Toward a scalable Bayesian workflow

Yuling Yao

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2021

Abstract

Toward a scalable Bayesian workflow

Yuling Yao

A scalable Bayesian workflow needs the combination of fast but reliable computing, efficient but targeted model evaluation, and extensive but directed model building and expansion. In this thesis, I develop a sequence of methods to push the scalability frontier of the workflow.

First, I study diagnostics of Bayesian computing. The Pareto smoothed importance sampling stabilizes importance weights using a generalized Pareto distribution fit to the upper tail of the distribution of the simulated importance ratios. The method, which empirically performs better than existing methods for stabilizing importance sampling estimates, includes stabilized effective sample size estimates, Monte Carlo error estimates and convergence diagnostics. For variational inference, I propose two diagnostic algorithms. The Pareto smoothed importance sampling diagnostic gives a goodness of fit measurement for joint distributions, while the variational simulation-based calibration assesses the average performance of point estimates. I further apply this importance sampling strategy to causal inference and develop diagnostics for covariate imbalance in observational studies.

Second, I develop a solution to continuous model expansion using adaptive path sampling and tempering. This development is relevant to both model-building and computing in the workflow. For the former, I provide an automated way to connect models via a geometric bridge such that a supermodel encompasses individual models as a special case. For the latter, I use adaptive path sampling as a preferred strategy to estimating the normalizing constant and marginal density, based on which I propose two metastable sampling schemes. The continuous simulated tempering

aims at multimodal posterior sampling, and the implicit divide-and-conquer sampler aims for a funnel-shaped entropic barrier. Both schemes are highly automated and empirically perform better than existing methods for sampling from metastable distributions.

Last, a complete Bayesian workflow distinguishes itself from a one-shot data analysis by its enthusiasm for multiple model fittings, and open-mindedness to model misspecification. I take the idea of stacking from the point estimation literature and generalize to the combination of Bayesian predictive distributions. Using importance sampling based leave-one-out approximation, stacking is computationally efficient. I compare stacking, Bayesian model averaging, and several variants in a decision theory framework. I further apply the stacking strategy to multimodal sampling in which Markov chain Monte Carlo algorithms can have difficulty moving between modes. The result from stacking is not necessarily equivalent, even asymptotically, to fully Bayesian inference, but it serves many of the same goals. Under misspecified models, stacking can give better predictive performance than full Bayesian inference, hence the multimodality can be considered a blessing rather than a curse. Furthermore, I show that stacking is most effective when the model predictive performance is heterogeneous in inputs, such that it can be further improved by hierarchical modeling. To this end, I develop hierarchical stacking, in which the model weights are input-varying yet partially-pooled, and further generalize this method to incorporate discrete and continuous inputs, other structured priors, and time-series and longitudinal data—big data need big models, and big models need big model evaluation, and big model evaluation itself needs extra data collection and model building.

Table of Contents

III Learning from many models by Bayesian stacking

List of Algorithms

List of Propositions

Acknowledgements

Starting by "this thesis would not have been possible without these people" would be cliché and kitsch. Such claim misleadingly defines the causal effect of an unrealistic and irreproducible intervention, although part of this thesis addresses leave-one-out cross validation or influence analysis. That being said, the aid from my advisor, collaborators and friends throughout my graduate study and research is not a latent parameter to estimate; it is input that is palpable and treasurable.

It would be unscalable to exhaustively list all the support I enjoyed from my doctoral advisor Andrew Gelman. Aside from inspirations I have learned from his blackboard and chalks, emails, overleaf editing logs, and those meetings with three hundred topics emerging in the same room, Andrew presents me an example of what a good statistician could be. To the same extent that model fitting benefits from fake data simulation, my trajectory toward an aspiring researcher is boosted by these simulations of fake Andrew in my head—"What would Andrew say to a sloppy graph? What model would Andrew consider given this dataset? What insights would Andrew write down on his pocket notebook after reading these papers?"—Well, even this metaphor is adapted from his.

I am grateful to Aki Vehtari for his guidance and discussions on many, if not all, of my research works. In addition to all bibliographies he meticulously pointed me to, a simulated Aki would also appear in the previous imaginary loop: "What would Aki say to this new idea? Would Aki be happy with this software implementation?"

I would like to thank collaborators with whom I have worked closely: Dan Simpson, Lex van Geen, Bob Carpenter, Yu-Sung Su, Jonah Gabry, Ben Bales, Jonathan Auerbach, Gregor Pirš, Charles Margossian, Collin Cademartori, and others. As a universal statistical principle applies, "the most important thing is what data you work with, not what you do with the data", whose corollary lower bounds how much I have already learned from *stacking* all these collaborators.

Chapter 1. Introduction: Pushing the frontier of a scalable workflow

"Had we but world enough and time,
This coyness, were no crime.
We would sit down, and think which way
To walk, and pass our long love's day."
—Andrew Marvell

There are various levels of Bayesian statistical practice. To begin with, given one observed dataset and one belief model, *Bayesian inference* or Bayesian computation goes from likelihood and prior to posterior, which is a deterministic procedure followed by Bayes' rule, and is only judged by whether the computation result or its approximation is faithful to the posterior distribution.

On top of Bayesian inference, *Bayesian logic* addresses decision theory. Bayesian logic is not about applying the Bayes rule, but quantifying uncertainty using the probability theory. Consequently, there is a tension between Bayesian inference and Bayesian decision theory: the prima-facie-always-coherent Bayesian posterior predictions would not be Bayes-optimal unless both the model is correct and the prediction is averaged under the prior replications. These assumptions are rarely valid. Bayesian inference thereby does not automatically ensure an optimal prediction nor a calibrated uncertainty estimation—the model is misspecified; the correct prior is not even always well-defined; the target utility function may care more about what has been modeled, such as different prediction horizon in time series, the calibration and the generalization ability to a new area, or the intervals and quantiles of the outcome.

As a more realistic procedure, *Bayesian data analysis* involves three steps of model building, inference, and model checking and improvement. A complete *Bayesian workflow* refers to Bayesian data analysis for a sequence of models, and more generally would require efforts on data inquiry and experimental design to facilitate this loop. To evolve models in the workflow serves two goals. First, via an open-ended and interactive model modification, selection, averaging, we are building better models in terms of better outcome prediction, uncertainty quantification, and the decision

making. Second, the workflow is toward model understanding. A model is not only a prediction it could generate, but also a probabilistic explanation of the data generating process.

Taking all steps into account, to push the scalability boundary of the Bayesian workflow requires advances in three modules:

1. **Fast but reliable computing.** With the advent of Markov chain Monte Carlo (MCMC) algorithms such as the Gibbs sampler, Metropolis, and Hamiltonian Monte Carlo, along with generic black-box implementations such as Stan, modern Bayesian inference is highly automated. With a regular density and an ergodic sampler, the Monte Carlo estimate is ensured accurate with a long enough run.

   However, there are problems in which we cannot afford exact MCMC: when the data size is too large, when models are too complex, when the posterior distributions contain pathological geometries, or when we are interested in the tail of the posterior distribution where we almost never have enough Monte Carlo draws. These challenges require either a specialized sampler that targets the particular posterior geometry we are working with, or a specialized post-processing technique that reduce finite-draw Monte Carlo errors by reusing existing draws. This thesis presents progress in both directions.

   With even more data and bigger models, faster or parallelizable algorithms like Laplace approximation, variational inference (VI), approximation Bayesian computation (ABC), and expectation propagation (EP) provide alternative approaches to approximate Bayesian inference. Besides the requirement on how to further speed up these algorithms, computation diagnoses are a necessary part of the workflow to tell the reliability of computing.

2. **Efficient but targeted model evaluation.** For many models, the implicit goal is to predict future unseen outcomes. We typically judge the prediction accuracy by some scoring rules or user-specific utility functions, and evaluate the model performance by posterior predictive check or cross-validation. Besides checking whether the current model is sensible, model evaluation serves as an intermediate step toward model improvement, selection, or averaging,

hence needs to be called repeatedly. With a small dataset, we need to handle the large variance in model evaluation. With a big dataset, exact cross-validation is intensive and needs efficient approximations.

With conditionally independent and identically distributed (IID) outcomes, this model evaluation step can be automated, such as using importance sampling based leave-one-out. Later in the thesis, we present other complications that require a tailored model evaluation: when the data has more structures such as spatial, temporal, longitudinal, sequential, or causal components; when we have a specific quantity to infer such as the treatment effect in an observational study, the transformation of individual outcomes, the interval estimate of posterior predictions; or when the future prediction is known to happen under covariate shift or at a given covariate location.

3. **Extensive but directed model building/expansion.** The model complexity is not necessarily the same as the modeling complexity: how much human effort is needed to build, customize and improve the model. An over-parameterized deep neural network is large in terms of model expressiveness, but can also be routinely applied to a family of data analysis tasks with minimal user adaptation.

   Big data not only have a big sample size and a high dimension, but also a complex data structure: open-ended observational studies rather than randomized experiments, corrupted measurement with errors rather than clean data, hierarchical data collection with potential imbalance rather than iid samples. All of these complications are extra information we would like to encode during bespoke modeling, often through an iterative procedure on a broad class of models. However, repeatedly building and fitting new models pose a cost to human intelligence and computation resource. This cost results in a tension between automation and customization, between the complexity of the workflow and the final accuracy of the model fitting, and between a fit-as-many-models-as-you-want and a one-fit-all attitude.

   To achieve a scalable workflow, it is then tempting to direct, if not fully automate, this model

building step such that new models are generated from existing ones. This present thesis introduces two possible directions toward this goal: to expand one or several models using path sampling, and to combine many models using Bayesian stacking.

Which piece above is the main scalability bottleneck in the workflow? There is no static answer. In bygone days, Bayesian statistics was criticized as too impractical to apply in any but the simplest models for the difficulty of computing. With modern generic computation algorithms, Bayesian methods are sometimes criticized to be fit too easily and thoughtlessly. The relative ease of inference puts more of a burden on model building and evaluation. Furthermore, these three tasks are often tangled, which makes the challenge of scalability a product rather than a summation of each piece. On the brighter side, this entanglement also suggests we could bypass the difficulty in one piece by fixing problems in other parts. To name a few,

- Model evaluation requires implicit model building. For example, the usual leave-one-out cross-validation reuses data as pseudo Monte Carlo draws. The pointwise model evaluations in spatial, temporal, longitudinal, multilevel or causal data are themselves spatial, temporal, longitudinal, multilevel or causal. Though a sample average is sometimes enough for the overall model performance, careful modeling with potentially different assumptions is needed to assess fine-grained model behavior.

- Model building should be aware of what inference algorithm it will be equipped with. For example, the simple Lasso is well compatible with the max-a-posteriori (MAP) estimate, whereas the full Bayes solution to the same Laplace prior regression is often inferior in predictions; The MAP estimate of group-level standard deviation in a hierarchical model equipped with an improper gamma prior is often desired, whereas the Bayes solution to the same zero avoiding prior model is highly biased. This concern is especially relevant when using pre-trained model architectures.

- Model building should be aware of what decision problem it will be used in. For example in time series analysis, a model good at predicting the one-day-ahead outcome is not necessarily

good at one-month-ahead. Causal inference is another example in which the prediction is targeted on the counterfactual conditional prediction.

- The computation problem is often an indicator of model misspecification. When the computation is too slow for one model, we have the freedom to (a) wait longer, (b) find a quicker sampler that is more compatible with the current posterior, (c) adopt faster computation approximations, (d) simplify or regularize the model to reduce computing burden, or (e) modify the model to address modeling issues.

- The computing budgets are competing between flawlessly fitting one model and roughly fitting many models. Put it in another way, the ultimate purpose of Bayesian computation is not to compute the posterior integral from a given model as accurately as possible, but to facilitate the model building-fitting-improvement loop.

