discusses methods for on-line testing exchangeability. The first procedure of testing exchangeability on-line is described in Vovk et al. [2003]. The core testing mechanism is an exchangeability martingale. Exchangeability martingales are constructed using a sequence of p-values. The algorithm for generating p-values assigns small p-values to unusual examples. It implies the idea of designing martingales that would have a large value if too many small p-values were generated, and suggests corresponding power martingales. Other martingales (simple mixture and sleepy jumper) implement more complicated strategies, but follow the same idea of scoring on small p-values.

Ho [2005] applies power martingales to the problem of change detection in time-varying data streams. The author shows that small p-values inflate the martingale values and suggests to use the martingale difference as another test for the problem.

To the best of my knowledge, no study has aimed to find any other ways of translating p-values into a martingale value. This chapter describes two new more flexible methods of constructing exchangeability martingales for a given sequence of p-values.

4.2. Exchangeability martingales

This section outlines necessary definitions and results of previous studies.

4.2.1. Martingales for testing

The main tool for testing exchangeability on-line suggested by Vovk et al. [2003] is a martingale. The value of the martingale reflects the strength of evidence against the exchangeability assumption. An *exchangeability martingale* is a sequence of non-negative

random variables $S_0, S_1, \ldots$ that keep the conditional expectation:

$$S_n \geq 0$$
$$S_n = \mathbb{E}(S_{n+1} \mid S_1, \ldots, S_n),$$

where $\mathbb{E}$ refers to the expected value with respect to any exchangeable distribution on examples. It is also assumed that $S_0 = 1$. Note that, under a weak assumption that examples are from a Borel space, we will obtain an equivalent definition if we replace "any exchangeable distribution on examples" by "any distribution under which the examples are i.i.d." (remember the discussion of de Finetti's theorem in Section 4.1).

To understand the idea behind testing with martingale, we can imagine a coin tossing game. A gambler places bets on heads or tails: he does not bet more than the current capital and never risks bankruptcy. The gambler would like to test the coin on fairness. One way to do this is to place bets according to some strategy against the fairness of the coin. Suppose he decides to bet on heads. If the chosen strategy leads to a large growth of his initial capital then either the coin is biased or the coin is fair, but a freak coincidence has taken place. In this game, the strategy of the gambler can be thought of as a martingale and its value reflects the acquired capital. According to Ville's inequality (see Ville [1939], p. 100)

$$\mathbb{P}\{\exists n : S_n \geq C\} \leq 1/C, \quad \forall C > 0. \tag{4.1}$$

In other words, it is unlikely for any S_n to have a large value. For the problem of testing exchangeability, if the final value M of a martingale is large (20 and 100 are convenient rules of thumb), then the exchangeability assumption for the data can be rejected at any significance level $\delta > 1/M$.

4.2.2. On-line calculation of p-values

This chapter follows the notation of Section 1.2 and focuses on the classification problem, i.e., $\mathbf{Y}$ is finite. Consider a sequence of examples $(z_1, z_2, \ldots)$ that is assumed to be drawn from an exchangeable distribution. The goal is to test whether the assumption is correct. The first part of the on-line testing is generating a sequence of p-values for the given sequence of examples. Conformal transducers are used to generate the sequence of p-values. The process was briefly discussed in Section 1.2 (see page 26), and for the purpose of this chapter it will be described explicitly. The 1-Nearest Neighbour (1NN) algorithm is used as the underlying method to compute the conformity scores. The algorithm is simple but it works well enough in many cases (see, e.g., Hastie et al. [2013], pp. 463–475). A natural way to define the conformity score of an example is by comparing its distance to the examples with the same label to its distance to the examples with a different label:

$$\alpha_i = A\Big(\lbag z_1, \ldots, z_i \rbag, z_j\Big) := \frac{\min_{j \neq i : y_i \neq y_j} d(x_i, x_j)}{\min_{j \neq i : y_i = y_j} d(x_i, x_j)}, \tag{4.2}$$

where $d(x_i, x_j)$ is the Euclidean distance (it is the KNN-ratio conformity measure defined on page 51 for $K = 1$). According to the chosen conformity measure, α_i is low if the example is close to another example with a different label and far from any examples with the same label. Using the calculated conformity scores of all observed examples, the p-value p_n corresponding to the last example is calculate by (1.2) (on page 25). To make it more precise, Protocol 4.2 summarises the process of on-line calculation of p-values (it is clear that it can also be applied to a finite sequence $(z_1, \ldots, z_n)$ producing a finite sequence $(p_1, \ldots, p_n)$ of p-values).

Theorem 1 (page 26) states that examples generated by an exchangeable distribution provide independent and uniformly distributed p-values. This allows us to test exchangeability by calculating martingales as functions of the p-values.

Protocol 4.2 Generating p-values on-line

Input: $(z_1, z_2, \ldots)$ sequence of examples
$(\tau_1, \tau_2, \ldots)$ sequence of independent random numbers in $[0, 1]$
Output: $(p_1, p_2, \ldots)$ sequence of p-values
for $i = 1, 2, \ldots$ **do**
observe a new example z_i
for $j = 1$ **to** i **do**
$\alpha_j := A\Big(\{z_1, \ldots, z_i\}, z_j\Big)$
end for
$p_i := \frac{|\{j:\alpha_j<\alpha_i\}|+\tau_i|\{j:\alpha_j=\alpha_i\}|}{i}$
end for

4.3. Martingales based on p-values

This section focuses on the second part of on-line testing: given a sequence of p-values a martingale is calculated as a function of the p-values.

For each $i \in \{1, 2, \ldots\}$, let $f_i : [0, 1]^i \to [0, \infty)$. Let $(p_1, p_2, \ldots)$ be the sequence of p-values generated by Protocol 4.2. We consider martingales S_n of the form

$$S_n = \prod_{i=1}^{n} f_i(p_i), \quad n = 1, 2, \ldots, \tag{4.3}$$

where we denote $f_i(p) = f_i(p_1, \ldots, p_{i-1}, p)$ and call the function $f_i(p)$ a *betting function*. To be sure that (4.3) is indeed a martingale the following constraint on the betting functions f_i are needed:

$$\int_0^1 f_i(p)\,\mathrm{d}p = 1, \quad i = 1, 2, \ldots$$

Then we can check:

$$\mathbb{E}(S_{n+1} \mid S_0, \ldots, S_n) = \int_0^1 \prod_{i=1}^{n} \Big(f_i(p_i)\Big) f_{n+1}(p)\,\mathrm{d}p = \prod_{i=1}^{n} \Big(f_i(p_i)\Big) \int_0^1 f_{n+1}(p)\,\mathrm{d}p = \prod_{i=1}^{n} f_i(p_i) = S_n.$$

Note that if for any p-value $p \in [0, 1]$ we have $f_i(p) = 0$ then the martingale can

become zero and will never change after that. Therefore, it is reasonable to consider positive $f_i(p)$.

Using representation (4.3) one can update the martingale on-line: having calculated the p-value p_i for a new example in Protocol 4.2 the current martingale value becomes $S_i = S_{i-1} \cdot f_i(p_i)$. To complete the definition of martingales 4.3, next the betting functions f_i are described.

4.3.1. Previous results: power martingales and their mixtures

Previous studies have proposed to use a power betting function

$$\forall i: \ f_i(p) = \varepsilon p^{\varepsilon - 1},$$

where $\varepsilon \in [0, 1]$. Several martingales were constructed based on this function. The *power martingale* for some ε, denoted as M_n^ε, uses the fixed betting function and defined as

$$M_n^\varepsilon = \prod_{i=1}^{n} \varepsilon p_i^{\varepsilon - 1}.$$

The *simple mixture* martingale, denoted as M_n, is the mixture of power martingales over different $\varepsilon \in [0, 1]$:

$$M_n = \int_0^1 M_n^\varepsilon d\varepsilon.$$

Another martingale used in previous works is the sleepy jumper martingale. To construct this martingale generalised power martingales are defined by

$$M_n^{(\varepsilon)} = \prod_{i=1}^{n} \varepsilon_i p_i^{(\varepsilon_i - 1)},$$

where $(\varepsilon) = (\varepsilon_1, \ldots, \varepsilon_n)$ and all $\varepsilon_i \in [0, 1]$. The *sleepy jumper* martingale a mixture of the generalised power martingales with respect to the distribution called the sleepy

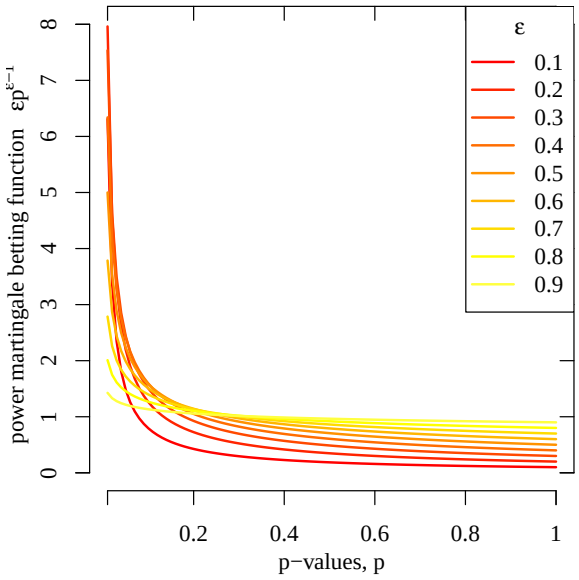

Figure 4.1.: The betting functions that are used to construct the power, simple mixture and sleepy jumper martingales.

jumper distribution (see p. 176 [Vovk et al., 2005a]). Intuitively, the martingale tries to track the best power martingale by switching between different ε.

Since all the previous martingales were constructed using the family of power betting functions, such a martingale will grow only if the sequence of p-values contains many small p-values. This follows from the shape of the power betting functions: see Figure 4.1. If the generated p-values concentrate in any other part of the unit interval, we cannot expect the martingale to grow. This is, however, a situation where lack of exchangeability makes the p-values cluster around 1: we observe examples that are ideal shapes of several kinds distorted by random noise, and the amount of noise decreases with time. Predicting the kind of a new example using the conformity measure (4.2) will then tend to produce large p-values. Therefore the main weak point of the martingales introduced before is that they are based on the family of power betting functions. Next two new martingales are described, and the main goal in their construction is to avoid any assumptions about the mechanism of generating p-values.

4.3.2. Mixtures of stepped martingales

We consider the second part of on-line testing: given a sequence of p-values, the goal is to test it for uniformity in $[0,1]$. Martingales used in this chapter are defined by their betting functions. Betting functions from the power family (introduced before) score only on low p-values, and therefore any other deviation from uniformity in p-values cannot be detected. If the distribution of p-values is not uniform, the p-values are concentrated in certain parts of the unit interval; then one could construct a betting function that scores on those values. But again the martingale based on this betting function could not detect other violations. To find a martingale that can grow for any kind of deviations from uniformity, it would be natural to consider the family of all positive continuous betting functions (each of them scores on a certain violation) and, as any type of violations is allowed, construct mixtures of martingales corresponding to this family. But it is not clear if this mixture can be calculated efficiently. As a feasible approximation of this idea, this subsection introduces families of positive stepped betting functions and then martingales based on them. Such a family of betting functions is an approximation for the family of positive continuous betting functions; each of the betting functions scores on a different deviation from uniformity in a sequence of p-values. And it is shown that mixtures of such martingales can be calculated efficiently.

Consider the family of positive stepped betting functions with k breakpoints:

$$\begin{aligned} f(p) := &\begin{cases} 0, & p = 0 \\ d_i, & \frac{i-1}{k} < p \le \frac{i}{k}, i = 1, \dots, k \end{cases}, \\ &\frac{1}{k}\sum_{i=1}^{k} d_i = 1, \\ &d_i > 0, i = 1 \dots k. \end{aligned} \tag{4.4}$$

(These are functions of the type $[0,1] \to [0,\infty)$.)

Denote $\rho_i := \frac{d_i}{k}, i = 1, \dots, k$, and let $E := \left\{\rho = (\rho_1, \dots, \rho_k) \in (0; \infty)^k : \sum_{i=1}^k \rho_i = 1\right\}$. Then each positive stepped betting function corresponds to a sequence in E.

Let us rewrite martingale (4.3) for a positive stepped betting function $f(p)$. For a sequence of p-values $(p_1, \dots, p_n)$ define the vector of k counts $m := (m_1, \dots, m_k)$, where

$$m_i := \left|\left\{p_j, j = 1 \dots, n : \frac{i-1}{k} < p_j \le \frac{i}{k}\right\}\right|, \quad i = 1, \dots, k, \tag{4.5}$$

is the number of p-values in the sequence that belong to $\left(\frac{i-1}{k}; \frac{i}{k}\right]$. Then the *stepped martingale* $S_n^{(\rho)}$ corresponding to a stepped betting function defined by $\rho = (\rho_1, \dots, \rho_k)$ can be written as

$$S_n^{(\rho)} = \prod_{j=1}^n f(p_j) = \prod_{i=1}^k (d_i)^{m_i} = \prod_{i=1}^k (k\rho_i)^{m_i} = k^{\sum_{i=1}^k m_i} \prod_{i=1}^k \rho_i^{m_i} = k^n \prod_{i=1}^k \rho_i^{m_i}. \tag{4.6}$$

To eliminate the dependency on ρ let us define a measure on E and then construct mixtures of martingales $S_n^{(\rho)}$ using the measure. The *Dirichlet measure* on E is defined by

$$\mu_b(\rho_1, \dots, \rho_k) := \frac{\rho_1^{b_1 - 1} \cdots \rho_k^{b_k - 1}}{B(b_1, \dots, b_k)},$$

where $b := (b_1, \dots, b_k)$ is a vector of parameters and

$$B(b_1, \dots, b_k) := \frac{\Gamma(b_1) \cdots \Gamma(b_k)}{\Gamma(b_1 + \cdots + b_k)}$$

is the beta function of k variables.