Along with this goal toward a scalable workflow, this thesis contains a collection of methodology progresses, covering many directions listed above. The remaining chapters are grouped into three themes: making use of the unnormalized posterior density before sampling, expanding a model using tempering and path sampling, and combining models using stacking. Many chapters are adapted from previously published articles, and are rearranged in a logically coherent order: Some discussions in this introduction are from Gelman et al. (2020) and Gelman and Yao (2021). Chapter 2 introduces Pareto smoothed importance sampling, adapted from Vehtari et al. (2019b). Chapter 3 applies this idea to diagnosing the approximation accuracy of variational inference, previously published as Yao et al. (2018b). Chapter 4 explores this tail-shape-of-density-ratio approach to diagnose covariate imbalance in observations studies.

Chapter 5 discusses the challenge of normalization constant estimation and its relevance to Bayesian computing and model expansion. The technique is also useful as a postprocessing tool to reduce finite sample Monte Carlo errors in general marginal density estimation. Chapter 6 proposes a continuously tempered sampler using adaptive path sampling. This technique is dual-purposed—As a computation approach, the tempering method makes it viable to efficiently sample from a

metastable density; As a modeling approach, tempering provides a solution to expand or distillate models via geometric bridges. These two chapters are modified from Yao et al. (2020a) along with extra results.

Chapter 7 presents the stacking framework, a semi-automated solution to combining Bayesian models, merging Yao et al. (2018a) and Yao (2019). The comparison between Bayesian model averaging and stacking is another reflection of the tension between Bayesian inference and Bayesian logic. Chapter 8 applies this stacking approach to combine non-mixing Bayesian computations, proposed in Yao et al. (2020b). It is an alternative solution to metastable density sampling, although the inference result typically differs from the exact posterior such as from path sampling. This approach is relevant to an efficient workflow in which we free up asynchronous computation time that can be reallocated to explorations of more models. Chapter 9 generalizes the stacking idea to Bayesian hierarchical stacking, adapted from Yao et al. (2021b). Chapter 10 carries on the discussion of several asymptotic properties of stacking and hierarchical stacking.

Finally, this thesis is concluded by a handful of open questions for future investigations.

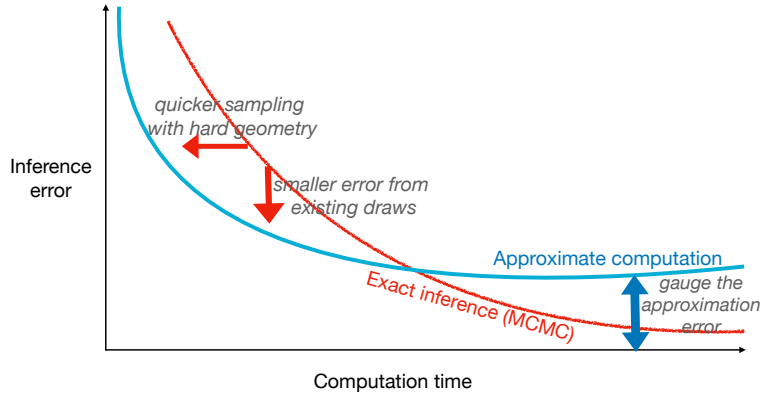


Figure 1.1: *Toward scalable Bayesian computation. Bayesian inference typically relies on slow-but-eventually-accurate MCMC methods or fast-but-likely-corrupted approximations. For the former, we would like to push the efficiency boundary by either sampling quicker from challenging posterior densities or reducing inference error from existing posterior draws. For the latter, we would like a more reliable pipeline that diagnoses how large the approximation error is. This present thesis presents an approach to variational inference diagnostics, a tempering algorithm that is designed for efficient sampling in metastable posterior densities, and a method for reducing Monte Carlo errors from finite posterior draws.*

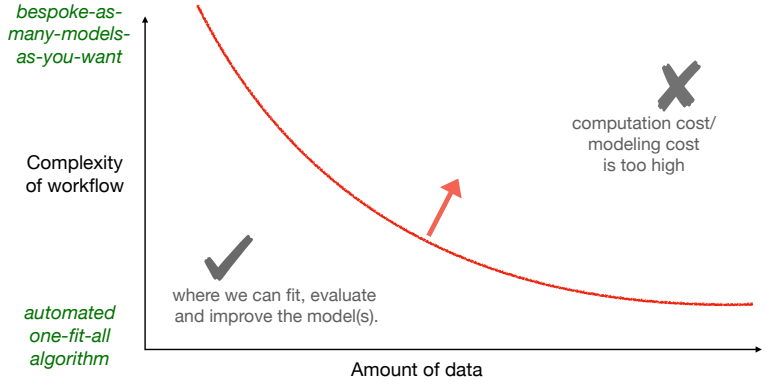


Figure 1.2: *Yet more importantly, toward a scalable Bayesian workflow. The workflow contains model building, fitting, evaluation, and improvement, often in an iterative loop. Given a limited budget, we are facing (a) the tradeoff between computation and model building: whether to accurately fit one model and to roughly fit many models and (b) the tradeoff between the complexity of the workflow and the final model fitting accuracy: whether to build and fit many models or to apply an automated one-fit-all algorithm. Besides the scalable computation techniques in Figure 1.1, this thesis contains several other progress toward an efficient workflow, including (a) using tempering and path sampling to expand one or several models, (b) using stacking as a semi-automated solution to building meta-models, and (c) combining non-mixing inferences to free up asynchronous computation time.*

Part I

Learning before sampling:

Making the most of the unnormalized density

Chapter 2. Importance sampling and Pareto smoothed importance sampling[1]

"To see a world in a grain of sand, And a heaven in a wild flower,
Hold infinity in the palm of your hand, And eternity in an hour."
—William Blake

2.1 When importance sampling would work or fail

The posterior distribution $p(\theta|y)$ is central to Bayesian computation. Suppose the goal is to compute the posterior expectation of some function $h(\theta)$, one natural solution is to draw simulations from the distribution $p(\theta|y)$, and compute this expectation using Monte Carlo integral. But even before we can draw any sample, typically we are able to evaluate this target density (up to a normalizing constant) and the function of interest. In this chapter we would suppress dependence on y and refer to this target density as $p(\theta)$. In this chapter, we investigate importance sampling, the methodology foundation for two other diagnostics we will introduce in the next two chapters.

Importance sampling is a simple modification to the Monte Carlo method for computing expectations that is useful when there is an auxiliary distribution $g(\theta)$ that is easier to directly sample from than the target distribution $p(\theta)$, which may be only known up to a proportionality constant. The starting point is the simple Monte Carlo estimate for the expectation of a function h,

$$I_h = \mathbb{E}_p(h) = \int h(\theta)p(\theta)\,d\theta \approx \frac{1}{S}\sum_{s=1}^{S} h(\theta_s),$$

which requires exact draws $\theta_s,\ s = 1,\ldots,S$ from $p(\theta)$. The importance sampling estimate for the same expectation is

$$\frac{\sum_{s=1}^{S} r_s h(\theta_s)}{\sum_{s=1}^{S} r_s}, \qquad r_s = \frac{f(\theta_s)}{g(\theta_s)}, \tag{2.1}$$

[1]This chapter is a shortened version of Vehtari et al. (2019b)

which only requires draws θ_s from a proposal distribution $g(\theta)$.

The success of the self-normalized importance sampling estimator depends on the distribution of importance ratios r_s. When the proposal distribution is a poor approximation to the target distribution, the distribution of importance ratios can have a heavy right tail. This can lead to unstable importance weighted estimates, sometimes with infinite variance.

The textbook examples of poorly performing importance samplers occur in low dimensions when the proposal distribution has lighter tails than the target, but it would be a mistake to assume that heavy tailed proposals will stabilize importance samplers. This intuition is particularly misplaced in high dimensions, where importance sampling can fail even when the ratios have finite variance. MacKay (2003, Section 29.2) provides an example of what goes wrong in high dimensions: essentially, the ratios r_s will vary by several orders of magnitude and the estimator (2.1) will be dominated by a few draws. Hence, even if the approximating distribution is chosen so that the importance ratios are bounded and thus (2.1) has finite variance, the bound can be so high and the variance so large that the behaviour of the self-normalized importance sampling estimator is practically indistinguishable from an estimator with infinite variance. This suggests that if we want an importance sampling method that works in high dimensions, we need to move beyond being satisfied with estimates that have finite variance and towards methods with built-in error assessment.

2.2 Stabilizing importance sampling estimates by modifying the ratios

The stability of self-normalized importance sampling methods can be improved by directly modifying the computed ratios. For notational convenience, we rewrite these importance samplers as,

$$\int h(\theta)p(\theta)\,d\theta \approx \frac{\sum_{s=1}^{S} w_s h(\theta_s)}{\sum_{s=1}^{S} w_s}, \tag{2.2}$$

where $w_s = r_s$ would recover the standard self-normalized importance sampler.

Ionides (2008) showed that the truncation rule

$$w_s = \min\left(r_s, \sqrt{S}\bar{r}\right), \tag{2.3}$$

where $\bar{r}$ is the average of the original S importance ratios, is consistent with finite variance and asymptotic normality. The critical advantage conveyed by the truncation is that these properties now extend to problems that only have integrable ratios; that is, we get asymptotic normality under the assumption that $\mathbb{E}(|r_s|) < \infty$ instead of under the stronger condition $\mathbb{E}(|r_s|^2) < \infty$ which is needed for the unmodified importance sampling estimator (2.1) to have finite variance.

This simple modification to the raw importance ratio greatly extends the range of problems for which the estimator has finite variance and asymptotic normality. Unfortunately, while the truncation can improve the stability of the weights, our experiments show that the simple weight modification scheme can be too severe, leading to larger than necessary finite sample bias.

Modeling the tail of the importance ratios. We propose a new scheme for modifying the extreme importance ratios that adapts more readily to the problem under consideration than the universal truncation rule of Ionides (2008).

To motivate the new scheme, we begin by noting that the success of plain importance sampling depends entirely on how many moments the importance ratios r_s possess, with the estimator having finite variance if the importance weights have finite variance. Chen and Shao (2004) and Koopman et al. (2009) show that when the importance ratios have more than two finite moments, the convergence rate of the estimators improves. This suggests that using information about the distribution of $r_s|r_s > u$, for some threshold $u \to \infty$ as $S \to \infty$, should allow us to improve the quality of our importance sampling estimators.

Pickands (1975) proved, under commonly satisfied conditions, that as sample size increases the

distribution of $r_s | r_s > u$ is well approximated by the three-parameter generalized Pareto distribution,

$$p(y|u,\sigma,k) = \begin{cases} \frac{1}{\sigma}\left(1 + k\left(\frac{y-u}{\sigma}\right)\right)^{-\frac{1}{k}-1}, & k \neq 0 \\ \frac{1}{\sigma}\exp\left(\frac{y-u}{\sigma}\right), & k = 0, \end{cases} \tag{2.4}$$

where u is a lower bound parameter, y is restricted to the range (u, ∞), σ is a non-negative scale parameter, and k is an unconstrained shape parameter. The generalized Pareto distribution has $\lfloor 1/k \rfloor$ finite moments when $k > 0$, and thus we can infer the number of existing moments of the weight distribution by focusing on k.

To estimate the parameters in the generalized Pareto distribution, we use the M largest importance ratios, where

$$M = \begin{cases} 3\sqrt{S}, & S > 225, \\ S/5, & S \leq 225. \end{cases}$$

Restricting the tail modeling to a subset of the largest importance ratios implicitly defines a value of u in the Pareto distribution. The above choice of M was made based on extensive computational theory and in line with the requirements for consistent estimation (Pickands, 1975). In practice, we have found that the method is mostly indifferent to the exact form of M; for instance, using $M = S/5$ in all cases was suggested by Vehtari et al. (2017) and it worked well even though it is not asymptotically correct. We tested several other methods (Scarrot and MacDonald, 2012) for selecting u directly, but found them to be more noisy than this simple heuristic.

The scale and shape parameters, σ and k, can be estimated using the highly efficient, low-bias method of Zhang and Stephens (2009). We chose this approach due to its efficiency and its ability to be used automatically without human intervention.

Occasionally, we will use a h-specific tail estimate $\hat{k}_h$ for the tail of the distribution $h(\theta)r(\theta)$, $\theta \sim g(\theta)$. This can be useful when $h(\theta)$ is unbounded or goes to zero, as it is possible if h grows fast enough (or goes to zero fast enough) relative to $r(\theta)$ that the tail behavior of $r(\theta)h(\theta)$ can be qualitatively different to the tail behavior of $r(\theta)$.

Algorithm 1: Pareto smoothed importance sampling(PSIS)

Input: Raw importance ratios r_s, $s = 1, \ldots, S$, ordered from highest to lowest
Output: PSIS-smoothed importance weights w_s, $s = 1, \ldots, S$

1 Set $M = 3\sqrt{S}$ if $S > 225$, or $S/5$ if $S \leq 225$;
2 Set $w_s = r_s$, $s = 1, \ldots, M - 1$;
3 Set $\hat{u} = r_M$;
4 Estimate $(\hat{k}, \hat{\sigma})$ in the generalized Pareto distribution from the M largest importance ratios, using the algorithm of Zhang and Stephens (2009);
5 Set $w_{M+z-1} = \min\left(F^{-1}\left(\frac{z-1/2}{M}\right), \max_s(r_s)\right)$, for each $z = 1, \ldots, M$;
6 If the estimated shape parameter $\hat{k}$ exceeds 0.7, report a warning that the resulting importance sampling estimates are likely to be unstable;

Although in most cases we cannot verify that the distribution of r_s lies in the domain of attraction for an extremal distribution, we will use this as a working assumption for building our method. Pickands (1975) notes that "most 'textbook' continuous distribution functions" (Gumbel, 1958) lie in the domain of attraction of some extremal distribution function. For finite S, we could alternatively interpret $\hat{k}$ as saying that the sample $(r(\theta_s))_{s=1}^S$ is not statistically distinguishable from a sample of size S from a distribution with tail index $\hat{k}$.

2.3 Pareto smoothed importance sampling (PSIS)

We propose a new method to stabilize the importance weights by replacing the M largest weights above the threshold u by a set of well-spaced values that are consistent with the tails of the importance distribution,

$$\begin{aligned} w_{M+z-1} &= F^{-1}\left(\frac{z-1/2}{M}\right) \\ &= r_{(S-M+1:S)} + k^{-1}\sigma\left[\left(1 - \frac{z-1/2}{M}\right)^{-k} - 1\right] \\ &:= r_{(S-M:S)} + \tilde{w}_z, \end{aligned}$$

where $z = 1, \ldots, M$, and F^{-1} is the inverse-CDF of the generalized Pareto distribution fitted to the M largest importance ratios.

The resulting Pareto smoothed importance sampler can be written as

$$I_h^S = \frac{1}{S}\sum_{s=1}^{S}\left(r(\theta_s) \wedge r_{(S-M+1):S}\right) h(\theta_s) + \frac{1}{S}\sum_{m=1}^{M}\tilde{w}_m h(\theta_{S-M+m}).$$

This facilitates the view of PSIS as a generalization of truncated importance sampling with a problem-adapted threshold and a bias correction term. This estimator, summarized in Algorithm 1, is asymptotically consistent and has finite, vanishing variance under relatively light conditions. The final step in the algorithm warns the user if the estimated smoothness parameter in the generalized Pareto distribution is larger than 0.7. This is an automatic stability check that we justify using a complexity analysis in Section 2.4.

We recommend estimating the Monte Carlo error by formally adapting the estimator provided by Owen (2013, Ch. 9) for self-normalized importance sampling,

$$\widehat{\mathrm{Var}}(\hat{I}_h^S) \approx \sum_{s=1}^{S} w_s^2 (h(\theta_s) - \tilde{\mu})^2.$$

If the draws have been obtained via MCMC, we recommend adjusting the above variance estimator to account for the autocorrelation in the sample,

$$\widehat{\mathrm{Var}}(\hat{I}_h^S) \approx \sum_{s=1}^{S} w_s^2 (h(\theta_s) - \tilde{\mu})^2 / R_{\mathrm{eff,MCMC}},$$

where $R_{\mathrm{eff,MCMC}} = S_{\mathrm{eff,MCMC}}/S$ is the relative efficiency of MCMC, and the effective sample size $S_{\mathrm{eff,MCMC}}$ for $h(\theta)$ is computed using the split-chain effective sample size estimate method (Vehtari et al., 2020a).

Vehtari et al. (2019b) further show that PSIS is asymptotically consistent and has finite variance under some mild, but complex conditions on the distribution of the ratios $r(\theta_s)$. The detailed theory discussion is suppressed from the current thesis for brevity.

2.4 Using $\hat{k}$ as a diagnostic of importance sampling

As part of the PSIS procedure, we estimate the shape parameter k of the limiting generalized Pareto distribution for the upper tail of the importance ratios. A simple option would be to use this estimate $\hat{k}$ and trust importance sampling as reliable if $\hat{k} \leq 0.5$. Our theory shows, however, that PSIS leads to valid and well-behaved importance sampling routines for any integrable density. This suggests that requiring the estimated shape parameter be less than 0.5 will be unnecessarily stringent.

Extensive experiments show that PSIS gives reliable (that is, low-bias, low variance) estimators of I_h when $\hat{k} < 0.7$. This can be interpreted as the estimate being reliable when the sample used to compute the PSIS estimate is indistinguishable from one that comes from a density with more than $0.7^{-1} = 1.43$ finite moments. When $\hat{k} > 0.7$ the convergence of the estimator becomes too slow to be useful.

In this section, we make the distinction between the true, but unknown, k and our finite sample estimate $\hat{k}$ as a way to understand the differences between the asymptotic behaviour of PSIS and its finite sample behaviour. We first show theoretically that the computational complexity of importance sampling has a meaningful qualitative change around $k = 0.7$. We then demonstrate that the finite sample estimate $\hat{k}$ is a useful indicator of the practical convergence of PSIS even when the distribution of ratios has finite variance. In particular, we show that $\hat{k}$ can identify poorly behaved but finite variance proposals in high dimensions.

PSIS is reliable when $k < 0.7$. Several authors (Sanz-Alonso, 2018; Agapiou et al., 2017; Chatterjee and Diaconis, 2018), have proved, under a variety of assumptions, that a minimum sample size to control the importance sampling error is roughly $\propto \exp\left(\mathrm{KL}(p||g)\right) = \exp\left(\int p(\theta) \log\left(\frac{g(\theta)}{p(\theta)}\right) d\theta\right)$. The most comprehensive results are due to Chatterjee and Diaconis (2018), who show that a larger sample size than this gives tail guarantees for the error in self-normalized importance sampling, while for smaller sample sizes it is not possible to control the large deviations. We can turn this theory into a heuristic bound by assuming that the density ratios $r(\theta) = p(\theta)/g(\theta)$ exactly form a

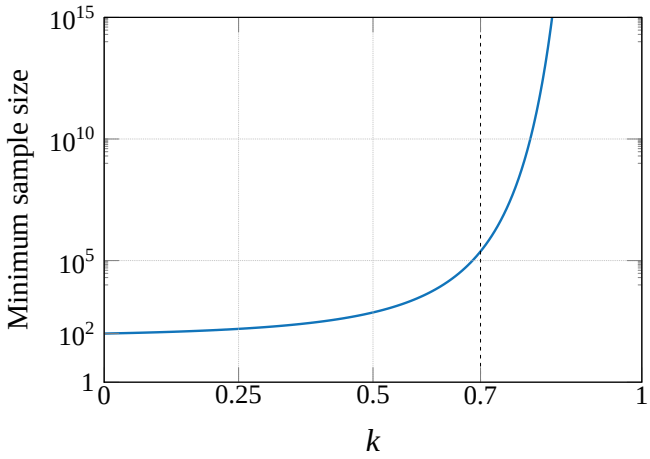


Figure 2.1: *Minimum sample size required as a function of k, as computed according to the heuristic* (2.5). *The required sample size quickly grows infeasibly large when* $k > 0.7$.

Pareto distribution with shape parameter $k \in (0, 1)$ under g, in which case we can estimate $\mathrm{KL}(p||g)$ by $\int_1^\infty r \log r \frac{1}{k} r^{-1/k-1} dr = k(1-k)^{-2}$. Hence, the minimal sample size,

$$S = O\left(\exp\left(k(1-k)^{-2}\right)\right). \tag{2.5}$$

Figure 2.1 shows the estimated sample size as a function as k using the approximation (2.5). Although we do not recommend using this estimator directly due to both the uncertainty in estimating k and the inaccuracy of the asymptotic approximation, we can see the quick increase in the order of magnitude for the minimum sample size around $k = 0.7$.

Although PSIS gives consistent, finite variance estimates when $k \in (0.7, 1)$, these complexity considerations suggest that they will not be practical in this regime. Simple experiments suggest that this type of behavior also holds for TIS, suggesting that the $k \approx 0.7$ may be a fundamental complexity barrier for useful importance sampling estimators.

$\hat{k}$ is a good convergence diagnostic in finite samples and in high dimensions. Although importance sampling papers classically focus on the asymptotic properties of the estimator, the literature is full of examples where an importance sampling estimator is consistent, has finite variance, and is asymptotically normal but still fails to work. Most of these examples, such as

the one in MacKay (2003, Sec. 29.2) where the importance ratios are bounded, occur in high dimensions. Hence, if we want PSIS to work reliably for problems of any dimension, we need to have a diagnostic that can flag poor convergence for any given sample of importance weights.

In the previous section, we argued that if we know the tail behaviour of the importance ratios, we can tell if PSIS will succeed within a reasonable computational budget. In this section, we argue that the estimate $\hat{k}$ of the true tail index can quantify the finite sample behaviour. It does not matter what the actual tail behaviour is if the observed importance ratios "look" heavy tailed. In particular, we suggest the heuristic that if $\hat{k} > 0.7$, then the PSIS estimator will be unreliable.

2.5 Application: leave-one-out cross-validation by importance sampling

In this section we present three examples where Pareto smoothed importance sampling improves the estimates and where the Pareto shape estimate $\hat{k}$ is a useful diagnostic. We demonstrate the use of Pareto smoothed importance sampling for leave-one-out (LOO) cross-validation approximation. The ith leave-one-out cross-validation predictive density can be approximated with

$$p(\tilde{y}_i|y_{-i}) \approx \frac{\sum_{s=1}^{S} w_i(\theta^s) p(\tilde{y}_i|\theta^s)}{\sum_{s=1}^{S} w_i(\theta^s)}. \tag{2.6}$$

Importance sampling LOO was proposed by Gelfand et al. (1992a), but for long time it was not widely used as the estimator is unreliable if the weights have infinite variance. For some simple models, such as linear and generalized linear models with specific priors, it is possible to analytically check the sufficient conditions for the variance of the importance weights in IS-LOO to be finite (Peruggia, 1997; Epifani et al., 2008), but this is not generally possible.

Vehtari et al. (2017) demonstraed

2.6 Discussion

Importance weighting is a widely used tool in statistical computation. Even in the modern era of Markov chain Monte Carlo, approximate algorithms are often necessary, in which case we should

adjust approximations when possible to better match target distributions. However, a challenge for practical applications of importance weighting is the well known fact that importance-weighted estimates are unstable if the weights have high variance.