Let us calculate the mixture of stepped martingales $S_n^{(\rho)}$ with respect to the Dirichlet

measure:

$$
\begin{aligned}
U_n^b := \int_E S_n^{(\rho)} \,\mathrm{d}\mu_b(\rho) = \int_E k^n \cdot \prod_{i=1}^k \rho_i^{m_i} \cdot \frac{\prod_{i=1}^k \rho_i^{b_i-1}}{B(b_1,\ldots,b_k)} \,\mathrm{d}\rho = \\
\int_E k^n \cdot \frac{B(b_1+m_1,\ldots,b_k+m_k)}{B(b_1,\ldots,b_k)} \cdot \frac{\prod_{i=1}^k \rho_i^{b_i+m_i-1}}{B(b_1+m_1,\ldots,b_k+m_k)} \,\mathrm{d}\rho = \\
\int_E k^n \cdot \frac{B(b_1+m_1,\ldots,b_k+m_k)}{B(b_1,\ldots,b_k)} \cdot \,\mathrm{d}\mu_{b+m}(\rho) = \\
k^n \cdot \frac{B(b_1+m_1,\ldots,b_k+m_k)}{B(b_1,\ldots,b_k)} = k^n \cdot \frac{\prod_{i=1}^k \Gamma(b_i+m_i)}{\Gamma(\sum_{i=1}^k b_i+n)} \cdot \frac{\Gamma(\sum_{i=1}^k b_i)}{\prod_{i=1}^k \Gamma(b_i)} = \\
k^n \cdot \prod_{i=1}^k \frac{\Gamma(b_i+m_i)}{\Gamma(b_i)} \cdot \frac{\Gamma(\sum_{i=1}^k b_i)}{\Gamma(\sum_{i=1}^k b_i+n)}.
\end{aligned}
\tag{4.7}
$$

Using the fact that $\Gamma(z+t) = z \cdot (z+1) \cdots (z+t-1) \cdot \Gamma(z)$ we have

$$
U_n^b = k^n \cdot \prod_{\substack{i=1,\ldots k \\ m_i>0}} \left[b_i(b_i+1)\cdots(b_i+m_i-1) \right] / \prod_{j=0}^{n-1} \left(\sum_{i=1}^k b_i + j \right).
$$

For each vector of parameters $b = (b_1, \ldots, b_k)$ we can calculate the corresponding martingale U_n^b. The Dirichlet measure corresponding to $b : b_i := 1, i = 1, \ldots, k$, is equivalent to the uniform distribution over E. For our purposes it would be natural to choose the uniform distribution over betting functions and in this case U_n^b can be rewritten as

$$
U_n^1 = k^n \cdot \prod_{i=1,\ldots k} m_i! / \prod_{j=0}^{n-1} (k+j)\,, \tag{4.8}
$$

where m_i are given by (4.5). Such a martingale will be called the *mixture of stepped martingales*.

The martingale parameter k is the number of breakpoints for stepped betting functions (4.4). Intuition behind the choice of k is similar to choosing the number of bins for the histogram of data. There are various useful guidelines for choosing the number of bins, for example, to set $k \propto n^{1/3}$ (see, e.g., Devroye and Györfi [1985, pp. 97–100]).

Another way of looking at the mixture of stepped martingales is the Bayesian point of view. The Bayesian approach in the present context involves defining a family of betting functions, choosing a prior distribution on the betting functions, and integrating the martingales corresponding to these betting functions with respect to the prior distribution. In our case, the prior distribution over the family of stepped betting functions is the Dirichlet distribution defined by measure μ_b. This is a popular choice of prior distribution which makes the calculation of the posterior distribution easy (see, e.g., [Cox and Hinkley, 1974, pp. 371–372]). Initially, the prior is the uniform distribution, which corresponds to the vector of parameters $b : b_i := 1, i = 1, \dots, k$. Once a p-value arrives the posterior distribution can be found by updating the vector of parameters to $b + m$, where m is the vector of counts (4.5). (Intuitively, the prior is shifted towards betting functions that provide better growth for the current sequence of p-values.) The posterior distribution then is used as the basis for calculating the mixture of stepped martingales. In our calculations these steps happen implicitly, but the result can be seen in the third line in equation (4.7): the mixture is calculated using the Dirichlet measure μ_{b+m}.

The performance of martingale (4.8) on two benchmark data sets will be presented in Section 4.4.

4.3.3. Plug-in martingales

Stepped martingales discussed in the previous subsection used the same betting function at each step, which is the reason why each single martingale is able to detect only certain deviations from uniformity. Instead of mixing such martingales with fixed betting functions, this subsection follows another approach – the plug-in approach. The martingale constructed below uses different betting functions for each step. The main idea is to use past p-values to learn the type of deviation from uniformity, and then adjust the betting

function to grow for this deviation.

Construction of plug-in martingales

Let us use an estimated probability density function as the betting function $f_i(p)$. At each step the probability density function is estimated using the accumulated p-values:

$$\rho_i(p) = \widehat{\rho}(p_1, \ldots, p_{i-1}, p), \tag{4.9}$$

where $\widehat{\rho}(p_1, \ldots, p_{i-1}, p)$ is the estimate of the probability density function using the p-values $p_1, \ldots, p_{i-1}$ output by Protocol 4.2.

Substituting these betting functions into (4.3) we get a new martingale that is called a *plug-in martingale*. The martingale avoids betting if the p-values are distributed uniformly, but if there is any peak it will be used for betting.

Estimating a probability density function. For the experiments presented in this chapter the statistical environment and language R was used. The `density` function in its `Stats` package implements kernel density estimation with different parameters. But since p-values always lie in the unit interval, the standard methods of kernel density estimation lead to poor results for the points that are near the boundary. To get better results for the boundary points the sequence of p-values is reflected to the left from zero and to the right from one. Then the kernel density estimate is calculated using the extended sample $\cup_{i=1}^{n}\{-p_i, p_i, 2 - p_i\}$. The estimated density function is set to zero outside the unit interval and then normalised to integrate to one. For the results presented in this chapter the parameters used are the Gaussian kernel and Silverman's "rule of thumb" for bandwidth selection. Other settings have been tried as well, but the results are comparable and lead to the same conclusions.

The values S_n of the plug-in martingale can be updated recursively. Suppose comput-

ing the conformity scores $(\alpha_1, \ldots, \alpha_n)$ from $(z_1, \ldots, z_n)$ takes time $g(n)$ and evaluating (4.9) takes time $h(n)$. Then updating S_{n-1} to S_n takes time $O(g(n)+n+h(n))$: indeed, it is easy to see that calculating the rank of α_n in the multiset $\{\alpha_1, \ldots, \alpha_n\}$ takes time $\Theta(n)$.

The performance of the plug-in martingale on benchmark data sets will be presented in Section 4.4. The rest of the current section proves that the plug-in martingale provides asymptotically a better growth rate than any martingale with a fixed betting function. To prove this asymptotical property of the plug-in martingale the following assumptions are needed.

Assumptions

Consider an infinite sequence of p-values $(p_1, p_2, \ldots)$. (This is simply a deterministic sequence.) For its finite prefix $(p_1, \ldots, p_n)$ define the corresponding empirical probability measure $\mathbf{P}_n$: for a Borel set A in $\mathbb{R}$,

$$\mathbf{P}_n(A) = \frac{|\{i = 1, \ldots, n : p_i \in A\}|}{n}.$$

We say that the sequence $(p_1, p_2, \ldots)$ is *stable* if there exists a probability measure $\mathbf{P}$ on $\mathbb{R}$ such that:

1. $\mathbf{P}_n \xrightarrow[n\to\infty]{\text{weak}} \mathbf{P}$;
2. there exists a positive continuous density function $\rho(p)$ for $\mathbf{P}$: for any Borel set A in $\mathbb{R}$, $\mathbf{P}(A) = \int_A \rho(p)dp$.

Intuitively, the stability means that asymptotically the sequence of p-values can be described well by a probability distribution.

Consider a sequence $(f_1(p), f_2(p), \ldots)$ of betting functions. (This is simply a deterministic sequence of functions $f_i : [0,1] \to [0,\infty)$, although we are particularly interested in

the functions $f_i(p) = \rho_i(p)$, as defined in (4.9).) We say that this sequence is *consistent for* $(p_1, p_2, \ldots)$ if

$$\log\bigl(f_n(p)\bigr) \xrightarrow[n\to\infty]{\text{uniformly in } p} \log\bigl(\rho(p)\bigr).$$

Intuitively, consistency is an assumption about the algorithm that is use to estimate the function $\rho(p)$; in the limit it is desirable to get a good approximation.

Growth rate of plug-in martingale

The following result says that, under the assumptions described above, the logarithmic growth rate of the plug-in martingale is better than that of any martingale with a fixed betting function (remember that by a betting function is implied any function mapping $[0, 1]$ to $[0, \infty)$).

Theorem 7. *If a sequence* $(p_1, p_2, \ldots) \in [0, 1]^\infty$ *is stable and a sequence of betting functions* $\bigl(f_1(p), f_2(p), \ldots\bigr)$ *is consistent for it then, for any positive continuous betting function* f*,*

$$\liminf_{n\to\infty} \left(\frac{1}{n} \sum_{i=1}^{n} \log\bigl(f_i(p_i)\bigr) - \frac{1}{n} \sum_{i=1}^{n} \log\bigl(f(p_i)\bigr) \right) \geq 0.$$

First the meaning of Theorem 7 is explained and then it is proved. According to representation (4.3) after n steps the martingale grows to

$$\prod_{i=1}^{n} f_i(p_i). \tag{4.10}$$

As mentioned earlier (see page 87), it is reasonable to restrict considerations to $f_i(p) > 0$. Then we can rewrite product (4.10) as sum of logarithms, which gives us the logarithmic growth of the martingale:

$$\sum_{i=1}^{n} \log\Bigl(f_i(p_i)\Bigr).$$

We assume that the sequence of p-values is stable and the sequence of estimated prob-

ability density functions that is used to construct the plug-in martingale is consistent. Then the limit inequality in Theorem 7 states that the logarithmic growth rate of the plug-in martingale is asymptotically at least as high as that of any martingale with a fixed betting function (which have been suggested in previous studies).

To prove Theorem 7 the following lemma will be used.

Lemma 1. *For any probability density functions ρ and f (so that $\int_0^1 \rho(p)dp = 1$ and $\int_0^1 f(p)dp = 1$),*

$$\int_0^1 \log\Big(\rho(p)\Big)\rho(p)dp \geq \int_0^1 \log\Big(f(p)\Big)\rho(p)dp.$$

Proof of Lemma 1. It is well known [Kullback, 1959, p. 14] that the Kullback–Leibler divergence is always non-negative:

$$\int_0^1 \log\Big(\frac{\rho(p)}{f(p)}\Big)\rho(p)dp \geq 0.$$

This is equivalent to the inequality asserted by Lemma 1. □

Proof of Theorem 7. Suppose that, contrary to the statement of Theorem 7, there exists $\delta > 0$ such that

$$\liminf_{n\to\infty}\left(\frac{1}{n}\sum_{i=1}^{n}\log\big(f_i(p_i)\big) - \frac{1}{n}\sum_{i=1}^{n}\log\big(f(p_i)\big)\right) < -\delta. \tag{4.11}$$

Then choose an ϵ satisfying $0 < \epsilon < \delta/4$.

Substituting the definition of $\rho(p)$ into Lemma 1 we obtain

$$\int_0^1 \log\Big(\rho(p)\Big)d\mathbf{P} \geq \int_0^1 \log\Big(f(p)\Big)d\mathbf{P}. \tag{4.12}$$

From the stability of $(p_1, p_2, \ldots)$ it follows that there exists a number $N_1 = N_1(\epsilon)$ such

that, for all $n > N_1$,

$$\left| \int_0^1 \log\Big(f(p)\Big) d\mathbf{P}_n - \int_0^1 \log\Big(f(p)\Big) d\mathbf{P} \right| < \epsilon$$

and

$$\left| \int_0^1 \log\Big(\rho(p)\Big) d\mathbf{P}_n - \int_0^1 \log\Big(\rho(p)\Big) d\mathbf{P} \right| < \epsilon.$$

Then inequality (4.12) implies that, for all $n \geq N_1$,

$$\int_0^1 \log\Big(\rho(p)\Big) d\mathbf{P}_n \geq \int_0^1 \log\Big(f(p)\Big) d\mathbf{P}_n - 2\epsilon.$$

By the definition of the probability measure $\mathbf{P}_n$, the last inequality is the same thing as

$$\frac{1}{n} \sum_{i=1}^{n} \log\Big(\rho(p_i)\Big) \geq \frac{1}{n} \sum_{i=1}^{n} \log\Big(f(p_i)\Big) - 2\epsilon. \tag{4.13}$$

By the consistency of $\big(f_1(p), f_2(p), \ldots\big)$ there exists a number $N_2 = N_2(\epsilon)$ such that, for all $i > N_2$ and all $p \in [0, 1]$,

$$\Big|\log\big(f_i(p)\big) - \log\big(\rho(p)\big)\Big| < \epsilon. \tag{4.14}$$

Let us define the number

$$M = \max_{i,p} \Big|\log\big(f_i(p)\big) - \log\big(\rho(p)\big)\Big|. \tag{4.15}$$

From (4.14) and (4.15) we have

$$\Big|\log\big(f_i(p)\big) - \log\big(\rho(p)\big)\Big| \leq \begin{cases} M, & i \leq N_2 \\ \epsilon, & i > N_2. \end{cases} \tag{4.16}$$

Denote $N_3 = \max(N_1, N_2)$. Then, using (4.16) and (4.13), we obtain, for all $n > N_3$,

$$\frac{1}{n}\sum_{i=1}^{n}\log\Big(f_i(p_i)\Big) \geq \frac{1}{n}\sum_{i=1}^{n}\log\Big(f(p_i)\Big) - 3\epsilon - \frac{MN_3}{n}.$$

Denoting $N_4 = \max(N_3, \frac{MN_3}{\epsilon})$, we can rewrite the last inequality as

$$\frac{1}{n}\sum_{i=1}^{n}\log\Big(f_i(p_i)\Big) \geq \frac{1}{n}\sum_{i=1}^{n}\log\Big(f(p_i)\Big) - 4\epsilon,$$

for all $n > N_4$. Finally, recalling that $\epsilon < \frac{\delta}{4}$, we have, for all $n > N_4$,

$$\frac{1}{n}\sum_{i=1}^{n}\log\Big(f_i(p_i)\Big) - \frac{1}{n}\sum_{i=1}^{n}\log\Big(f(p_i)\Big) \geq -\delta.$$

This contradicts (4.11) and therefore completes the proof of Theorem 7. □

4.4. Empirical study

This section investigates the performance of mixtures of stepped martingales and plug-in martingales, and compares them with that of the simple mixture martingale. Two benchmark data sets have been tested for exchangeability: the USPS data set and the Statlog Satellite data set. (Presented results are for the seed 0 of the R pseudorandom number generator: the seed was set to 0 to generate the sequence $(\tau_1, \tau_2, \ldots)$ for calculating p-values, and also the seed was set to 0 before generating the permutation for shuffling data. Similar results were observed in experiments with other random seeds.)

USPS data set

For the first set of experiments the benchmark USPS data set (see Appendix A.1) is used. Testing is performed for the whole data set of 9298 examples. It is well known

that the examples in this data set are not perfectly exchangeable [Vovk et al., 2003], and any reasonable test should reject exchangeability there.

The top plot in Figure 4.2 shows the typical performance of the martingales when the exchangeability assumption is satisfied for sure: all examples have been randomly shuffled before the testing. The bottom plot in Figure 4.2 shows the performance of the martingales when the examples arrive in the original order: first 7291 of the training sequence and then 2007 of the test sequence. The p-values are generated on-line by Protocol 4.2 and three martingales are calculated from the same sequence of p-values. The final value for the simple mixture martingale is 6.9×10^9, the final value for the mixture of stepped martingales for $k = 3$ (explained below) is about 2.3×10^5 and the final value for the plug-in martingale is 1.4×10^8.

For the mixture of stepped martingales parameter $k = 3$ was chosen by looking at the corresponding final values for $1 \le k \le 30$. These final values are shown in the left plot of Figure 4.4. For this data set of 9298 examples the best performance of the mixture of stepped martingales is for $k = 3$.

The left plot in Figure 4.5 shows the betting functions that correspond to the plug-in martingale and the "best" power martingale. For the plug-in martingale, the function is the estimated probability density function calculated using the whole sequence of p-values. The betting function for the family of power martingales corresponds to the parameter ε^* that provides the largest final value among all power martingales. It explains why we could not see advantages of the new approaches for this data set: all the martingales grew up to approximately the same level. There is not much difference between the best betting functions for the old and new methods, and the new methods suffer because of their greater flexibility.

The results presented in Figure 4.2 can be also explained in terms of rejecting or not rejecting the hypothesis of exchangeability for the data. Let us find the maximal

martingale value for each of the martingales: for the simple mixture martingale it is 7.6×10^9, for the mixture of stepped martingales it is 4.9×10^5, and for the plug-in martingale it is 1.5×10^8. Using Ville's inequality (4.1) these values can be converted into significance levels for rejecting the hypothesis of exchangeability. It gives us the following conclusions – all three martingales reject exchangeability for the data: the simple mixture martingale at the significance level of 10^{-9}, the mixture of stepped martingales at the significance level of 10^{-5}, and the plug-in at the significance level of 10^{-8}.

Statlog Satellite data set

The second set of experiments uses the Statlog Satellite data set (described in Appendix A.3). Testing is preformed for the whole data set of 6435 examples.

The top plot in Figure 4.3 shows the performance of martingales for randomly shuffled examples of this data set. As expected, the martingales do not reject the exchangeability assumption there. The bottom plot in Figure 4.3 presents the performance of the martingales when the examples arrive in the original order. The final value for the simple mixture martingale is 3.0×10^2, the final value of the mixture of stepped martingales for $k = 10$ is 2.3×10^{30} and the final value for the plug-in martingale is 8.0×10^{16}. For the Statlog Satellite data set of 6435 examples the best performance of the mixture of stepped martingales is for $k = 10$ (see the right plot in Figure 4.4).

The corresponding betting functions for the plug-in martingale and the "best" power martingale are presented in the right plot in Figure 4.5. For this data set the generated p-values have a tricky distribution. The family of power betting functions $\varepsilon p^{\varepsilon-1}$ cannot provide a good approximation. The power martingales lose on p-values close to the second peak of the p-values distribution. But the two new martingales are more flexible and ended up with much higher final values.

Again, let us use Ville's inequality (4.1) to find the significance level for rejecting

exchangeability for the data. The maximal martingale values are following: for the simple mixture martingale it is 2.27×10^{11}, for the mixture of stepped martingales it is 1.82×10^{51}, and for the plug-in martingale it is 7.75×10^{20}. Therefore all three martingales reject exchangeability for the data: the simple mixture martingale at the significance level of 10^{-11}, the mixture of stepped martingales at the significance level of 10^{-51}, at the plug-in at the significance level of 10^{-20}.

It can be argued that the old method as well as the two new methods work for the Statlog Satellite data set in the sense of rejecting the exchangeability assumption at any of the commonly used thresholds (such as 20 or 100, corresponding to the significance levels of 0.05 and 0.01). However, the situation would have been different had the data set consisted of only the first 1000 examples: the final value of the simple mixture martingale would have been 0.013 whereas the final value of the mixture of stepped martingales for $k = 10$ would have been 1.3×10^{13} and the final value of the plug-in martingale would have been 4.6×10^{15}. Or, in terms of rejecting or not rejecting the hypothesis of exchangeability: the maximal value of the simple mixture martingale for the first 1000 examples is less than 1 and therefore the hypothesis can be accepted, whereas the two new martingales reject exchangeability at the significance level of 10^{-13}.

4.5. Discussions and conclusion

This chapter has discussed on-line testing exchangeability. The method is based on the calculation of exchangeability martingales. Previous studies followed the natural idea that lack of exchangeability leads to new examples looking strange as compared to the old ones and therefore to small p-values (for example, if the data-generating mechanism changes its regime and starts producing a different kind of examples). However, it is not always true for all kinds of deviations from exchangeability (see the example on page 89).

The goal of this chapter has been to find exchangeability martingales that does not

need any assumptions about the p-values generated by the method of conformal prediction. The studied martingales are defined by a betting function, that translates a p-value into a positive value. Two approaches – Bayesian and plug-in – have been considered in order to construct two new types of martingales. (It is generally believed that the Bayesian approach is more efficient than the plug-in approach [see, e.g., Bernardo and Smith, 2000, p. 483].)

Following the Bayesian approach, the mixtures of stepped martingales have been constructed. An efficient way of calculating these martingales has been described.

Following the plug-in approach, the plug-in martingale has been constructed. This martingale adapts to the unknown distribution of p-values by estimating a good betting function from the past data. It has been proved that for stable sequences of p-values the more adaptive plug-in martingale provides asymptotically the best result compared with any other martingale with a fixed betting function.

The performance of the two new martingales and the simple mixture martingale has been studied empirically for two benchmark data sets. For the first data set the performance of these three martingales is similar, but for the second data set the flexibility of the two new martingales becomes essential. The results of the experiments confirm that calculated from the same sequence of p-values the plug-in martingale extracts approximately the same amount or more information about the data-generating distribution as compared to the previously introduced power martingales. Secondly, we have seen that the performance of the mixture of stepped martingales depends on the choice of parameter k; for certain values of k this mixture of stepped martingales can grow to a much higher value than the plug-in martingale (see the final values for the Statlog Satellite data set in the original order). On the other hand, we have seen situations where the mixture of stepped martingales is less sensitive than the plug-in martingale (see testing for the USPS data set in the original order, or the first 500 steps of testing the Statlog

Satellite data set in the original order). Therefore both of the new martingales are useful for testing and deserve further investigation.

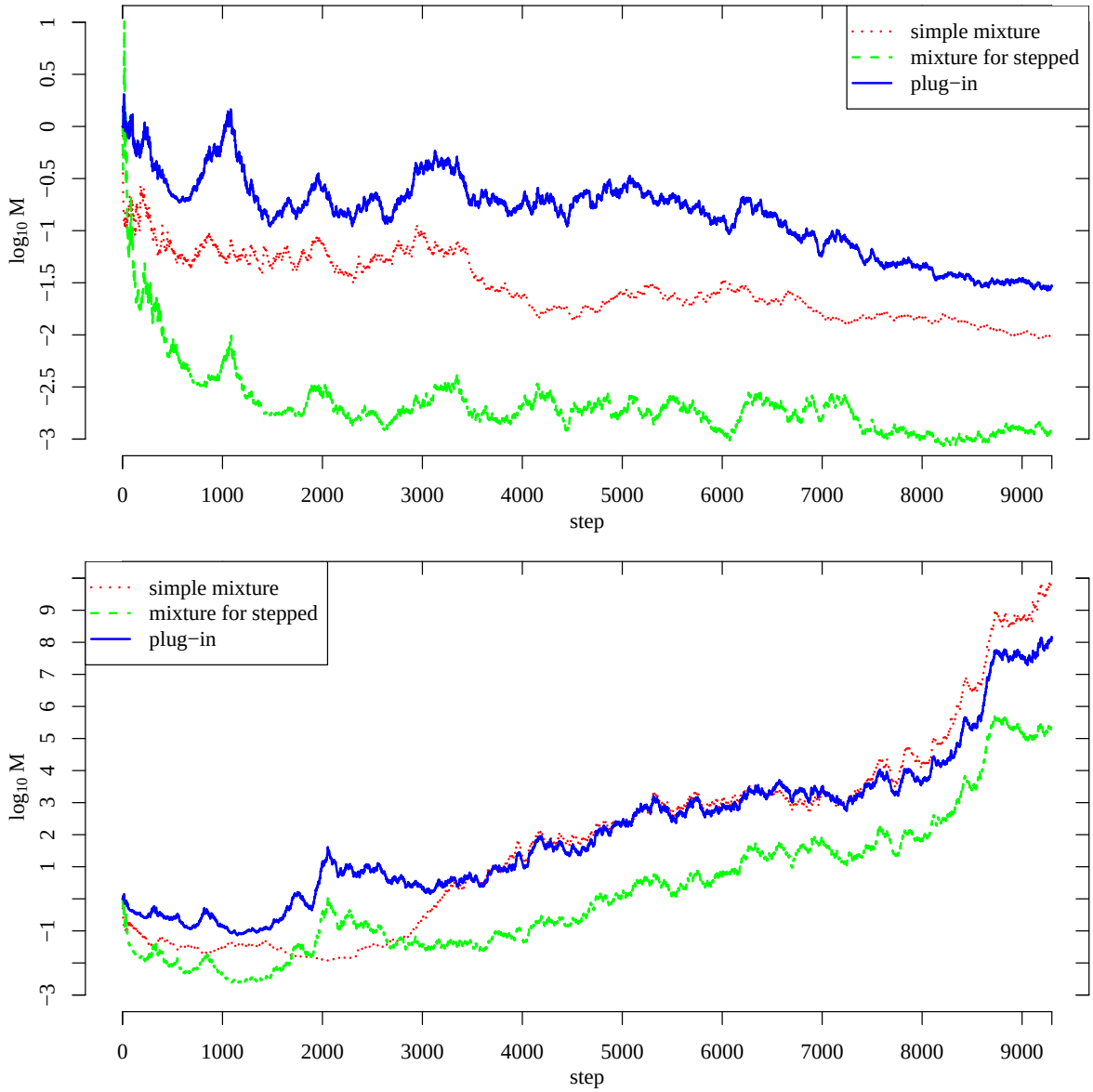

Figure 4.2.: The growth of the martingales for the USPS data set. Top plot: the data set randomly shuffled before on-line testing; the exchangeability assumption is satisfied (the final martingale values are lower than 0.03). Bottom plot: the data in the original order; the exchangeability assumption is rejected (the final martingale values are greater than 2.2×10^5).