In this chapter we have shown that it is possible to reduce the mean square error of importance sampling estimates using a particular stabilizing transformation that we call Pareto smoothed importance sampling (PSIS). The key step is to replace the largest weights by expected quantiles from a generalized Pareto distribution. We have also demonstrated greatly improved Monte Carlo error estimates, natural diagnostics for gauging the reliability of the estimates, and empirical convergence rate results that closely follow known theoretical results.

The most important feature of PSIS is the $\hat{k}$ diagnostic, which allows the user to assess the reliability of the estimator:

- If $k < \frac{1}{3}$ the Berry-Esseen theorem (Chen and Shao, 2004; Koopman et al., 2009) states faster convergence rate to normality. If $\hat{k} < \frac{1}{3}$, we observe that importance sampling is stable and all IS, TIS and PSIS work well.

- If $k < \frac{1}{2}$ then the distribution of importance ratios has finite variance and the central limit theorem holds (Geweke, 1989). However the finite sample bias may still be quite large and the value of $\hat{k}$ may be much larger than $1/2$ indicating that the importance sampler is unreliable for this realization of weights.

- If $\frac{1}{2} \leq k < 0.7$ then the variance is infinite and plain IS can behave quite poorly. However both TIS and PSIS work well in this regime.

- If $0.7 < k < 1$ it quickly becomes too expensive to get an accurate estimate. We do not recommend using importance sampling when $\hat{k} > 0.7$.

- If $k \geq 1$ then neither the variance nor the mean of raw ratios exists. The convergence rate is close to zero and bias can be large with practical sample sizes S.

With these recommendations, PSIS is a reliable, accurate, and trustworthy variant of important sampling that comes with a in-built heuristic that allows it to fail loudly when it becomes unreliable.

Chapter 3. Assessing variational inference using posterior density ratios[1]

"You may hide yourself in a thousand forms,
Still, All-beloved, I recognize you.
You may cover yourself in magic mists,
All-present, I can always tell that it is you."

—Johann Wolfgang von Goethe

3.1 Yes, but did it work?

Variational Inference (VI), including a large family of posterior approximation methods like stochastic VI (Hoffman et al. 2013), black-box VI (Ranganath et al. 2014), automatic differentiation VI (ADVI, Kucukelbir et al. 2017), and many other variants, has emerged as a widely-used method for scalable Bayesian inference. These methods come with few theoretical guarantees and it's difficult to assess how well the computed variational posterior approximates the true posterior.

Instead of computing expectations or sampling draws from the posterior $p(\theta \mid y)$, variational inference fixes a family of approximate densities $\mathcal{Q}$, and finds the member q^* minimizing the Kullback-Leibler (KL) divergence to the true posterior: $\mathrm{KL}\left(q(\theta), p(\theta \mid y)\right)$. This is equivalent to maximizing the evidence lower bound (ELBO):

$$\mathrm{ELBO}(q) = \int_\Theta \left(\log p(\theta, y) - \log q(\theta)\right) q(\theta) d\theta. \tag{3.1}$$

There are many situations where the VI approximation is flawed. This can be due to the slow convergence of the optimization problem, the inability of the approximation family to capture the true posterior, the asymmetry of the true distribution, the fact that the direction of the KL divergence under-penalizes approximation with too-light tails, or all these reasons. We need a diagnostic algorithm to test whether the VI approximation is useful.

[1]This chapter is a slight modified version of Yao et al. (2018b).

There are two levels of diagnostics for variational inference. First the convergence test should be able to tell if the objective function has converged to a local optimum. When the optimization problem (3.1) is solved through stochastic gradient descent (SGD), the convergence can be assessed by monitoring the running average of ELBO changes. Researchers have introduced many convergence tests based on the asymptotic property of stochastic approximations (e.g., Sielken, 1973; Stroup and Braun, 1982; Pflug, 1990; Wada and Fujisaki, 2015; Chee and Toulis, 2018). Alternatively, Blei et al. (2017) suggest monitoring the expected log predictive density by holding out an independent test dataset. After convergence, the optimum is still an approximation to the truth. This chapter is focusing on the second level of VI diagnostics whether the variational posterior $q^*(\theta)$ is close enough to the true posterior $p(\theta|y)$ to be used in its place.

Purely relying on the objective function or the equivalent ELBO does not solve the problem. An unknown multiplicative constant exists in $p(\theta, y) \propto p(\theta \mid y)$ that changes with reparametrization, making it meaningless to compare ELBO across two approximations. Moreover, the ELBO is a quantity on an uninterpretable scale, that is it's not clear at what value of the ELBO we can begin to trust the variational posterior. This makes it next to useless as a method to assess how well the variational inference has fit.

In this chapter we propose two diagnostic methods that assess, respectively, the quality of the entire variational posterior for a particular data set, and the average bias of a point estimate produced under correct model specification.

The first method is based on generalized Pareto distribution diagnostics used to assess the quality of a importance sampling proposal distribution in Pareto smoothed importance sampling (PSIS, Vehtari et al., 2019b). The benefit of PSIS diagnostics is two-fold. First, we can tell the discrepancy between the approximate and the true distribution by the estimated continuous $\hat{k}$ value. When it is larger than a pre-specified threshold, users should be alert of the limitation of current variational inference computation and consider further tuning it or turn to exact sampling like Markov chain Monte Carlo (MCMC). Second, in the case when $\hat{k}$ is small, the fast convergence rate of the importance-weighted Monte Carlo integration guarantees a better estimation accuracy. In such

sense, the PSIS diagnostics could also be viewed as a post-adjustment for VI approximations. Unlike the second-order correction Giordano et al. (2018), which relies on an un-testable unbiasedness assumption, we make diagnostics and adjustment at the same time.

The second diagnostic considers only the quality of the median of the variational posterior as a point estimate (in Gaussian mean-field VI this corresponds to the modal estimate). This diagnostic assesses the average behavior of the point estimate under data from the model and can indicate when a systemic bias is present. The magnitude of that bias can be monitored while computing the diagnostic. This diagnostic can also assess the average calibration of univariate functionals of the parameters, revealing if the posterior is under-dispersed, over-dispersed, or biased. This diagnostic could be used as a partial justification for using the second-order correction of Giordano et al. (2018).

3.2 Is the jointd distribution good enough: PSIS diagnostic

If we can draw a sample $(\theta_1, \ldots, \theta_S)$ from $p(\theta|y)$, the expectation of any integrable function $\mathbb{E}_p[h(\theta)]$ can be estimated by Monte Carlo integration: $\sum_{s=1}^{S} h(\theta_s)/S \xrightarrow{S\to\infty} \mathbb{E}_p\left[h(\theta)\right]$. Alternatively, given samples $(\theta_1, \ldots, \theta_S)$ from a proposal distribution $q(\theta)$, the *importance sampling* (IS) estimate is $\left(\sum_{s=1}^{S} h(\theta_s) r_s\right) / \sum_{s=1}^{S} r_s$, where the importance ratios r_s are defined as

$$r_s = \frac{p(\theta_s, y)}{q(\theta_s)}. \tag{3.2}$$

In general, with a sample $(\theta_1, \ldots, \theta_S)$ drawn from the variational posterior $q(\theta)$, we consider a family of estimates with the form

$$E_p[h(\theta)] \approx \frac{\sum_{s=1}^{S} h(\theta_s) w_s}{\sum_{s=1}^{S} w_s}, \tag{3.3}$$

which contains two extreme cases:

1. When $w_s \equiv 1$, estimate (3.3) becomes the plain VI estimate that is we completely trust the

VI approximation. In general, this will be biased to an unknown extent and inconsistent. However, this estimator has small variance.

2. When $w_s = r_s$, (3.3) becomes *importance sampling*. The strong law of large numbers ensures it is consistent as $S \to \infty$, and with small $O(1/S)$ bias due to self-normalization. But the IS estimate may have a large or infinite variance.

There are two questions to be answered. First, can we find a better bias-variance trade-off than both plain VI and IS?

Second, VI approximation $q(\theta)$ is not designed for an optimal IS proposal, for it has a lighter tail than $p(\theta|y)$ as a result of entropy penalization, which lead to a heavy right tail of r_s. A few large-valued r_s dominates the summation, bringing in large uncertainty. But does the finite sample performance of IS or stabilized IS contain the information about the dispensary measure between $q(\theta)$ and $p(\theta|y)$?

Here we propose to answer both questions by Pareto smoothed importance sampling (PSIS, Chapter 2). The fitted shape parameter $\hat{k}$, turns out to provide the desired diagnostic measurement between the true posterior $p(\theta|y)$ and the VI approximation $q(\theta)$. A generalized Pareto distribution with shape k has finite moments up to order $1/k$, thus any positive $\hat{k}$ value can be viewed as an estimate to

$$k = \inf \left\{ k' > 0 : E_q \left(\frac{p(\theta|y)}{q(\theta)} \right)^{\frac{1}{k'}} < \infty \right\}. \tag{3.4}$$

$\hat{k}$ is invariant under any constant multiplication of p or q, which explains why we can suppress the marginal likelihood (normalizing constant) $p(y)$ and replace the intractable $p(\theta|y)$ with $p(\theta, y)$ in (3.2).

After log transformation, (3.4) can be interpreted as *Rényi divergence* (Rényi, 1961) with order

α between $p(\theta|y)$ and $q(\theta)$:

$$k = \inf\left\{k' > 0 : D_{\frac{1}{k'}}(p||q) < \infty\right\},$$
$$\text{where } D_\alpha(p||q) = \frac{1}{\alpha - 1}\log\int_\Theta p(\theta)^\alpha q(\theta)^{1-\alpha} d\theta.$$

It is well-defined since Rényi divergence is monotonic increasing on order α. Particularly, when $k > 0.5$, the χ^2 *divergence* $\chi(p||q)$, becomes infinite, and when $k > 1$, $D_1(p||q) = \mathrm{KL}(p, q) = \infty$, indicating a disastrous VI approximation, despite the fact that $\mathrm{KL}(q, p)$ is always minimized among the variational family. The connection to Rényi divergence holds when $k > 0$. When $k < 0$, it predicts the importance ratios are bounded from above.

This also illustrates the advantage of a continuous $\hat{k}$ estimate in our approach over only testing the existence of second moment of $\mathbb{E}_q(q/p)^2$ (Epifani et al., 2008; Koopman et al., 2009) – it indicates if the Rényi divergence between q and p is finite for all continuous order $\alpha > 0$.

Meanwhile, the shape parameter k determines the finite sample convergence rate of both IS and PSIS adjusted estimate. Geweke (1989) shows when $\mathbb{E}_q[r(\theta)^2] < \infty$ and $\mathbb{E}_q[\big(r(\theta)h(\theta)\big)^2] < \infty$ hold (both conditions can be tested by $\hat{k}$ in our approach), the central limit theorem guarantees the square root convergence rate. Furthermore, when $k < 1/3$, then the Berry-Essen theorem states faster convergence rate to normality (Chen and Shao, 2004). Cortes et al. (2010) and Cortes et al. (2019) also link the finite sample convergence rate of IS with the number of existing moments of importance ratios.

PSIS has smaller estimation error than the plain VI estimate, which we will experimentally verify this in Section 3.5. A large $\hat{k}$ indicates the failure of finite sample PSIS, so it further indicates the large estimation error of VI approximation. Therefore, even when the researchers' primary goal is not to use variational approximation q as an PSIS proposal, they should be alert by a large $\hat{k}$ which tells the discrepancy between the VI approximation result and the true posterior.

According to empirical study in Vehtari et al. (2019b), we set the threshold of $\hat{k}$ as follows.