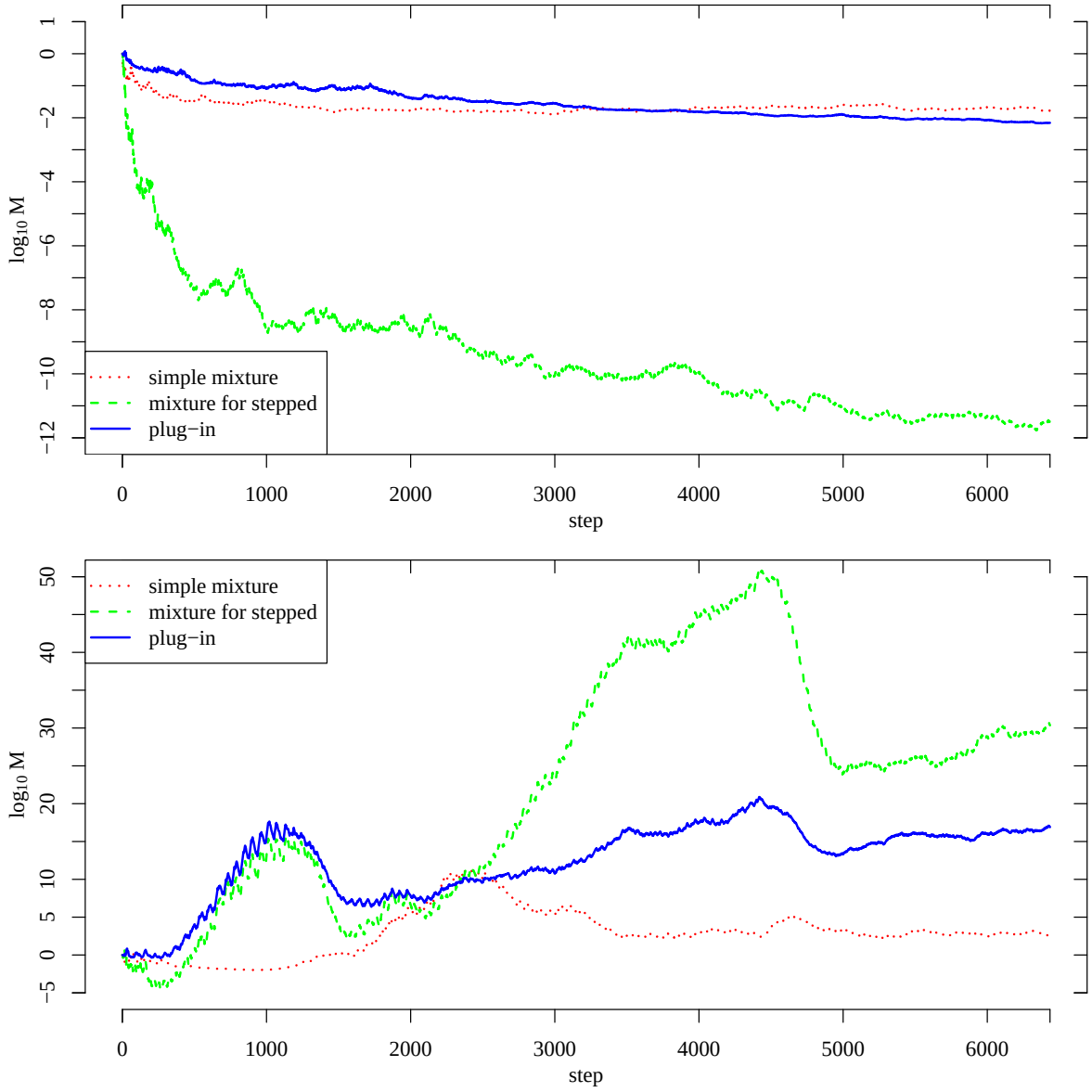

Figure 4.3.: The growth of the martingales for the Statlog Satellite data set. Top plot: the data set randomly shuffled before on-line testing; the exchangeability assumption is satisfied (the final martingale values are lower than 0.02). Bottom plot: data in the original order; the exchangeability assumption is rejected (the final value of the simple mixture martingale is 3.0×10^2, the final value of the mixture of stepped martingales (for $k = 10$) is about 2.3×10^{30}, and the final value of the plug-in martingale is 8.0×10^{16}).

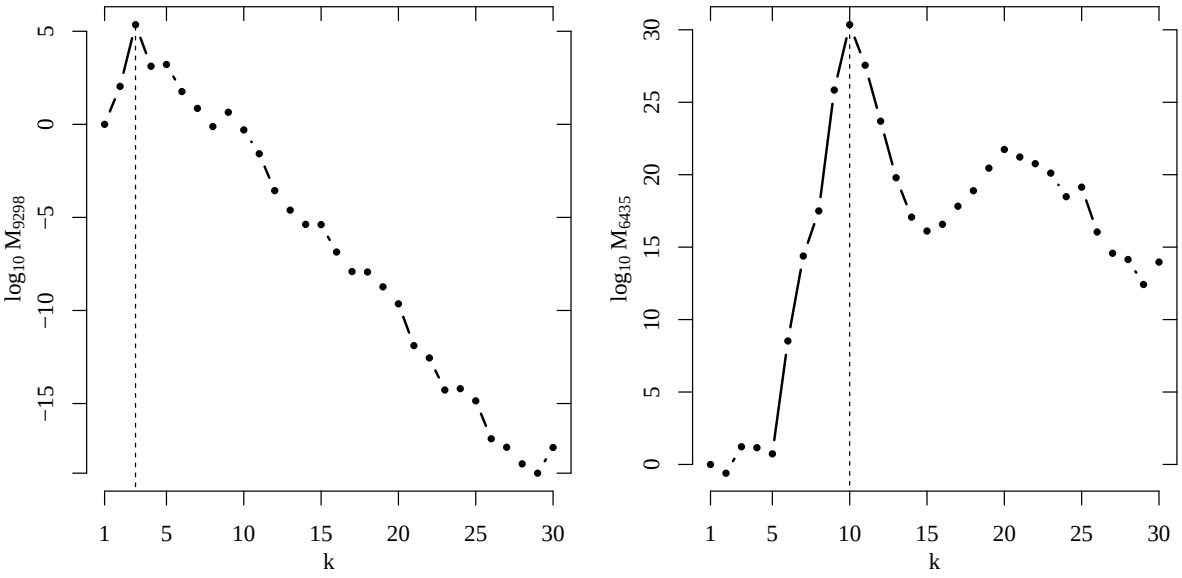

Figure 4.4.: Left plot: the final values of mixtures of stepped martingales for the USPS data set in the original order (for different values of parameter k); the best performance is for $k = 3$. Right plot: the same for the Statlog Satellite data set; the best performance is for $k = 10$.

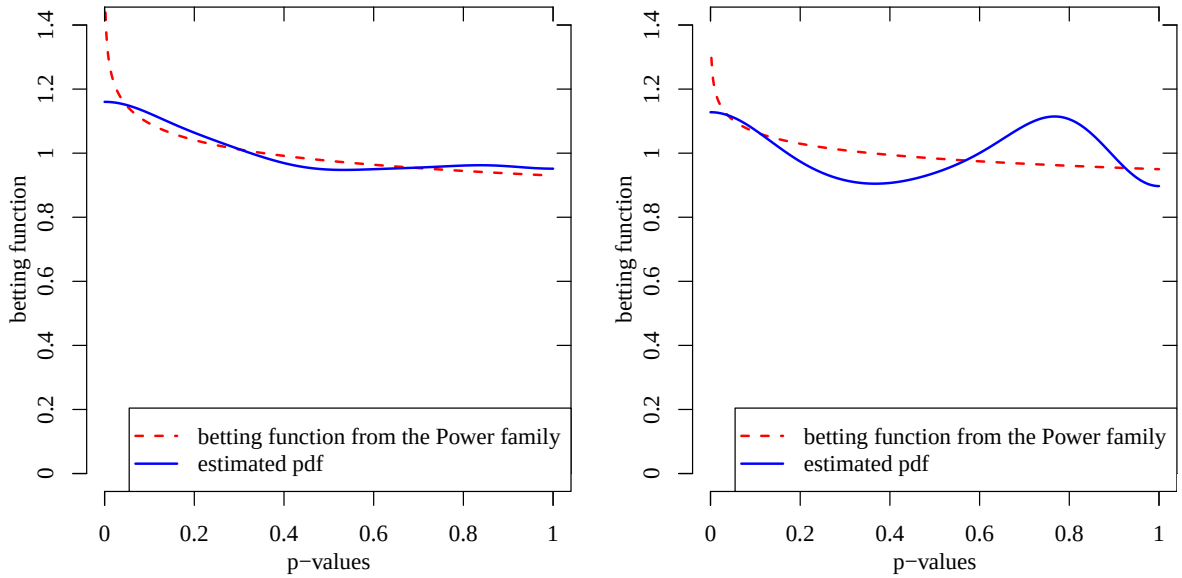

Figure 4.5.: Left plot: the betting functions for testing the USPS data set for examples in the original order. Right plot: the same for the Statlog Satellite data set.

Chapter 5.

Gauss linear assumption: testing and prediction

This chapter focuses on the Gauss linear assumption. More specifically, it applies on-line testing described in Chapter 4 to this assumption, and empirically studies the performance of both on-line testing and conformal prediction under this assumption. Section 5.1 outlines related results. Section 5.2 reiterates necessary terminology and describes the on-line compression model (OCM) for the Gauss linear assumption. Section 5.3 empirically studies conformal prediction under this OCM for synthetic data sets. Section 5.4 applies the idea of on-line testing to the Gauss linear assumption. Then the performance of testing is empirically studied for both synthetic data sets and the benchmark Abalone data set. In the case of the Abalone data set, the testing results are used to select a more realistic model for the data, and finally this model is used to obtain conformal prediction for the data. Section 5.5 summarises the chapter.

5.1. Introduction

Conformal prediction for the regression problem has been studied before, e.g.: [Vovk et al., 2005a, Section 2.3] describe efficient solutions for the regression problem under the exchangeability assumption; in [Vovk et al., 2005b] the regression problem is studied under several important models, including the Gauss linear model.

In [Lei and Wasserman, 2014] the regression problem is studied under the assumption that data is i.i.d. The authors combine the idea of conformal prediction with density estimation to construct distribution-free prediction sets and study their finite sample coverage guaranties.

This chapter empirically studies CPs under the Gauss linear assumption described in [Vovk et al., 2005b, Section 8.5]. In addition, this efficient algorithm for prediction is adapted for being used in on-line testing the Gauss linear assumption.

5.2. Definitions

As usual, this chapter follows the notation of Section 1.2; in particular, it considers the object space $\mathbf{X} := \mathbb{R}^K$ (objects are vectors of K real numbers) and the label space $\mathbf{Y} := \mathbb{R}$.

5.2.1. Gauss linear model

Denote $\mathbf{z}_n := (1, x_n)$, then the linear regression model has the form

$$y_n = \gamma \cdot \mathbf{z}_n + \xi_n, \tag{5.1}$$

where $\gamma = (\gamma_1, \ldots, \gamma_{K+1})$ is a vector of $K + 1$ unknown coefficients and $\xi_n, n = 1, 2, \ldots$, are random variables. For the *Gauss linear model* the random variables ξ_n are assumed

to be independent of each other and normally distributed with zero mean and variance $\sigma^2 > 0$ (the same for all the variables). (Note that there is no assumption on the distribution generating objects x_n.)

Protocol

This chapter discusses prediction in the on-line mode (Protocol 1.1 on page 24). In the case of regression under the Gauss linear assumption, a prediction set is an interval of values and the information efficiency of such a prediction naturally measured as the length of the interval (see [Vovk et al., 2005b] and [Lei and Wasserman, 2014]). To study properties of such predictions two variables will be calculated: the number of errors during the first n steps, denoted Err_n^ϵ, and the median length of prediction intervals after the first n steps, denoted L_n^ϵ. Protocol 5.3 summarises the calculations of the variables.

Protocol 5.3 On-line protocol for regression

$\mathrm{Err}_0^\epsilon := 0$ for all $\epsilon \in (0,1)$;
for $n = 1, 2, \ldots$ **do**
  Reality outputs $x_n \in \mathbb{R}^K$
  Predictor outputs $\Gamma_n^\epsilon \subseteq \mathbb{R}$ for all $\epsilon \in (0,1)$
  Reality outputs $y_n \in \mathbb{R}$
  $\mathrm{err}_n^\epsilon := \begin{cases} 1, & \text{if } y_n \in \Gamma_n^\epsilon \\ 0, & \text{otherwise} \end{cases}$ for all $\epsilon \in (0,1)$;
  $\mathrm{Err}_n^\epsilon := \mathrm{Err}_{n-1}^\epsilon + \mathrm{err}_n^\epsilon$ for all $\epsilon \in (0,1)$;
  $l_n^\epsilon := |\Gamma_n^\epsilon|$ for all $\epsilon \in (0,1)$;
  $L_n^\epsilon := \mathrm{Median}(l_1^\epsilon, \ldots, l_n^\epsilon)$ for all $\epsilon \in (0,1)$
end for

It is known that prediction intervals produced by the Gauss linear model (5.1) are uninformative before the number of observed examples reaches $K+3$. This chapter uses an extended notion of validity. We say that a confidence predictor *valid in the Gauss linear model* if for any significance level $\epsilon \in (0,1)$ and for any step $n \geq K+3$ it makes an error with probability ϵ, w.r. to the model, and the errors are independent (cf. with

the definition of validity on page 32). Practically, the validity of predictions under the model means that for any step $n \geq K+3$ the ratio $\mathrm{Err}_n^\epsilon/n$ is about ϵ for any significance level $\epsilon \in (0,1)$.

Statistical fluctuations. The validity of predictions at a significance level ϵ can be checked by comparing the expected number of errors for the level with the observed value Err_n^ϵ. Often in practice these two numbers are close but not perfectly equal. To verify that the difference between the observed and expected numbers of errors can be explained due to statistical fluctuations one can apply statistical testing. It follows from Theorem 2 that values Err_n^ϵ are distributed according to binomial distribution $B(n,\epsilon)$ with parameters n (number of trials) and ϵ (probability of success), and therefore it can be used to find the probability of observing the actual number of errors. The probability to have more than k errors after n trials is

$$\mathbb{P}_{\mathrm{Err}_n^\epsilon \sim B(n,\epsilon)}(\mathrm{Err}_n^\epsilon \geq k) = F(n-k, n, 1-\epsilon),$$

where $F(x,n,p)$ is the value of the distribution function for binomial distribution $B(n,p)$ at point x. Having found this probability one can report a significance level to accept the hypothesis that $\mathrm{Err}_n^\epsilon \sim B(n,\epsilon)$, i.e. the predictions are valid.

5.2.2. OCM for the Gauss linear assumption

Let us restate the definition of an OCM (page 28) in the context of the Gauss linear model (5.1).

First, the summary in the context of the Gauss linear model is constructed as follows. The statistics for the Gauss linear model are

$$S_n(x_1, y_1, \ldots, x_n, y_n) := \left\{ x_1, \ldots, x_n, \sum_{i=1}^n y_i x_i, \sum_{i=1}^n y_i, \sum_{i=1}^n y_i^2 \right\}.$$

Under the Gauss linear model these are the sufficient statistics, and therefore such summaries contain all information about the distribution of $(x_1, y_1), \ldots, (x_n, y_n)$ (see further details, e.g., in [Shafer and Vovk, 2008] on p. 412).

The OCM for the Gauss linear model, called the *Gauss linear OCM*, consists of five elements $(\Sigma, \square, \mathbf{Z}, F, B)$, where:

1. $\mathbf{Z} := \mathbb{R}^K \times \mathbb{R}$.

2. Σ is the set of elements S_n, such that for each integer n and for each sequence $((x_1, y_1), \ldots, (x_n, y_n))$ the summary S_n is defined by

$$S_n = S_n(x_1, y_1, \ldots, x_n, y_n) := \left\{ x_1, \ldots, x_n, \sum_{i=1}^{n} y_i x_i, \sum_{i=1}^{n} y_i, \sum_{i=1}^{n} y_i^2 \right\};$$

   (as usual, $\sum_{i=1}^{0} y_i x_i$ and $\sum_{i=1}^{0} y_i$ are 0 and the empty summary $\square := \{0, 0, 0\}$).

3. The forward function F updates a summary S_n after observing a new example (x_{n+1}, y_{n+1}) as follows

$$S_{n+1} = F(S_n, (x_{n+1}, y_{n+1})) := \left\{ x_1, \ldots, x_n, x_{n+1}, \sum_{i=1}^{n+1} y_i x_i, \sum_{i=1}^{n+1} y_i, \sum_{i=1}^{n+1} y_i^2 \right\}.$$

4. For each summary $S_n \in \Sigma$ the backward kernel $B(S_n)$ defines the joint probability distribution over pairs $(S, (x, y)) \in \Sigma \times \mathbf{Z}$ as described below.

Backward kernel distribution for the Gauss linear OCM. According to the definition of the summary for the Gauss linear assumption, S_n contains all the observed objects $x_i, i = 1 \ldots, n$. Therefore for given S_n the backward kernel $B(S_n)$ actually describes a distribution of sequences $(\tilde{y}_1, \ldots \tilde{y}_n)$ that are consistent with the summary S_n. Such sequences have to agree with $K + 1$ linear equations (for each of $K + 1$ coordinates of $\mathbf{z}_n$) and the sum $\sum_{i=1}^{n} y_i^2$ is fixed. Consider a sphere which is an intersection of

the n-dimensional sphere of radius $\sum_{i=1}^{n} y_i^2$ centred at the origin and the hyperplane $(c_1, \ldots, c_n)$ defined by

$$\sum_{i=1}^{n} c_i = \sum_{i=1}^{n} y_i;$$
$$\sum_{i=1}^{n} c_i \cdot x_{i,p} = \sum_{i=1}^{n} y_i x_{i,p}, \quad p = 1, \ldots, K.$$

Then the sequences $(\tilde{y}_1, \ldots \tilde{y}_n)$ distribute uniformly over the sphere of intersection. These sequences specify corresponding values for $\tilde{S}_{n-1}$ and $\tilde{y}_n$ such that $F(\tilde{S}_{n-1}, \tilde{y}_n) = S_n$, that gives us the distribution of $(S, (x, y)) \in \Sigma \times \mathbf{Z}$ conditional on S_n.

5.3. Conformal prediction under the Gauss linear OCM

This section discusses conformal prediction under the Gauss linear assumption. It begins with the description of efficient calculations for prediction intervals introduced in [Vovk et al., 2005b, Section 8.5], and follows by an empirical study of predictions for synthetic data sets.

5.3.1. Efficient calculations of prediction intervals

The number of examples that are used to obtain a prediction interval under the Gauss linear model needs to be greater than $K + 2$, otherwise the prediction is the real line $\mathbb{R}$. For each $l = K + 3, K + 4, \ldots$, define $\mathbf{Z}_l$ the matrix of size $l \times (K + 1)$ whose rows are $\mathbf{z}_i', i = 1, \ldots, l$, and $\mathbf{y}_l$ the column vector of length l whose elements are labels $y_i, i = 1, \ldots, l$. Protocol 5.4 summarises efficient calculations of prediction intervals under the Gauss linear OCM.

Let us comment on the calculations in Protocol 5.4 for $n \geq K + 3$. The least squares estimate for the regression coefficients $\hat{\gamma}_{n-1}$ is calculated using previous examples; $\hat{y}_n$ is

Protocol 5.4 Efficient calculations of prediction intervals under the Gauss linear OCM

Input: $((x_1, y_1), (x_2, y_2), \ldots)$ sequence of examples
ϵ significance level
Output: $(\Gamma^\epsilon_1, \Gamma^\epsilon_2, \ldots)$ prediction intervals for the label
for $n = 1, 2, \ldots$ **do**
  observe a new object x_n;
  if $n < K + 3$ **then**
    $\Gamma^\epsilon_n := \mathbb{R}$;
  else
    $\hat{\gamma}_{n-1} := (\mathbf{Z}'_{n-1}\mathbf{Z}_{n-1})^{-1}\mathbf{Z}'_{n-1}\mathbf{y}_{n-1}$;
    $\hat{y}_n := \hat{\gamma}_{n-1}\mathbf{z}_n$;
    $\hat{\sigma}^2_{n-1} := \frac{1}{n-K-2}(\mathbf{y}_{n-1} - \mathbf{Z}_{n-1}\hat{\gamma}_{n-1})'(\mathbf{y}_{n-1} - \mathbf{Z}_{n-1}\hat{\gamma}_{n-1})$;
    $\Delta_n := t^{\epsilon/2}_{n-K-2}\hat{\sigma}_{n-1}\sqrt{1 + \mathbf{z}'_n(\mathbf{Z}'_{n-1}\mathbf{Z}_{n-1})^{-1}\mathbf{z}_n}$, where $t^{\epsilon/2}_{n-K-2}$ is the upper $\epsilon/2$-quantile of the t-distribution with $n - K - 2$ degrees of freedom;
    $\Gamma^\epsilon_n := (\hat{y}_n - \Delta_n; \hat{y}_n + \Delta_n)$;
  end if
  observe the label y_n;
end for

the least squares prediction for object x_n; and $\hat{\sigma}^2_{n-1}$ is the standard estimate of σ^2 from $\mathbf{Z}_{n-1}$ and $\mathbf{y}_{n-1}$. The calculation of the prediction interval is based on a well-known fact that the ratio

$$t_n = \frac{y_n - \hat{y}_n}{\hat{\sigma}_{n-1}\sqrt{1 + \mathbf{z}'_n(\mathbf{Z}'_{n-1}\mathbf{Z}_{n-1})^{-1}\mathbf{z}_n}} \tag{5.2}$$

has the t-distribution with $n - K - 2$ degrees of freedom [Gosset (Student), 1908]. The prediction interval Γ^ϵ_n is obtained using quantiles of the t-distribution. From the definition of a quantile, follows that the true label $y_n \in \Gamma^\epsilon_n$ with probability $1 - \epsilon$, therefore it is a valid prediction interval. The intervals are precisely the predictions by the CP corresponding to the conformity measure

$$A\left(S_{n-1}(x_1, y_1, \ldots, x_{n-1}, y_{n-1}), (x_n, y_n)\right) := \frac{-|y_n - \hat{y}_n|}{\hat{\sigma}_{n-1}\sqrt{1 + \mathbf{z}'_n(\mathbf{Z}'_{n-1}\mathbf{Z}_{n-1})^{-1}\mathbf{z}_n}}. \tag{5.3}$$

For further details see [Vovk et al., 2005a, Section 8.5].

5.3.2. Empirical study

This section empirically studies predictions for randomly generated synthetic data sets. The validity of these predictions is evaluated for both cases – when data is generated according to the Gauss linear model as well as when the assumption is violated. In the case when data is generated according to the model, the efficiency of the predictions will be also assessed.

Synthetic data sets. For the experiments four data sets are generated. The data-generating program is written in R; for the results presented in this chapter the R random number generator is set to 0 before generating each of the four data sets. Each of these data sets consists of $N = 2000$ examples. Each example is described by $K = 100$ features and a label. Values for the features are generated independently from the normal distribution with zero mean and variance one. The label is calculated using (5.1) with the vector of coefficients $\gamma = (\gamma_1, \ldots, \gamma_{101})$ defined by

$$\gamma_k = \begin{cases} 100, & k = 1; \\ (-1)^{k+1} \cdot 10, & k = 2, \ldots, 11; \\ (-1)^{k+1}, & k = 12, \ldots, 101. \end{cases}$$

(According to this setting the first ten features of an object make a bigger contribution in the label value than the rest.) For the *Gauss* data set the noise variables ξ_i are generated from the normal distribution with zero mean and variance one. For the *Laplace*, *Exponential* and *Uniform* data sets ξ_i are drawn respectively from the Laplace distribution with location parameter zero and scale parameter $\frac{1}{\sqrt{2}}$, the exponential distribution with rate parameter one and the uniform distribution on $[-\sqrt{3}, \sqrt{3}]$. The parameters of these three distributions have been chosen to result in standard deviation one and mean zero (except for the exponentially distributed noise where mean is one). Figure 5.1

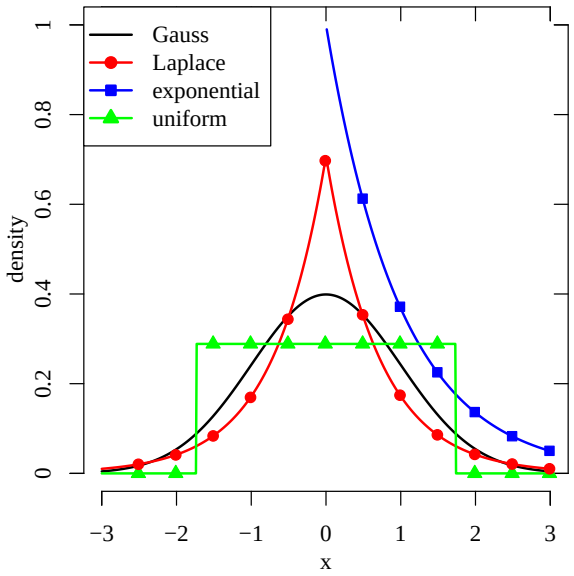

Figure 5.1.: Density functions for different types of noise used for synthetic data sets.

shows the density functions for corresponding noise variables.

Figure 5.2 shows error rates ($\mathrm{Err}_n^{0.01}$ and $\mathrm{Err}_n^{0.05}$ defined in Protocol 5.3) at the significance levels 1% (the left panel) and 5% (the right panel). This is an illustration of the validity property. As it was discussed in Section 5.2, the error rates are considered starting from step $K+3=103$ (Algorithm 5.4 produces informative on-line predictions starting from this step). The validity for predictions at the significance levels means that after 1897 successive on-line predictions (starting from the step 103) the total number of errors is about $1897\cdot 0.01\approx 19$ for the level 1% and it is $1897\cdot 0.05\approx 95$ for the level 5%. The dashed line in each of the plots shows the expected number of errors for each step of the on-line predictions. Let us check that for the Gauss data set the observed number of 121 errors for predictions at the significance level of 5% is within some allowed deviation. Using the approach described on page 111

$$\mathbb{P}_{\mathrm{Err}_{2000}^{0.05}\sim B(2000,0.05)}(\mathrm{Err}_{2000}^{0.05}\geq 121)\approx 0.02,$$

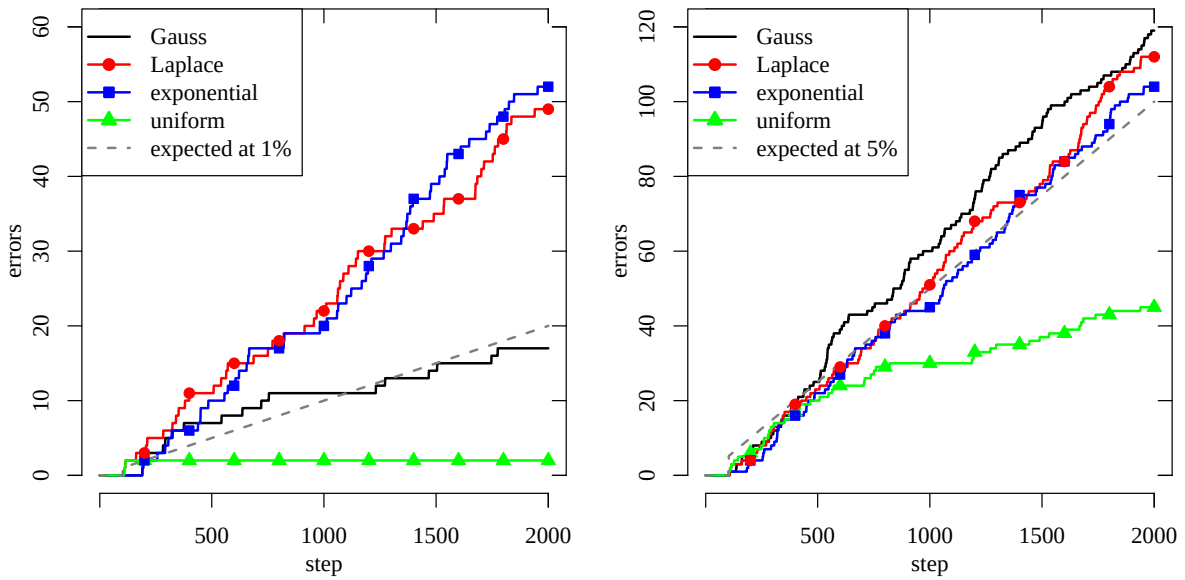

Figure 5.2.: Error rates for on-line prediction at the significance levels 1% (left) and 5% (right) are plotted for four synthetic data sets.

therefore the hypothesis that $\mathrm{Err}_{2000}^{0.05} \sim B(2000, 0.05)$ is accepted at the standard significance level of 0.01. The graphs support theoretical expectations: when the assumption is satisfied for data (the Gauss data set) the error rate (the solid line) is about the expected number of errors (up to statistical fluctuation). Next the efficiency of predictions for the Gauss data set is presented, and then the error rates for the rest of the data sets will be discussed.