- If $\hat{k} < 0.5$, we can invoke the central limit theorem to suggest PSIS has a fast convergence

rate. We conclude the variational approximation q is close enough to the true density. We recommend further using PSIS to adjust the estimator (3.3) and calculate other divergence measures.

- If $0.5 < \hat{k} < 0.7$, we still observe practically useful finite sample convergence rates and acceptable Monte Carlo error for PSIS. It indicates the variational approximation q is not perfect but still useful. Again, we recommend PSIS to shrink errors.

- If $\hat{k} > 0.7$, the PSIS convergence rate becomes impractically slow, leading to a large mean square error, and a even larger error for plain VI estimate. We should consider tuning the variational methods (e.g., re-parametrization, increase iteration times, increase mini-batch size, decrease learning rate, et.al.,) or turning to exact MCMC. Theoretically k is always smaller than 1, for $E_q\left[p(\theta|y)/q(\theta)\right] = p(y) < \infty$, while in practice finite sample estimate $\hat{k}$ may be larger than 1, which indicates even worse finite sample performance.

The proposed diagnostic method is summarized in Algorithm 2.

Algorithm 2: PSIS diagnostic for variational inference

Input: the joint density function $p(\theta, y)$; number of posterior samples S; number of tail samples M.

Run variational inference to $p(\theta|y)$, obtain VI approximation $q(\theta)$;
Sample $(\theta_s, s = 1, \ldots, S)$ from $q(\theta)$;
Calculate the importance ratio $r_s = p(\theta_s, y)/q(\theta_s)$;
Fit generalized Pareto distribution to the M largest r_s;
Report the shape parameter $\hat{k}$;
if $\hat{k} < 0.7$ **then**
    Conclude VI approximation $q(\theta)$ is close enough to the unknown truth $p(\theta|y)$;
    Recommend further shrinking errors by PSIS.
else
    Warn users that the VI approximation is not reliable.
end

3.3 Properties of the PSIS diagnostics

Invariance under reparametrization. Re-parametrization is common in variational inference. Particularly, the *reparameterization trick* (Rezende et al., 2014) rewrites the objective function to make gradient calculation easier in Monte Carlo integrations.

A nice property of PSIS diagnostics is that the $\hat{k}$ quantity is invariant under any re-parametrization. Suppose $\xi = T(\theta)$ is a smooth transformation, then the density ratio of ξ under the target p and the proposal q does not change:

$$\frac{p(\xi)}{q(\xi)} = \frac{p\left(T^{-1}(\xi)\right)\left|\det J_\xi T^{-1}(\xi)\right|}{q\left(T^{-1}(\xi)\right)\left|\det J_\xi T^{-1}(\xi)\right|} = \frac{p\left(\theta\right)}{q(\theta)}$$

Therefore, $p(\xi)/q(\xi)$ and $p(\theta)/q(\theta)$ have the same distribution under q, making it free to choose any convenient parametrization form when calculating $\hat{k}$.

However, if the re-parametrization changes the approximation family, then it will change the computation result, and PSIS diagnostics will change accordingly. Finding the optimal parametrization form, such that the re-parametrized posterior distribution lives exactly in the approximation family

$$p(T(\xi)) = p\left(T^{-1}(\xi)\right)\left|J_\xi T^{-1}(\xi)\right| \in \mathcal{Q},$$

can be as hard as finding the true posterior. The PSIS diagnostic can guide the choice of re-parametrization by simply comparing the $\hat{k}$ quantities of any parametrization. Section 3.5 provides a practical example.

Marginal PSIS diagnostics do not work. As dimension increases, the VI posterior tends to be further away from the truth, due to the limitation of approximation families. As a result, k increases, indicating inefficiency of importance sampling. This is not the drawback of PSIS diagnostics. Indeed, when the focus is the joint distribution, such behaviour accurately reflects the quality of the variational approximation to the joint posterior.

Denoting the one-dimensional true and approximate marginal density of the i-th coordinate θ_i

as $p(\theta_i|y)$ and $q(\theta_i)$, the marginal k for θ_i can be defined as

$$k_i = \inf \left\{ 0 < k' < 1 : \mathbb{E}_q \left(\frac{p(\theta_i|y)}{q(\theta_i)} \right)^{\frac{1}{k'}} < \infty \right\}.$$

The marginal k_i is never larger (and usually smaller) than the joint k in (3.4).

Theorem 3.1 (monotonicity of PSIS diagnostics)**.** *For any two distributions p and q with support* Θ *and the margin index i, if there is a number* $\alpha > 1$ *satisfying* $E_q\left(p(\theta)/q(\theta)\right)^\alpha < \infty$*, then* $E_q\left(p(\theta_i)/q(\theta_i)\right)^\alpha < \infty$.

Theorem 3.1 demonstrates why the importance sampling is usually inefficient in high dimensional sample space, in that the joint estimation is "worse" than any of the marginal estimation.

Should we extend the PSIS diagnostics to marginal distributions? We find two reasons why the marginal PSIS diagnostics can be misleading. Firstly, unlike the easy access to the unnormalized joint posterior distribution $p(\theta, y)$, the true marginal posterior density $p(\theta_i|y)$ is typically unknown, otherwise one can conduct one-dimensional sampling easily to obtain the the marginal samples. Secondly, a smaller $\hat{k}_i$ does not necessary guarantee a well-performed marginal estimation. The marginal approximations in variational inference can both over-estimate and under-estimate the tail thickness of one-dimensional distributions, the latter situation gives rise to a smaller $\hat{k}_i$. Section 3.5 gives an example, where the marginal approximations with extremely small marginal k have large estimation errors. This does not happen in the joint case as the direction of the Kullback-Leibler divergence $q^*(\theta)$ strongly penalizes too-heavy tails, which makes it unlikely that the tails of the variational posterior are significantly heavier than the tails of the true posterior.

3.4 Variational simulation-based calibration (VSBC)

The proposed PSIS diagnostic assesses the quality of the VI approximation to the full posterior distribution. It is often observed that while the VI posterior may be a poor approximation to the full posterior, point estimates that are derived from it may still have good statistical properties. In this section, we propose a new method for assessing the calibration of the center of a VI posterior.

This diagnostic is based on the proposal of Cook et al. (2006) for validating general statistical software. They noted that if $\theta^{(0)} \sim p(\theta)$ and $y \sim p(y \mid \theta^{(0)})$, then

$$\Pr_{(y,\theta^{(0)})}\left(\Pr_{\theta|y}(\theta < \theta^{(0)}) \leq \cdot)\right) = \text{Unif}_{[0,1]}([0, \cdot]).$$

To use the observation of Cook et al. (2006) to assess the performance of a VI point estimate, we propose the following procedure. Simulate $M > 1$ data sets $\{y_j\}_{j=1}^{M}$ as follows: Simulate $\theta_j^{(0)} \sim p(\theta)$ and then simulate $y_{(j)} \sim p(y \mid \theta_j^{(0)})$, where $y_{(j)}$ has the same dimension as y. For each of these data sets, construct a variational approximation to $p(\theta \mid y_j)$ and compute the marginal calibration probabilities $p_{ij} = \Pr_{\theta|y_{(j)}}\left(\theta_i \leq [\theta_j^{(0)}]_i\right)$.

To apply the full procedure of Cook et al. (2006), we would need to test $\dim(\theta)$ histograms for uniformity, however this would be too stringent a check as, like our PSIS diagnostic, this test is only passed if the variational posterior is a good approximation to the true posterior. Instead, we follow an observation of Anderson (1996) from the probabilistic forecasting validation literature and note that asymmetry in the histogram for $p_{i:}$ indicates bias in the variational approximation to the marginal posterior $\theta_i \mid y$.

The VSBC diagnostic tests for symmetry of the marginal calibration probabilities around 0.5 and either by visual inspection of the histogram of $p_{i:}$ and $1 - p_{i:}$ to check if they have the same distribution. When θ is a high-dimensional parameter, it is important to interpret the results of any hypothesis tests through a multiple testing lens.

In the early published version (Yao et al., 2018b), on top of a visual check, we also mentioned the use of use of Kolmogorov-Smirnov (KS) test to $p_{i:}$ and $1 - p_{i:}$. This additional test, however, can be highly biased because the dependence between p and $1 - p$. Hence, *we no longer recommend this hypothesis test step anymore.*

Unlike the PSIS diagnostic, which focuses on a the performance of variational inference for a fixed data set y, the VSBC diagnostic assesses the *average* calibration of the point estimation over all datasets that could be constructed from the model. Hence, the VSBC diagnostic operates under

a different paradigm to the PSIS diagnostic and we recommend using both as appropriate.

There are two disadvantages to this type of calibration when compared to the PSIS diagnostic. As is always the case when interpreting hypothesis tests, just because something works on average doesn't mean it will work for a particular realization of the data. The second disadvantage is that this diagnostic does not cover the case where the observed data is not well represented by the model. We suggest interpreting the diagnostic conservatively: if a variational inference scheme fails the diagnostic, then it will not perform well on the model in question. If the VI scheme passes the diagnostic, it is not guaranteed that it will perform well for real data, although if the model is well specified it should do well.

The VSBC diagnostic has some advantages compared to the PSIS diagnostic. It is well understood that, for complex models, the VI posterior can be used to produce a good point estimate even when it is far from the true posterior. In this case, the PSIS diagnostic will most likely

Algorithm 3: Variational simulated based calibration (VSBC)

Input: prior density $p(\theta)$, data likelihood $p(y \mid \theta)$; number of replications M; parameter dimensions K.

for $j = 1 : M$ **do**
  Generate $\theta_j^{(0)}$ from prior $p(\theta)$;
  Generate a size-n dataset $\left(y_{(j)}\right)$ from $p(y \mid \theta_j^{(0)})$;
  Run variational inference using dataset $y_{(j)}$, obtain a VI approximation distribution $q_j(\cdot)$;
  for $i = 1 : K$ **do**
    Label $\theta_{ij}^{(0)}$ as the i-th marginal component of $\theta_j^{(0)}$; Label θ_i^* as the i-th marginal component of θ^*;
    Calculate $p_{ij} = \Pr(\theta_{ij}^{(0)} < \theta_i^* \mid \theta^* \sim q_j)$;
  end
end
for $i = 1 : K$ **do**
  Test if the distribution of $\left\{p_{ij}\right\}_{j=1}^{M}$ is symmetric;
  if *rejected* **then**
    Conclude the VI approximation is biased in its i-th margin.
  end
end

indicate failure. The second advantage is that unlike the PSIS diagnostic, the VSBC diagnostic considers one-dimensional marginals θ_i (or any functional $h(\theta)$), which allows for a more targeted interrogation of the fitting procedure.

With stronger assumptions, The VSBC test can be formalized as in Theorem 3.2.

Theorem 3.2 (variational simulated based calibration)**.** *Denote θ as a one-dimensional parameter that is of interest. Suppose in addition we have: (i) the VI approximation q is symmetric; (ii) the true posterior $p(\theta|y)$ is symmetric. If the VI estimation q is unbiased, i.e., $\mathbb{E}_{\theta\sim q(\theta|y)}\,\theta = \mathbb{E}_{\theta\sim p(\theta|y)}\,\theta$, then the distribution of VSBC p-value is symmetric. Otherwise, if the VI estimation is positively/negatively biased, then the distribution of VSBC p-value is right/left skewed.*

The symmetry of the true posterior is a stronger assumption than is needed in practice for this result to hold. In the forecast evaluation literature, as well as the literature on posterior predictive checks, the symmetry of the histogram is a commonly used heuristic to assess the potential bias of the distribution. In our tests, we have seen the same thing occurs: the median of the variational posterior is close to the median of the true posterior when the VSBC histogram is symmetric. We suggest again that this test be interpreted conservatively: if the histogram is not symmetric, then the VI is unlikely to have produced a point estimate close to the median of the true posterior.

3.5 Examples

Both PSIS and VSBC diagnostics are applicable to any variational inference algorithm. Without loss of generality, we implement mean-field Gaussian automatic differentiation variational inference (ADVI) in this section.

Linear regression. Consider a Bayesian linear regression $y \sim \text{normal}(X\beta, \sigma^2)$ with prior $\{\beta_i\}_{i=1}^K \sim \text{normal}(0,1), \sigma \sim \text{gamma}(.5,.5)$. We fix sample size $n = 10000$ and number of regressors $K = 100$.

Figure 3.1 visualizes the VSBC diagnostic, showing the distribution of VSBC p-values of the

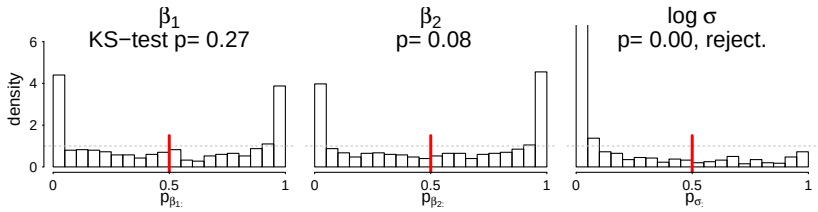


Figure 3.1: *VSBC diagnostics for β_1, β_2 and $\log \sigma$ in the Bayesian linear regression example. The VI estimation overestimates σ as p_σ is right-skewed, while β_1 and β_2 is unbiased as the two-sided KS-test is not rejected.*

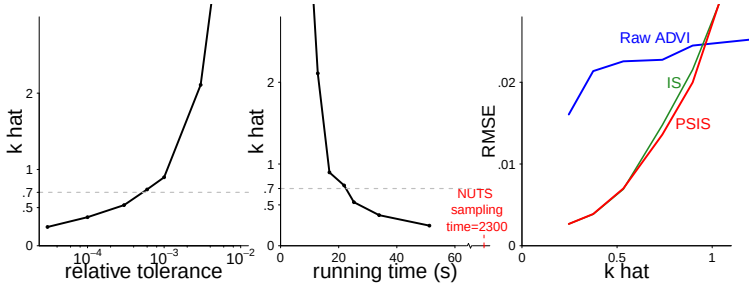


Figure 3.2: *ADVI is sensitive to the stopping time in the linear regression example. The default 0.01 threshold lead to a fake convergence, which can be diagnosed by monitoring PSIS $\hat{k}$. PSIS adjustment always shrinks the estimation errors.*

first two regression coefficients β_1, β_2 and $\log \sigma$ based on $M = 1000$ replications. The underestimation of posterior variance is reflected by the U-shaped distributions.

Using one randomly generated dataset in the same problem, the PSIS $\hat{k}$ is 0.61, indicating the joint approximation is close to the true posterior. However, the performance of ADVI is sensitive to the stopping time, as in any other optimization problems. As displayed in the left panel of Figure 3.2, changing the threshold of relative ELBO change from a conservative 10^{-5} to the default recommendation 10^{-2} increases $\hat{k}$ to 4.4, even though 10^{-2} works fine for many other simpler problems. In this example, we can also view $\hat{k}$ as a convergence test. The right panel shows $\hat{k}$ diagnoses estimation error, which eventually become negligible in PSIS adjustment when $\hat{k} < 0.7$. To account for the uncertainty of stochastic optimization and $\hat{k}$ estimation, simulations are repeated 100 times.

Logistic regression. Next we run ADVI to a logistic regression $Y \sim \text{Bernoulli}\left(\text{logit}^{-1}(\beta X)\right)$ with a flat prior on β. We generate $X = (x_1, \ldots, x_n)$ from $\text{N}(0, (1-\rho)I_{K\times K} + \rho 1_{K\times K})$ such that the correlation in design matrix is ρ, and ρ is changed from 0 to 0.99. The first panel in Figure 3.3

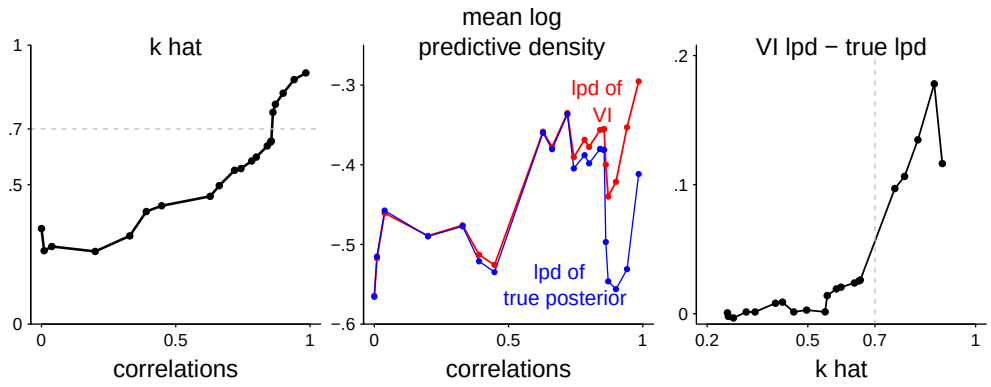


Figure 3.3: *In the logistic regression example, as the correlation in design matrix increase, the correlation in parameter space also increases, leading to larger* $\hat{k}$*. Such flaw is hard to tell from the VI log predictive density (lpd), as a larger correlation makes the prediction easier.* $\hat{k}$ *diagnose the discrepancy of VI lpd and true posterior lpd, with a sharp jump at* 0.7.

shows PSIS $\hat{k}$ increases as the design matrix correlation increases. It is not monotonic because β is initially negatively correlated when X is independent. A large ρ transforms into a large correlation for posterior distributions in β, making it harder to be approximated by a mean-field family, as can be diagnosed by $\hat{k}$. In panel 2 we calculate mean log predictive density (lpd) of VI approximation and true posterior using 200 independent test sets. Larger ρ leads to worse mean-field approximation, while prediction becomes easier. Consequently, monitoring lpd does not diagnose the VI behavior; it increases (misleadingly suggesting better fit) as ρ increases. In this special case, VI has larger lpd than the true posterior, due to the VI under-dispersion and the model misspecification. Indeed, if viewing lpd as a function $h(\beta)$, it is the discrepancy between VI lpd and true lpd that reveals the VI performance, which can also be diagnosed by $\hat{k}$. Panel 3 shows a sharp increase of lpd discrepancy around $\hat{k} = 0.7$, consistent with the empirical threshold we suggest.

Figure 3.4 compares the first and second moment root mean square errors (RMSE) $||\mathbb{E}_p \beta - \mathbb{E}_{q^*} \beta||_2$ and $||\mathbb{E}_p \beta^2 - \mathbb{E}_{q^*} \beta^2||_2$ in the previous example using three estimates: (a) VI without post-adjustment, (b) VI adjusted by vanilla importance sampling, and (c) VI adjusted by PSIS.

PSIS diagnostic accomplishes two tasks here: (1) A small $\hat{k}$ indicates that VI approximation is reliable. When $\hat{k} > 0.7$, all estimations are no longer reasonable so the user should be alerted. (2) It further improves the approximation using PSIS adjustment, leading to a quicker convergence rate and smaller mean square errors for both first and second moment estimation. Plain importance

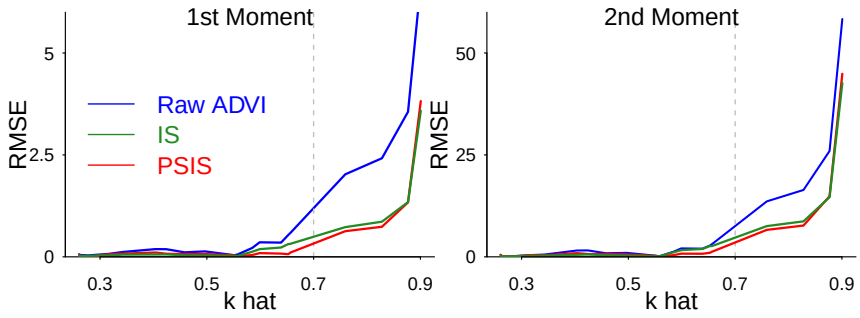


Figure 3.4: *In the logistic regression with varying correlations, the $\hat{k}$ diagnoses the root mean square of first and second moment errors. No estimation is reliable when $\hat{k} > 0.7$. Meanwhile, PSIS adjustment always shrinks the VI estimation errors.*

sampling has larger RMSE for it suffers from a larger variance.

Reparametrization in a hierarchical model. The *Eight-School Model* (Gelman et al., 2013, Section 5.5) is the simplest Bayesian hierarchical normal model. Each school reported the treatment effect mean y_i and standard deviation σ_i separately. There was no prior reason to believe that any of the treatments were more effective than any other, so we model them as independent experiments:

$$y_j|\theta_j \sim \mathrm{N}(\theta_j, \sigma_j^2), \quad \theta_j|\mu,\tau \sim \mathrm{N}(\mu,\tau^2), \quad 1 \le j \le 8,$$
$$\mu \sim \text{normal}(0,5), \qquad \tau \sim \text{half–Cauchy}(0,5).$$

where θ_j represents the treatment effect in school j, and μ and τ are the hyper-parameters shared across all schools.

In this hierarchical model, the conditional variance of θ is strongly dependent on the standard deviation τ, as shown by the joint sample of μ and $\log\tau$ in the bottom-left corner in Figure 3.5. The Gaussian assumption in ADVI cannot capture such structure. More interestingly, ADVI overestimates the posterior variance for all parameters θ_1 through θ_8, as shown by positive biases of their posterior standard deviation in the last panel. In fact, the posterior mode is at $\tau = 0$, while the entropy penalization keeps VI estimation away from it, leading to an overestimation due to the funnel-shape. Since the conditional expectation $\mathbb{E}[\theta_i|\tau,y,\sigma] = \left(\sigma_j^{-2} + \tau^{-2}\right)^{-1}$ is an increasing

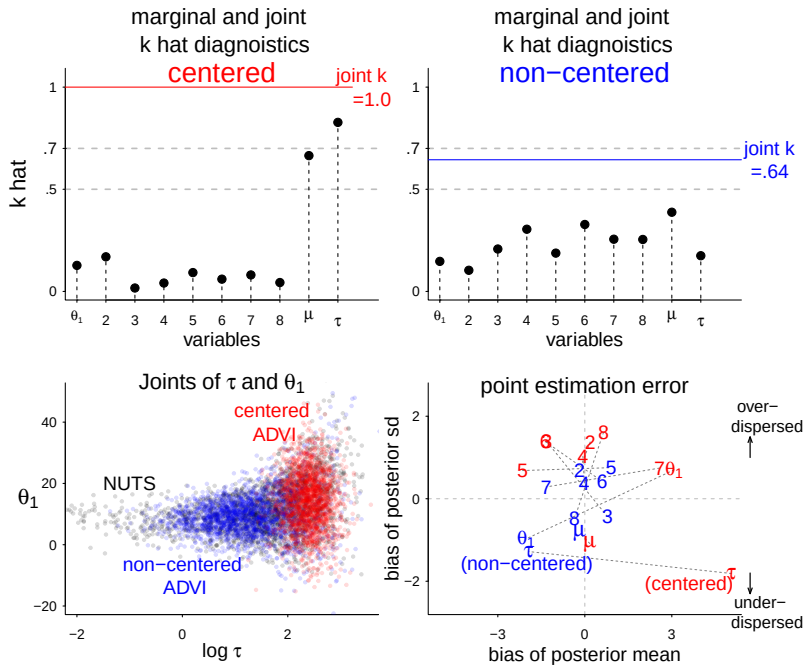


Figure 3.5: *The upper two panels shows the joint and marginal PSIS diagnostics of the eight-school example. The centered parameterization has* $\hat{k} > 0.7$*, for it cannot capture the funnel-shaped dependency between* τ *and* θ*. The bottom-right panel shows the bias of posterior mean and standard errors of marginal distributions. Positive bias of* τ *leads to over-dispersion of* θ*.*

function on τ, a positive bias of τ produces over-dispersion of θ.