Figure 5.3 shows the median length of prediction intervals ($L_n^{0.01}$ and $L_n^{0.05}$ defined in Protocol 5.3) at the significance levels 1% and 5% for the Gauss data set. It is known that if a distribution is approximately Gaussian then to cover about 99% of values one needs an interval of 2.5 standard deviation away from its mean value and to cover about 95% of values one needs an interval of two standard deviations away from its mean value. For the synthetic data (the standard deviation for noise in data is 1) it would give intervals of width 5 at 1% and intervals of width 4 at 5%. For the Gauss data set the CP generates intervals of width 5.4 at 1% and 4.2 at 5% (as soon as the sufficient number of examples has been observed).

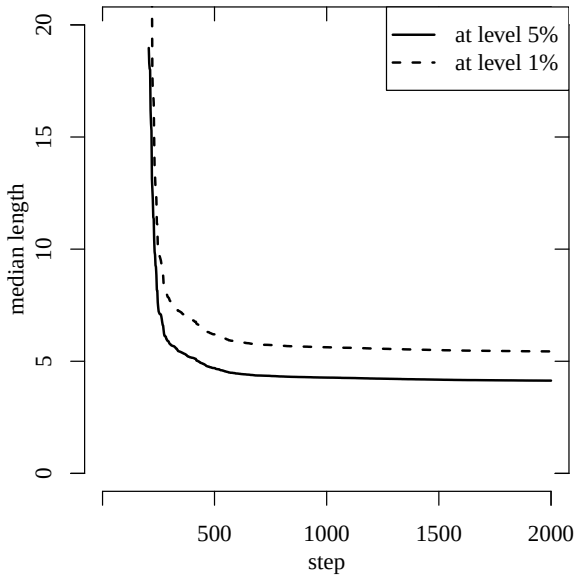

Figure 5.3.: Median length of prediction intervals for on-line prediction at the significance levels 1% and 5% are plotted for the synthetic Gauss data set.

In Figure 5.2 we can see different error rates when the assumption of the CP is violated (Laplace, Exponential and Uniform data sets). Each of these data sets corresponds to different deviations from the Gauss linear assumptions. The differences between the normal distribution and exponential or Laplace distributions result in non-valid prediction intervals at the level 1%, but at the level of 5% we can not see the non-validity for these two data sets. These two results show that when the assumption is violated predictions at certain significance levels (e.g., 5% in this experiment) may seem valid but they are non-valid at other levels (e.g., 1% in this experiment). For the Uniform data sets the results are non-valid but in another way: the number of errors are too low. Let us explain why the prediction quality (the number of errors) at 1% and 5% for the Uniform data set is better than that for the Gauss data set. Both of the data sets were generated with mean 0 and standard deviation 1. Consider the distributions of their noise variables. The uniform distribution is symmetric and has zero tails, and therefore it puts stronger bounds on possible values than the normal distribution does (see Figure 5.1). At small significance levels this results in a fewer number of errors for

data with uniform noise than it is expected for data with normal noise. But, of course, there is no guarantee that at other significance levels the corresponding number of errors will not exceed the levels.

Before applying this prediction method to real data we need to discuss the problem of testing the Gauss linear assumption. As Figure 5.2 shows, results of testing are important to be sure that at any significance level predictions under the OCM will be valid.

5.4. On-line testing the Gauss linear assumption

This section uses testing methods described in Chapter 4 for on-line testing the Gauss linear assumption. The main difference from testing exchangeability is that for the Gauss linear assumption p-values are output by conformal transducer under the Gauss linear OCM. Next the calculation of p-values under the Gauss linear OCM is described, and it follows by an empirical study of the performance of plug-in martingales for testing the Gauss linear assumption.

5.4.1. On-line calculation of p-values under the Gauss linear OCM

Consider a sequence of examples $((x_1, y_1), (x_2, y_2) \ldots)$; the goal is to generate the corresponding sequence of p-values under the Gauss linear assumption. Based on the same idea as for obtaining prediction intervals by Protocol 5.4, p-values for the true labels of the examples can be calculated efficiently. Consider calculations of p-value p_n for step n: the estimate for the regression coefficients $\hat{\gamma}_{n-1}$, the least squares prediction $\hat{y}_n$, and the estimate for the standard deviation $\hat{\sigma}_{n-1}$ are calculated using the sequence of previously observed examples $((x_1, y_1), \ldots, (x_{n-1}, y_{n-1}))$. The p-value p_n corresponding to a new example (x_n, y_n) is calculated using the value t_n of the random variable (5.2). Denote $t^{|t_n|}_{n-K-2}$ for the $|t_n|$-th quantile of the t-distribution with $n - K - 2$ degrees of freedom.

Then the p-value is

$$p_n := 2(1 - t^{|t_n|}_{n-K-1}).$$

Using the notation introduced in the previous section, Protocol 5.5 summarises on-line calculations of p-values under the Gauss linear OCM. (These p-values are computed using matrix operations, so complexity of this algorithm is polynomial in the number of examples n.)

Protocol 5.5 Generating p-values on-line under the Gauss linear OCM

Input: $((x_1, y_1), (x_2, y_2), \ldots)$ sequence of examples
Output: $(p_1, p_2, \ldots)$ sequence of p-values
for $n = 1, 2, \ldots$ **do**
  observe an example (x_n, y_n);
  if $n < K + 3$ **then**
    $p_n \sim U[0, 1]$;
  else
    $\hat{\gamma}_{n-1} := (\mathbf{Z}'_{n-1}\mathbf{Z}_{n-1})^{-1}\mathbf{Z}'_{n-1}\mathbf{y}_{n-1}$;
    $\hat{y}_n := \hat{\gamma}_{n-1}\mathbf{z}_n$;
    $\hat{\sigma}^2_{n-1} := \frac{1}{n-K-2}(\mathbf{y}_{n-1} - \mathbf{Z}_{n-1}\hat{\gamma}_{n-1})'(\mathbf{y}_{n-1} - \mathbf{Z}_{n-1}\hat{\gamma}_{n-1})$;
    $t_n := \frac{y_n - \hat{y}_n}{\hat{\sigma}_{n-1}\sqrt{1+\mathbf{z}'_n(\mathbf{Z}'_{n-1}\mathbf{Z}_{n-1})^{-1}\mathbf{z}_n}}$
    $t^{|t_n|}_{n-K-2}$ is the $|t_n|$-quantile of the t-distribution with $n-K-2$ degrees of freedom;
    $p_n := 2(1 - t^{|t_n|}_{n-K-2})$;
  end if
end for

The standard results of the theory of conformal prediction can be restated in the context of this chapter as follows: if examples are generated by a distribution that agrees with the Gauss linear OCM then p-values produced by Protocol 5.5 are independent and distributed uniformly in $[0, 1]$ (Theorem 2 on page 31).

Using the sequence of p-values plug-in martingales are calculated as it was described on page 94. In the current context, a high value of the martingales reflects a deviation from the Gauss linear assumption.

5.4.2. Empirical study

This section presents testing of the Gauss linear assumption for several data sets. The first set of experiments with the four synthetic data sets described (see Section 5.3.2) demonstrates that the procedure leads to reasonable results. The second set of experiments uses the benchmark Abalone data set: first on-line testing is applied to test two Gauss linear assumptions for this data, and then conformal predictions are obtained.

Synthetic data set

According to the way of generating the synthetic data sets, the Gauss data set agrees with the Gauss linear assumption and the assumption is violated for the rest of the data sets (Laplace, Exponential and Uniform). We expect the martingales to keep small values for the Gauss data set and to grow significantly for the rest of the data sets.

The first step of testing is calculating the sequence of p-values by Protocol 5.5. Then the p-values are observed one after another and the plug-in martingale (page 94) is calculated. Figure 5.4 shows the martingale growth for the data sets. The vertical axis is for the logarithm of martingale and the horizontal one is for the step (number of observed examples so far). The solid line with no characters on it shows the martingale for the Gauss data set: the logarithm of the martingale values is negative, i.e., the martingale values are about zero, meaning that the Gauss linear assumption is accepted (see Ville's inequality (4.1) on page 85). For the rest of the data sets (lines with different characters) the final values of the martingale are higher than 10^9, meaning that the Gauss linear assumption can be rejected at the significance level of 10^{-9}. This performance of the plug-in martingale demonstrates that the testing is valid (does not reject the model when it is true) and sensitive (detects different kinds of departure from the assumption).

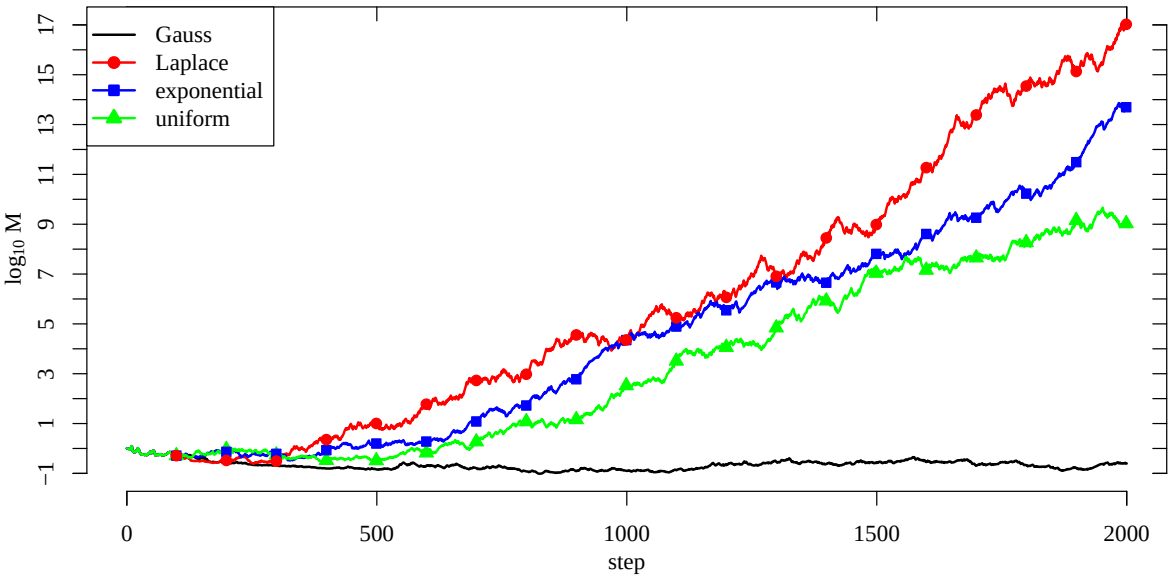

Figure 5.4.: The growth of the plug-in martingales for four synthetic data sets. The Gauss linear assumption is not rejected for the Gauss data set and can be rejected for the rest of these data sets at the significance level of 10^{-9}.

Abalone data set

The benchmark Abalone data set is described in Appendix A.4.