The top left panel shows the marginal and joint PSIS diagnostics. The joint $\hat{k}$ is 1.00, much beyond the threshold, while the marginal $\hat{k}$ calculated through the true marginal distribution for all θ are misleadingly small due to the over-dispersion.

Alerted by such large $\hat{k}$, researchers should seek some improvements, such as re-parametrization. The *non-centered parametrization* extracts the dependency between θ and τ through a transformation $\theta^* = (\theta - \mu)/\tau$:

$$y_j|\theta_j \sim \mathrm{N}(\mu + \tau\theta_j^*, \sigma_j^2), \quad \theta_j^* \sim \mathrm{N}(0, 1).$$

There is no general rule to determine whether non-centered parametrization is better than the centered one and there are many other parametrization forms. Finding the optimal parametrization can be as hard as finding the true posterior, but $\hat{k}$ diagnostics always guide the choice of parametrization. As shown by the top right panel in Figure 3.5, the joint $\hat{k}$ for the non-centered ADVI decreases to 0.64 which indicated the approximation is not perfect but reasonable and usable. The bottom-right

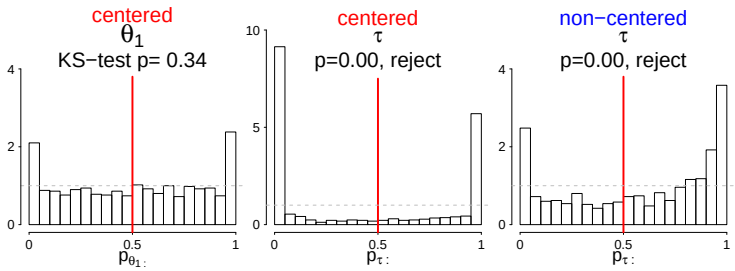


Figure 3.6: *In the eight-school example, the VSBC diagnostic verifies VI estimation of θ_1 is unbiased as the distribution of $p_{\theta_1:}$ is symmetric. τ is overestimated in the centered parametrization and underestimated in the non-centered one, as told by the right/ left skewness of $p_{\tau:}$.*

panel demonstrates that the re-parametrized ADVI posterior is much closer to the truth, and has smaller biases for both first and second moment estimations.

We can assess the marginal estimation using VSBC diagnostic, as summarized in Figure 3.6. In the centered parametrization, the point estimation for θ_1 is in average unbiased. The histogram for τ is right-skewed. Hence we conclude τ is over-estimated in the centered parameterization. On the contrast, the non-centered τ is negatively biased, as diagnosed by the left-skewness of $p_{\tau:}$. Such conclusion is consistent with the bottom-right panel in Figure 3.5.

To sum up, this example illustrates how the Gaussian family assumption can be unrealistic even for a simple hierarchical model. It also clarifies VI posteriors can be both over-dispersed and under-dispersed, depending crucially on the true parameter dependencies. Nevertheless, the recommended PSIS and VSBC diagnostics provide a practical summary of the computation result.

Cancer classification using horseshoe priors. We illustrate how the proposed diagnostic methods work in the Leukemia microarray cancer dataset that contains $D = 7129$ features and $n = 72$ observations. Denote $y_{1:n}$ as binary outcome and $X_{n\times D}$ as the predictor, the logistic regression with a regularized horseshoe prior (Piironen and Vehtari, 2017c) is given by

$$y|\beta \sim \text{Bernoulli}\left(\text{logit}^{-1}\left(X\beta\right)\right), \quad \beta_j|\tau,\lambda,c \sim \text{normal}(0,\tau^2\tilde{\lambda}_j^2),$$
$$\lambda_j \sim \text{C}^+(0,1), \quad \tau \sim \text{C}^+(0,\tau_0), \quad c^2 \sim \text{Inv–Gamma}(2,8).$$

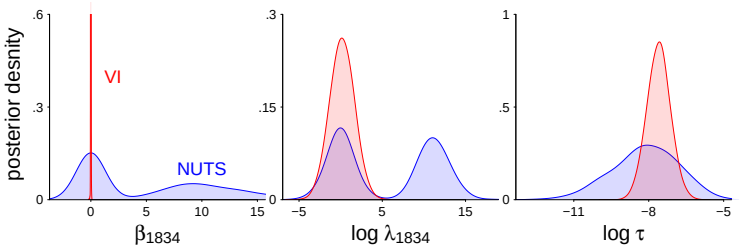


Figure 3.7: *The comparison of ADVI and true posterior density of* θ_{1834}, $\log \lambda_{1834}$ *and* τ *in the horseshoe logistic regression. ADVI misses the right mode of* $\log \lambda$, *making* $\beta \propto \lambda$ *become a spike.*

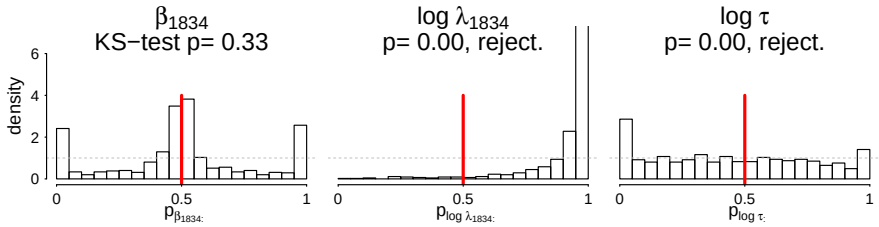


Figure 3.8: *VSBC test in the horseshoe logistic regression. It tells the positive bias of* τ *and negative bias of* λ_{1834}. β_{1834} *is in average unbiased for its symmetric prior.*

where $\tau > 0$ and $\lambda > 0$ are global and local shrinkage parameters, and $\tilde{\lambda}_j^2 = c^2\lambda_j^2/\left(c^2 + \tau^2\lambda_j^2\right)$. The regularized horseshoe prior adapts to the sparsity and allows us to specify a minimum level of regularization to the largest values.

ADVI is computationally appealing for it only takes a few minutes while MCMC sampling takes hours on this dataset. However, PSIS diagnostic gives $\hat{k} = 9.8$ for ADVI, suggesting the VI approximation is not even close to the true posterior. Figure 3.7 compares the ADVI and true posterior density of β_{1834}, $\log \lambda_{1834}$ and τ. The Gaussian assumption makes it impossible to recover the bimodal distribution of some β.

The VSBC diagnostics as shown in Figure 3.8 tell the negative bias of local shrinkage λ_{1834} from the left-skewness of $p_{\log \lambda_{1834}}$, which is the consequence of the right-missing mode. For compensation, the global shrinkage τ is over-estimated, which is in agreement with the right-skewness of $p_{\log \tau}$. β_{1834} is in average unbiased, even though it is strongly underestimated from in Figure 3.7. This is because VI estimation is mostly a spike at 0 and its prior is symmetric. As we have explained, passing the VSBC test means the average unbiasedness, and does not ensure the unbiasedness for a specific parameter setting. This is the price that VSBC pays for averaging over

all priors.

3.6 Discussion

The proposed diagnostics are local. As no single diagnostic method can tell all problems, the proposed diagnostic methods have limitations. The PSIS diagnostic is limited when the posterior is multimodal as the samples drawn from $q(\theta)$ may not cover all the modes of the posterior and the estimation of k will be indifferent to the unseen modes. In this sense, the PSIS diagnostic is a *local diagnostic* that will not detect unseen modes. For example, imagine the true posterior is $p = 0.8\text{N}(0, 0.2) + 0.2\text{N}(3, 0.2)$ with two isolated modes. Gaussian family VI will converge to one of the modes, with the importance ratio to be a constant number 0.8 or 0.2. Therefore k is 0, failing to penalize the missing density. In fact, any divergence measure based on samples from the approximation such as $\text{KL}(q, p)$ is *local*.

The bi-modality can be detected by multiple over-dispersed initialization. It can also be diagnosed by other divergence measures such as $\text{KL}(p, q) = \mathbb{E}_p \log(q/p)$, which is computable through PSIS by letting $h = \log(q/p)$.

In practice a marginal missing mode will typically lead to large joint discrepancy that is still detectable by $\hat{k}$, such as in Section 3.5.

The VSBC test, however, samples the true parameter from the prior distribution directly. Unless the prior is too restrictive, the VSBC p-value will diagnose the potential missing mode.

Tailoring variational inference for importance sampling. The PSIS diagnostic makes use of stabilized IS to diagnose VI. By contrast, can we modify VI to give a better IS proposal?

Geweke (1989) introduce an optimal proposal distribution based on split-normal and split-t, implicitly minimizing the χ^2 divergence between q and p. Following this idea, we could first find the usual VI solution, and then switch Gaussian to Student-t with a scale chosen to minimize the χ^2 divergence.

More recently, some progress is made to carry out variational inference based on Rényi diver-

gence (Li and Turner, 2016; Dieng et al., 2017). But a big α, say $\alpha = 2$, is only meaningful when the proposal has a much heavier tail than the target. For example, a normal family does not contain any member having finite χ^2 divergence to a Student-t distribution, leaving the optimal objective function defined by Dieng et al. (2017) infinitely large.

There are several research directions. First, our proposed diagnostics are applicable to these modified approximation methods. Second, PSIS re-weighting will give a more reliable importance ratio estimation in the Rényi divergence variational inference. Third, a continuous $\hat{k}$ and the corresponding α are more desirable than only fixing $\alpha = 2$, as the latter one does not necessarily have a finite result. Considering the role $\hat{k}$ plays in the importance sampling, we can optimize the discrepancy $D_\alpha(q||p)$ and $\alpha > 0$ simultaneously. We leave this for future research.

Chapter 4. Assessing covariate imbalance in observational studies

"The three thousand realms of existence are all possessed by life in a single moment."
—Zhiyi, *Mohe Zhiguan*

4.1 Introduction

Learning from data involves extrapolation. In observational studies to make relevant inferences, we need to extrapolate from sample data to population, from control to treatment group, and from measurements to underlying constructs of interest. To what extent these extrapolations are reliable depends on differences in the distribution of covariates between the treated and control group, and the amount that these disparities map to differences in treatment effects. If the treatment is randomly assigned, a model-free sample mean yields an unbiased average treatment effect. In contrast, when the covariates are highly imbalanced, inferences become sensitive to assumptions about the form of treatment interactions.

When the treatment z is binary, the imbalance is the difference of covariate distributions in the treated $p(x|z = 0)$ and control $p(x|z = 1)$ groups, measured either in sample or in the population. However, unlike the concept of *overlapping* that asks for a common support of these two distributions, the degree of *imbalance* is a continuous spectrum rather than a binary distinction. A hypothesis testing for imbalance is misleading, as a priori we know the observational data from treated and control groups are unlikely distributed identically, and hence with enough data we will always reject the null. The real assessment is to to choose the appropriate divergence metric, and to determine whether we should trust the extrapolation results given the covariate imbalance having readily occurred in the data.