Testing. Snelson et al. [2004] studied predictions for the data using so-called warped gaussian process, and following their results it would be of interest to test two models:

1. The age of abalone depends on the rest of features according to model (5.1) with normally distributed noise.

2. The same as 1 but the logarithm of abalone age is used instead of the age.

Figure 5.5 shows the martingale growth for the data set when p-values are calculated according to models 1 (age) and 2 ($\log_{10}$ age). The vertical axis is for the logarithm of martingale values and the horizontal one is for the step. The final martingale value is 10^{37} for the first assumption and it is $10^{3.4}$ for the second one. In other words, using Ville's inequality (4.1) on page 85, at the significance level of 10^{-37} model 1 is rejected,

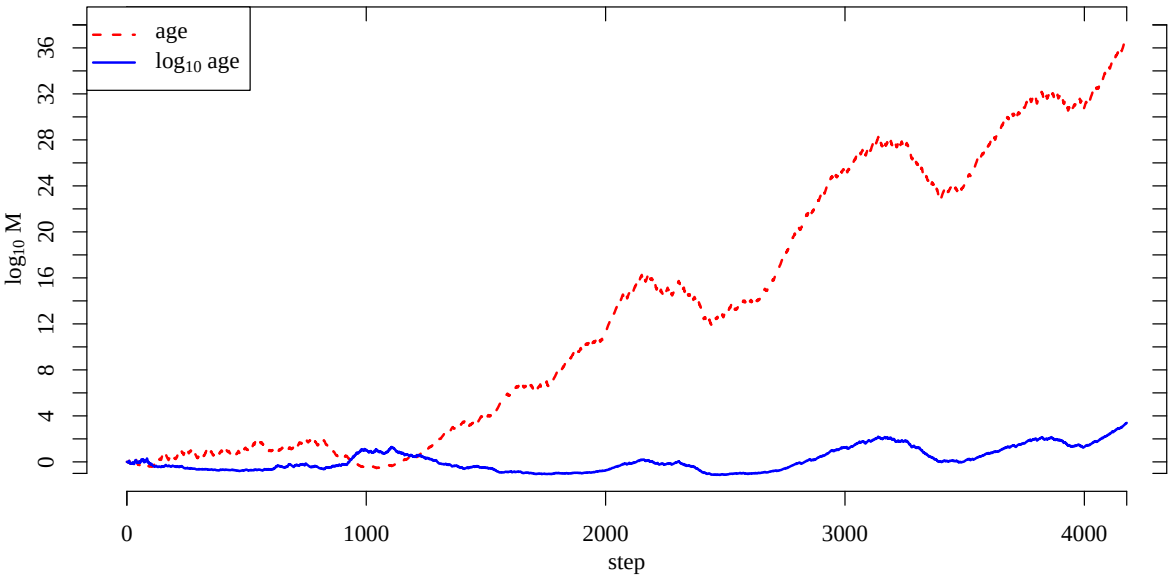

Figure 5.5.: The growth of the plug-in martingales for testing the Gauss linear assumption for (1) the abalone age or (2) the logarithm of abalone age. At the significance level of 10^{-37} the Gauss linear assumption for (1) is rejected, and at the significance level of 10^{-4} the assumption can be accepted for (2).

and at the significance level of 10^{-4} model 2 can be accepted. Therefore model 2 seems to be more realistic. (This agrees with the results for this data set in [Snelson et al., 2004].)

Prediction. Using the second model ($\log_{10}$ age) prediction intervals are computed for the logarithm of abalone age. The prediction intervals are calculated in the on-line mode (by Protocol 5.4). Figure 5.6 shows properties of the predictions at the significance levels 5% and 1%. The left plot is for the cumulative number of errors. The vertical axis is for the number of errors and the horizontal one is for the step. The black lines show observed errors ($\mathrm{Err}_n^{0.01}$ and $\mathrm{Err}_n^{0.05}$ defined in Protocol 5.3) and the grey lines are for the expected number of errors for each step at the chosen significance levels. The predictions are practically valid (in the sense mentioned in the end of Section 5.2.1). The right plot shows the median length of prediction intervals on the abalone age. The vertical axis is for the length of intervals on abalone age and the horizontal one is for the

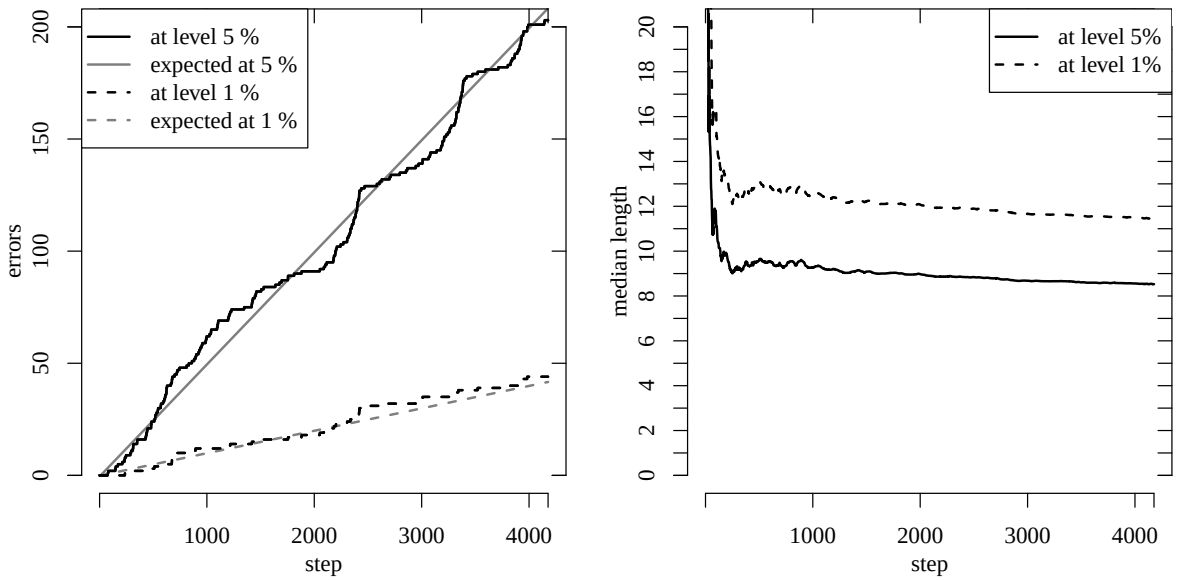

Figure 5.6.: On-line conformal prediction under the Gauss linear assumption for the logarithm of abalone age. Left plot: error rates at the significance levels 1% and 5%. Right plot: the median length of prediction intervals (on abalone age) at these significance levels.

step. Prediction intervals are obtained on the logarithm of abalone age, and then the corresponding values for the age are calculated. The final median length of the prediction intervals is 8.5 for the significance level of 5% and it is 11.5 for the significance level of 1%. Papadopoulos et al. [2011] used the Abalone data set and reported results for batch conformal prediction under the exchangeability assumption: the median length of prediction intervals is about 8 at the significance level of 5% and it is about 14 at 1%. In the current experiments the settings are different, yet it can be seen that these results on the median length of prediction intervals are typical for this data set.

5.5. Conclusion

This chapter has studied both conformal prediction under the Gauss linear assumption and testing this assumption. Efficient calculation of prediction intervals under this assumption has been discussed and studied empirically. The experiments on synthetic

data sets have illustrated properties of predictions for both cases when a data-generating mechanism agrees with the model as well as when this assumption is violated.

Plug-in martingales described in Chapter 4 have been applied for testing the Gauss linear assumption. The results for synthetic data sets support theoretical expectations: even for small departures from the assumption the final martingale values are large (it is more than 10^9 for the data sets of 2000 examples, meaning that the assumption is rejected at the significance level of 1%) and the martingale value is close to zero when the assumption is satisfied. To see the benefit of martingale testing let us compare Figure 5.4 and Figure 5.2: the martingale extracts information about agreement of data with the Gauss linear assumption that can not be seen clearly from tracking numbers of errors in predictions.

In addition, testing the Gauss linear assumption has been presented for the benchmark Abalone data set. The results suggest to use the Gauss linear assumption for the logarithm of abalone age. Finally, predictions on abalone age have been obtained by the CP under the Gauss linear model for the logarithm of abalone age. Again, the validity and efficiency of the predictions have been shown.

Though this chapter mainly discusses predictions and testing for the Gauss linear assumption, this approach is wider in the sense that OCMs for other assumptions and CPs under these OCMs can be constructed in order to test these assumptions or obtain predictions under them.

Chapter 6.

Conclusion

This thesis has studied conformal prediction under assumptions of certain on-line compression models and testing these assumptions. Based on the results of previous studies the following needs were identified in order to further research in this area:

1. the need for useful measures of information efficiency for conformal prediction;

2. the need to study efficiency of conformal predictors (CPs) under stronger assumptions and compare them to that of CPs under the traditional assumption of exchangeability;

3. the need to study and develop methods to test assumptions of on-line compression models, as the automatic validity of conformal predictions is proved under these models.

Next it is expounded how each of these needs was approached in this work.

It is known that CPs are automatically valid under their models. But there has previously been no systematic way of measuring the efficiency of conformal prediction. This work started by studying the different criteria of efficiency for conformal prediction. In the idealised setting, when a data-generating distribution is known, the idealised

conformity measures that optimise the criteria were described; as a result, some of the criteria were called "proper". The main advantage of proper criteria is that they encourage to use the conditional probability $P(y \mid x)$ as the conformity score for an example (x, y), whereas for other "improper" criteria optimal conformity measure may depend on the arbitrary choice of a "choice function" and encourage certain "strategic behaviour" (for example, for the M criterion, not assessing the differences, which may be important, between potential labels other than the value of a choice function $\hat{y}(x)$ for an object x). The experiments with empirical counterparts of the idealised conformity measures confirmed that the conclusions about the kind of behaviour encouraged by different criteria hold in the realistic setting. Hence it was suggested to use the proper criteria of efficiency in place of commonly used (improper) ones.

Conformal prediction was studied under hypergraphical models. These models represent specific assumptions about the structure of data. Several new CPs under these models were constructed and their performance was studied under the exchangeability model and under a non-trivial hypergraphical model. For the predictors, the model that is used for calculating p-values is called the "hard model" (the validity of the predictions depends on the correctness of the model), and the model that is used for calculating conformity scores is called the "soft model" (only efficiency may suffer if the assumption of the model is violated). The first empirical result is that hypergraphical models are useful for conformal prediction: predictions under more specific hypergraphical models are more efficient than those under the exchangeability model. The second observation is that for usual (unconditional) CPs hypergraphical models only need to be used as soft models; the performance of unconditional CPs does not suffer much if the exchangeability model continues to be used as the hard model. Finally, label conditional conformal prediction was studied under hypergraphical models. The experiments showed that the performance of label conditional CPs can be close to that of unconditional CPs when

hypergraphical models are used as both the hard and soft models.

The method of on-line testing assumptions was studied and extended. This testing is based on calculations of special martingales; the main goal is to construct martingales that can grow to larger values when the tested assumption is falsified. The martingales are functions of the sequence of p-values generated in the on-line mode by a conformal transducer. (Sequences of data that agree with assumptions of conformal transducers provide the corresponding sequences of p-values distributed uniformly in $[0, 1]$; therefore martingales for data can be functions of the p-values aiming to detect deviations from uniformity for the later.) All previously suggested martingales are constructed to grow if too many small p-values are in the sequence. This makes them unable to detect other possible deviations from uniformity in p-values (which corresponds to violations of the tested assumptions for data). This work introduced two new methods for constructing martingales from the sequence of p-values that do not require any assumptions about the p-values. The first method – mixtures of stepped martingales – is an example of the Bayesian approach in the context of martingale testing: families of stepped martingales were defined (with each of them constructed to grow for a different deviation from uniformity in p-values), and the uniform prior is used to integrate martingales from these families. The second method – plug-in martingales – is an example of the plug-in approach: a plug-in martingale learns the distribution of p-values from past data and adapts itself to grow if the distribution is not uniform. It was proved that, under a stationarity assumption, these plug-in martingales grow at least as high as any of previously introduced martingales. The empirical study showed that both of the new martingales are useful for on-line testing.

Finally, conformal prediction and on-line testing were studied under the Gauss linear assumption. The experiments demonstrated that the plug-in martingale detects different types of deviation from the Gauss linear assumption and is useful for this testing. In

addition, conformal prediction under this assumption was reported for a benchmark data set.

Future research

The following directions would be of interest for future study:

- Criteria of efficiency for conformal predictions as suggested in this thesis are for the classification problem, where the label space is finite. It would therefore be of interest to define similar criteria in the case of regression (the label space is $\mathbb{R}$), anomaly detection (the label space can be multi-dimensional and objects are absent) and clustering (labels are absent), and to describe idealised conformity measures that might be optimal for them.

- CPs whose hard model is the exchangeability model and the soft model is a hypergraphical model are very valuable, as their validity only requires the exchangeability assumption. The phenomenon that their performance does not greatly suffer in comparison to pure hypergraphical CPs deserves further investigation. This can be done in two ways: check it empirically for other data sets, and theoretically describe the expected difference in the performance of CPs based on different models.

- The performance of the mixtures of stepped martingales and the plug-in martingales were similar in the experiments presented in this thesis. It would therefore be of further interest to describe the situations when one of the martingales performs better than another and possible limitations of the martingales.

- This thesis discussed testing exchangeability and testing the Gauss linear assumption. This idea of testing can be applied to other statistical models, especially to hypergraphical models, in order to test different structures for data.

Appendix A.

Data sets

This appendix describes benchmark data sets used in the thesis.

A.1. USPS data set

The US Postal Service (USPS) data set is a collection of handwritten digits from real-life postal codes. The sizes of the training and test sequences are 7291 and 2007, respectively.

Each example is described by 256 features representing the brightness of pixels on the 16×16 gray-scaled image displaying a digit and its label (the digit). The brightness takes values in the interval $(-1, 1)$ and the label is a decimal digit from 0 to 9. Figure A.1 shows several examples from the data set.

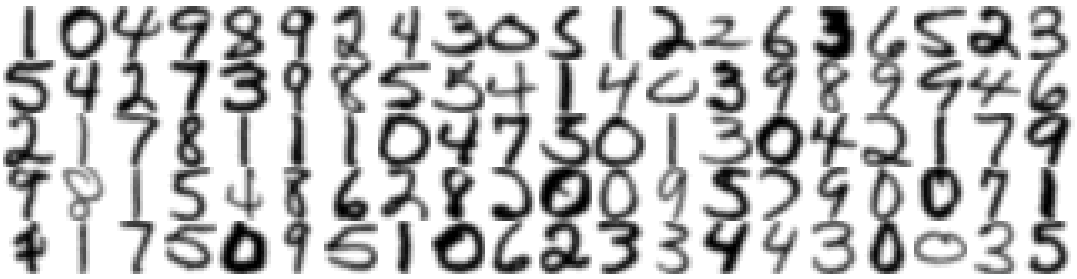

Figure A.1.: Examples of the USPS data set.

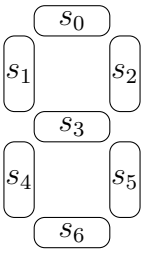

Figure A.2.: The seven-segment display.

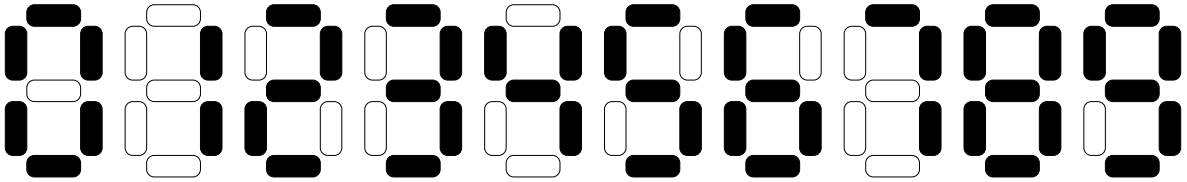

Figure A.3.: Ideal LED images.

For experiments in Chapter 2 each example is normalised as described in Appendix B.3 in Vovk et al. [2005a], so that the mean brightness of pixels in each picture is 0 and its standard deviation is 1.

For experiments in Chapter 4 the training and test sequences are merged and this is referred as the "USPS data set in the original order". A known fact is that the training and test sequences seem to have different distributions [Vovk et al., 2005a, Appendix B.1]. In order to satisfy the exchangeability assumption, the examples in the original USPS data set are randomly permuted, this is always stated in the experiments by saying "USPS data set randomly shuffled".

A.2. LED data sets

LED data sets are generated by a program from [Bache and Lichman, 2013]. The problem is to predict a digit from an image in the seven-segment display shown in Figure A.2.

Figure A.3 shows examples of objects in the data sets (these are the ten "ideal im-

ages" of digits; there are also digits corrupted by noise). The seven LEDs (light emitting diodes) can be lit in different combinations to represent a digit from 0 to 9. The program generates examples with noise. There is an ideal image for each digit, as shown in Figure A.3. An example has seven binary features $s_0, \ldots, s_6$ (s_i is 1 if the ith LED is lit) and the label c, which is a decimal digit. The program randomly chooses a label (0 to 9 with equal probabilities), inverts each feature in its ideal image with probability p_{noise} ($p_{\text{noise}} = 1\%$ for the experiments presented in this thesis) independently, and adds the noisy image and the label to the data set. As usual, examples are generated independently.

A.3. Statlog Satellite data set

The Statlog Satellite data set is available from [Bache and Lichman, 2013]. The data set consists of 6435 examples (divided into 4435 training examples and 2000 test examples).

The data were generated from a section (82×100 pixels) of an image of the earth surface taken by the Landsat satellite. The pictures are taken in four different spectral bands and so each pixel is described by four frequency values (each between 0 and 255). Each example is represented by 36 multi-spectral values of pixels in 3×3 neighbourhoods in the satellite image, and the label indicates the classification of the central pixel in each neighbourhood. Labels are integers from 1 to 7, excluding 6: class 1 is for red soil, class 2 is for cotton crop, class 3 is for grey soil, class 4 is for damp grey soil, class 5 is for soil with vegetation stubble, and class 7 is for very damp grey soil. Originally there was also a mixture class, where all types are present (class 6), but there are no instances of it in the data set.

For experiments in this thesis the whole Statlog Satellite data set is used, and in this case it is said "Statlog Satellite data set in the original order". In order to satisfy the exchangeability assumption, examples of the Statlog original data set are randomly

Table A.1.: The summary of features in the Abalone data set.

Name	Type	Description
Sex	nominal	M, F, and I (infant)
Length	continuous	longest shell measurement
Diameter	continuous	perpendicular to length
Height	continuous	with meat in shell
Whole weight	continuous	whole abalone
Shucked weight	continuous	weight of meat
Viscera weight	continuous	gut weight (after bleeding)
Shell weight	continuous	after being dried
Rings	integer	+1.5 gives the age in years

permuted, this is always stated in the experiments.

A.4. Abalone data set

The Abalone data set is available from [Bache and Lichman, 2013]. The usual process of determining the age of abalone is boring and time consuming. This data set aims to predict the age of abalone from various physical measurements that are easy to obtain.

The data set consists of 4177 examples. Each example is represented by 7 features and the label. Table A.1 describes the features and the label, which is named Rings. The label takes integer values from 1 to 29.

In this thesis the problem is considered as the regression problem and predictions are continuous numbers. To use the standard linear regression model (see Section 5.2), values of feature "Sex" are represented as integer numbers: 1 for M, 2 for I, and 3 for F.

Appendix B.

Implementation in R

R implementations of the most interesting methods studied in this thesis are available as several R packages. Each package is accompanied by its manual describing implemented methods and giving examples of use. The following list summaries the packages:

- (CP package) For Section 1.2 and some results of Chapter 2 the general framework for conformal prediction under the exchangeability assumption is available[1]. The package contains functions for on-line, batch and inductive conformal prediction. The implementation allows users to write new nonconformity functions (the "non-conformity" is the opposite to "conformity", i.e., for any conformity function A the corresponding nonconformity function $B := -A$ [Vovk et al., 2005a, see p. 23–24]) and plug them into the framework. The KNN-ratio conformal predictors (see (2.10) on page 51) are implemented for on-line, batch and inductive settings. The manual[2] describes the functions and shows examples of prediction for the USPS data set. The examples include experiments with the tangent distance. For this purpose the additional package[3] is used and its manual is available[4].

[1] `http://www.cs.rhul.ac.uk/~valentina/R-packages/CP_1.0.tar.gz`
[2] `http://www.cs.rhul.ac.uk/~valentina/R-packages/CP-manual.pdf`
[3] `http://www.cs.rhul.ac.uk/~valentina/R-packages/TangentDistance_1.0.tar.gz`
[4] `http://www.cs.rhul.ac.uk/~valentina/R-packages/TangentDistance-manual.pdf`

- (CriteriaForCP package) For Chapter 2 the package[5] implements criteria of efficiency for conformal prediction, and KNN-CoP and KNN-SP conformal predictors (for batch and inductive settings) (see (2.11) and (2.12) on page 52). The manual[6] gives examples of use the functions with the standard Iris data set, as well as shows some experiments from Chapter 2 for the USPS data set.

- (HCP package) For Chapter 3 the package[7] implements hypergraphical CoP and hypergraphical SP conformal predictors for the on-line settings. Also the calculation of idealised predictions for a given distribution is implemented. The manual[8] shows some experiments from Chapter 3 using the LED data set.

- (OnlineTesting package) For Chapter 4 an implementation of martingales is available[9]. The package contains the plug-in, mixtures of stepped martingales, simple mixture, sleepy jumper and power martingales. The manual[10] shows examples of calculating the martingales for randomly generated sequences of p-values, as well as it describes testing for the USPS data set.

Installation packages in R. The standard R function `install.packages` can be used to install package `<pkg-name>` from the downloaded archive `<pkg-name>.tar.gz` as it is shown in the script below.

```
# this is my path to <pkg-name>.tar.gz
path <- normalizePath("C://Downloads//<pkg-name>.tar.gz")
# install the package
install.packages(path, repos = NULL, type = "source")
# use the package
library(<pkg-name>)
```

[5] http://www.cs.rhul.ac.uk/~valentina/R-packages/CriteriaForCP_1.0.tar.gz
[6] http://www.cs.rhul.ac.uk/~valentina/R-packages/CriteriaForCP-manual.pdf
[7] http://www.cs.rhul.ac.uk/~valentina/R-packages/HCP_1.0.tar.gz
[8] http://www.cs.rhul.ac.uk/~valentina/R-packages/HCP-manual.pdf
[9] http://www.cs.rhul.ac.uk/~valentina/R-packages/OnlineTesting_1.0.tar.gz
[10] http://www.cs.rhul.ac.uk/~valentina/R-packages/OnlineTesting-manual.pdf

Bibliography

Kevin Bache and Moshe Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Vineeth N. Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk, editors. *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations, and Applications*. Elsevier, Waltham, MA, 2013. To appear.

José M. Bernardo and Adrian F. M. Smith. *Bayesian Theory*. Wiley, Chichester, 2000.

Robert G. Cowell, A. Philip Dawid, Steffen L. Lauritzen, and David J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer, New York, 1999.

David R. Cox and David V. Hinkley. *Theoretical Statistics*. Chapman and Hall, London, 1974.

Luc Devroye and László Györfi. *Nonparametric Density Estimation: the L1 View*. Wiley series in probability and mathematical statistics. Wiley, 1985.

Valentina Fedorova, Ilia Nouretdinov, and Alex Gammerman. Testing the Gauss linear assumption for on-line predictions. *Progress in Artificial Intelligence*, 1:205–213, 2012a.

Valentina Fedorova, Ilia Nouretdinov, Alex Gammerman, and Vladimir Vovk. Plug-in martingales for testing exchangeability on-line. In *Proceedings of the Twenty Ninth*

International Conference on Machine Learning (ICML 2012), Edinburgh, Scotland, UK, 2012b.

Valentina Fedorova, Alex Gammerman, Ilia Nouretdinov, and Vladimir Vovk. Conformal prediction under hypergraphical models. On-line Compression Modelling project (New Series), Working Paper 9, 2013a. URL `http://alrw.net`. This is an extended version of the paper appeared in the AIAI 2013 Proceedings.

Valentina Fedorova, Alex Gammerman, Ilia Nouretdinov, and Vladimir Vovk. Conformal prediction under hypergraphical models. In *Proceedings of the Ninth International Conference on Artificial Intelligence Applications and Innovations*, pages 371–383, Paphos, Cyprus, 2013b.

Valentina Fedorova, Alex Gammerman, Ilia Nouretdinov, and Vladimir Vovk. Conformal prediction under hypergraphical models. In *Proceedings of Imperial College Computing Student Workshop*, pages 27–34, London, UK, 2013c.

Alex Gammerman, editor. *Special Issue on Conformal Prediction and its Applications*, volume 1 of *Progress in Artificial Intelligence*. Springer, 2012.

Alex Gammerman and Vladimir Vovk. Hedging predictions in machine learning. On-line Compression Modelling project (New Series), Working Paper 2, 2007. URL `http://alrw.net`.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. Technical report, 2004.

William S. Gosset (Student). The probable error of a mean. *Biometrika*, 6:1–25, 1908.

Irwin Guttman, editor. *Statistical Tolerance Regions: Classical and Bayesian*. Griffin, London, 1970.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, second edition, 2013.

Shen-Shyang Ho. A martingale framework for concept change detection in time-varying data streams. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, pages 321–327, Bonn, Germany, 2005.

Ulf Johansson, Rikard Konig, Tuve Lofstrom, and Henrik Bostrom. Evolved decision trees as conformal predictors. In *Proceedings of the 2013 IEEE Conference on Evolutionary Computation*, volume 1, pages 1794–1801, Cancun, Mexico, 2013.

Solomon Kullback. *Information Theory and Statistics*. Wiley, New York, 1959.

Erich L. Lehmann. *Testing Statistical Hypotheses*. Springer, New York, second edition, 1986.

Jing Lei and Larry Wasserman. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society*, 76:71–96, 2014.

Jing Lei, James Robins, and Larry Wasserman. Distribution free prediction sets. *Journal of the American Statistical Association*, pages 278–287, 2013. Preliminary version published as Technical Report arXiv:1111.1418 [math.ST].

Thomas Melluish, Craig Saunders, Ilia Nouretdinov, and Vladimir Vovk. Comparing the Bayes and typicalness frameworks. In *Proceedings of the Twelfth European Conference on Machine Learning*, pages 360–371. Springer, 2001.

Harris Papadopoulos, Vladimir Vovk, and Alex Gammerman. Regression conformal prediction with nearest neighbours. *Journal of Artificial Intelligence Research*, 40: 815–840, 2011.

Harris Papadopoulos, Alex Gammerman, and Vladimir Vovk, editors. *Special Issue on Conformal Prediction and its Applications*. Annals of Mathematics and Artificial Intelligence. Springer, 2014. To appear (most of the papers published on Online First).

Mark J. Schervish. *Theory of Statistics*. Springer, New York, 1995.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9:371–421, 2008.

Edward Snelson, Carl E. Rasmussen, and Zoubin Ghahramani. Warped Gaussian processes. In *Advances in Neural Information Processing Systems 16 (NIPS)*, 2004.

Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, pages 1134–1142, 1984.

Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

Jean Ville. *Etude Critique de la Notion de Collectif*. Gauthier-Villars, Paris, 1939.

Vladimir Vovk, Ilia Nouretdinov, and Alex Gammerman. Testing exchangeability on-line. In *Proceedings of the 20th International Conference on Machine Learning (ICML 2003)*, pages 768–775, Washington, DC, 2003.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, New York, 2005a.

Vladimir Vovk, Ilia Nouretdinov, and Alex Gammerman. On-line predictive linear regression. On-line Compression Modelling project (New Series), Working Paper 1, 2005b. URL `http://alrw.net`.

Vladimir Vovk, Valentina Fedorova, Ilia Nouretdinov, and Alex Gammerman. Criteria of efficiency for conformal prediction. On-line Compression Modelling project (New Series), Working Paper 11, 2014. URL `http://alrw.net`.