There is a rich literature that raised concerns for lack of overlapping in causal data analysis (e.g., Imbens, 2004; Imai et al., 2008; Hill and Su, 2013; Imbens, 2015; Vegetabile et al., 2020). Recent

works (Shalit et al., 2017; Sundin et al., 2019) has theoretically analyzed how the lower bound the mean squared and Type-S error of the treatment effect estimation as a function of some chosen imbalance metric, while these bounds are not tight for practical usage. Most existing methods either relied on further model assumptions, or attempted to compute some divergence measure between $p(x|z = 0)$ and $p(x|z = 1)$. In addition to a hard-to-interpret scale of the divergence measure, the choice of the divergence is also arbitrary, and many proposed divergences are not even invariant under reparameterization and affine transformation of covariates, the basic preprocessing of data. This is especially a concern in modern high dimensional data setting (D'Amour et al., 2020) where the imbalance can be arbitrarily worse by adding more unrelated or weakly predictive variables into the covariate list.

This chapter proposes an accessible imbalance metric that is related to the Kullback–Leibler divergence between the covariate distributions, the divergence that we show is most relevant to average treatment effect (ATE) estimation. The proposed metric effectively takes into account only the predictively relevant covariates, and can be adapted to the average treatment effect in treated (ATT) and the average treatment effect in control (ATC) when the full imbalance is detected too severe. Besides used for imbalance diagnostics, our approach comes with a post-hoc design analysis that provides a rough scale of the sample size requirement, often more appropriate than usual power calculation.

4.2 Why overlapping matters

We start by a binary treatment assignment scheme $z = 0$ or 1. Denote the outcomes $y \in \mathcal{Y} \subset \mathbb{R}$ and other covariates $x \in \mathcal{X} \subset \mathbb{R}^d$, the observed data is a sequence of triples $(y_i, z_i, x_i)_{i=1,2,\ldots,n}$. We adopt a fully Bayesian view, and assume a probabilistic generative distribution with joint density $p_{\text{sample}}(y_{1:n}, z_{1:n}, x_{1:n})$ in the super population (i.e., the unknown data generating mechanism) from which the observed sample is drawn. We also assume the joint distribution is factorizable, i.e., $p(y_{1:n}, z_{1:n}, x_{1:n}) = \prod_{i=1}^{n} p_{\text{sample}}(y_i, z_i, x_i)$, which is related to the *individualistic treatment response* assumption and excludes interference. By the ignorability assumption, the expected treatment

effect at any individual covariate location $x^* \in \mathcal{X}$ is $\mathbb{E}(y|x^*, z^* = 1) - \mathbb{E}(y|x^*, z^* = 0)$.

A probabilistic assumption is

$$0 < \Pr(z = 1|x_i) < 1, \ \forall i, \tag{4.1}$$

the population version of which can also be phrased as that the covariate densities $p(x|z = 1)$ and $p(x|z = 0)$ have common support. It ensures that the average treatment effect (ATE) is a valid estimand by taking expectation over both x^* and y with respect to the joint distribution

$$\text{ATE} = \mathbb{E}_X \left(\mathbb{E}_Y(y|x, z^* = 1) - \mathbb{E}_Y(y|x, z^* = 0) \right). \tag{4.2}$$

From the generative model perspective, the two expectations in (4.2) are the integrals of the function $h(y, z, x) = y$ under the target distributions

$$p_{\text{target},1}(x, y, z) = p(y|x, z)p(x)\mathbb{1}(z = 1); \quad p_{\text{target},0}(x, y, z) = p(y|x, z)p(x)\mathbb{1}(z = 0).$$

The key feature of an observational study is the dependency of z on x. Hence, these two target distribution is different from the sampling distribution p_{sample} that is factorizable by

$$p_{\text{sample}}(x, y, z) = p(y|x, z)p(z|x)p(x),$$

nor its conditional distribution on $p_{\text{sample}}(x, y|z = 1)$ and $z = 0$.

To compute the expectation under the *target* destiny using simulation draws from some *proposal* sampling distribution is a common task is Bayesian computation. To be more explicit, we now view the observed data $(x_i, y_i, z_i)_{i=1}^n$ as pseudo Monte Carlo draws from their joint sampling distribution p^{sample}. To correct for the sample-target difference, these Monte Carlo draws are reweighed by their importance weights. For treated units $z_i = 1$, the importance weight is the ratio of the target and

the sampling density:

$$r_i = \frac{p_{\text{target},1}}{p_{\text{sample}}} = \frac{p(y_i|x_i, z_i)p(x_i)1(z_i = 1)}{p(y_i|x_i, z_i)p(x_i)p(z_i|x_i)} = \frac{1(z_i = 1)}{p(z = 1|x_i)} = \frac{1(z_i = 1)}{e(x_i)}, \tag{4.3}$$

and likewise $r_i = \frac{1(z_i=0)}{1-e(x_i)}$ for control units, where $e(x) \coloneqq p(z = 1|x)$ often called the propensity score at x. The importance sampling estimate is the usual weighting estimate in causal inference,

$$\text{ATE} \approx \frac{\sum_{i:z_i=1} y_i e(x_i)^{-1}}{\sum_{i:z_i=1} e(x_i)^{-1}} - \frac{\sum_{i:z_i=0} y_i(1 - e(x_i))^{-1}}{\sum_{i:z_i=0}(1 - e(x_i))^{-1}}. \tag{4.4}$$

In practice these probabilities are estimated from data, so practitioners are actually using inverse estimated probability weighting.

The purpose of the current chapter is not to reinvent, or restrict estimation to, propensity score weighting. Rather, we are addressing the consequence of the covariate imbalance. Form the importance sampling perspective, the only assumption for a (self-normalized) importance sampling estimate to be (asymptotically) unbiased is the common support of $p(x|z = 1)$ and $p(x|z = 0)$, this is equivalent to the probabilistic assumption in causal inference, $0 < p_i(z_i = 1|x_i) < 1$. The law of large numbers ensures the weighted estimates converges to the true ATE if the sample size from both treated and control group is large enough.

However, unbiasedness does not ensure a finite sample accuracy of the treatment effect estimate. In randomized trials such that $p(z|x) = p(z)$ thereby $p^{\text{sample}} = p^{\text{target}}$, the sample average of $y|z = 1$ minus $y| = 0$ forms an unbiased estimate of ATE. As a Monte Carlo estimates, this sample average possesses $\sqrt{n}$ convergence rate regardless of dimensions. In importance sampling scheme, there is extra sampling variation in r. Only when both $\mathbb{E}_{p_{\text{sample}}} r^2 < \infty$ and $\mathbb{E}_{p_{\text{sample}}}(ry)^2 < \infty$ hold, the central limit theorem holds and brings square root convergence rate (Geweke, 1989). Furthermore, when the third moments is finite, then the Berry-Esseen theorem states faster convergence rate to normality (Chen and Shao, 2004). More generally, the finite sample convergence rate of the ATE estimates, viewed as an importance sampling estimate depended on the number of existing moments of importance ratios (Cortes et al., 2010; Chatterjee and Diaconis, 2018; Vehtari et al.,

2019b).

4.3 Proposed covariate assessment

For ATE estimates, the target distribution is the marginal distribution of covariates

$$p(x) = \pi_1 p_1(x) + \pi_0 p_0(x), \quad \pi_1 = \Pr(z = 1), \quad \pi_0 = \Pr(z = 0).$$

It is the mixture of covariate distribution in treated and control groups,

$$p_1(x) = p(x|z = 1), \quad p_0(x) = p(x|z = 0).$$

From a similar argument in the previous two chapters, we propose PSIS-diagnostics for ATE estimates. Recall a generalized Pareto distribution with shape parameter k and location-scale parameter (μ, τ) has the density

$$p(y|\mu, \sigma, k) = \begin{cases} \frac{1}{\sigma}\left(1 + k\left(\frac{y-\mu}{\sigma}\right)\right)^{-\frac{1}{k}-1}, & k \neq 0. \\ \frac{1}{\sigma}\exp\left(\frac{y-\mu}{\sigma}\right), & k = 0. \end{cases}$$

We summarize the method in Algorithm 4. The key diagnostic quantity is the shape parameter $\hat{k}$ of the importance weights from (4.3), which quantifies the divergence between treated or control group to the target distribution in ATE estimates.

To extrapolate between p_1 and p_0 is almost as hard (or easy) to extrapolate from p_1 or p_0 to the mixture, as

$$\mathrm{KL}(p||p_0) \leq \pi_1 \,\mathrm{KL}(p_1||p_0), \quad \mathrm{KL}(p||p_1) \leq \pi_0 \,\mathrm{KL}(p_0||p_1).$$

$$\mathrm{KL}(p||p_0) \geq \pi_1 \,\mathrm{KL}(p_1||p_0) + \log \pi_0, \quad \mathrm{KL}(p||p_1) \geq \pi_0 \,\mathrm{KL}(p_0||p_1) + \log \pi_1.$$

This equivalence is also verified by that the *matching* weight has the same distribution of the propensity score weights, up to a constant location shift: $r_i^{\text{matching}} = e_i/(1 - e_i) = 1/(1 - e_i) - 1 =$

Algorithm 4: Diagnostics for covariate imbalance

Result: $\hat{k}$ and n_{eff} diagnostics

Estimate the propensity score $\hat{e}(x_i) = \hat{P}(z = 1|x_i)$;
Fit the generalized Pareto distribution to the weights $\{1/\hat{e}(x_i) : z_i = 1\}$ and $\{1/(1 - \hat{e}(x_i)) : z_i = 0\}$;
Obtain the shape parameter $\hat{k}^1$ and $\hat{k}^2$ respectively;
if *both* $\hat{k}^1, \hat{k}^2 < 0.7$ **then**
    Pass the diagnostics for ATE, the imbalance is mild;
else
    if *one of* $\hat{k}^1$ *or* $\hat{k}^2 < 0.7$ **then**
        ATE or ATC is still computable;
    else
        Indicate severe non-overlapping and unstable causal estimates;
    end
end
if $0 < \hat{k}^1, \hat{k}^2 < 1$ **then**
    Compute effective sample size n_{eff} for next stage experiment design.
end

$r_i^{\text{weighting}} - 1$. The generalized Pareto distribution has already taken into account the location shift, therefore it makes no difference to examine the weights from inverse propensity score or matching. Put it in another way, matching and weighting have the same level of extrapolation, which is only determined by the divergence between p_0 and p_1 and vice versa. That said, various matching strategy (the choice of nearest neighbors, distance measures, kernels) improve the accuracy of estimates by imposing more regularization on the weights.

In addition, the KL divergence is not symmetric. It is possible that only one of $\text{KL}(p||p_0)$ and $\text{KL}(p||p_1)$ is small, which will be revealed by different $\hat{k}^1$ and n_{eff} for z_i/e_i and $(1 - z_i)/(1 - e_i)$. It provides a practical guidance for whether we replace the estimated ATE by ATT or ATC, and which of ATT or ATC is more feasible to estimate.

Using $\hat{k}$ in sample design. Under a variety of assumptions (e.g., Agapiou et al., 2017; Sanz-Alonso, 2018; Chatterjee and Diaconis, 2018; Vehtari et al., 2019b), the necessary and sufficient sample size to control the importance sampling error is roughly scales linearly with $\exp\left(\text{KL}(\text{sampling proposal} \,||\, \text{target})\right)$. The most comprehensive results are due to Chatterjee and

Diaconis (2018), who show that a larger sample size than this gives tail guarantees for the error in self-normalized importance sampling, while for smaller sample sizes it is not possible to control the large deviations.

We can turn this theory into a heuristic bound by assuming that the density ratios $r(\theta) = p(\theta)/g(\theta)$ exactly form a Pareto distribution with shape parameter $k \in (0, 1)$ under g, in which case we can estimate $\mathrm{KL}(p||g)$ by $\int_1^\infty r \log r \frac{1}{k} r^{-1/k-1} dr = k(1-k)^{-2}$. Hence, the effective sample size $n_{\text{eff}} = n\hat{k}^{-1}(1-\hat{k})^2,\ \hat{k} \in (0, 1)$.

Dimension reduction from propensity scores. Throughput the chapter we have used the phrase "heavy tail" vaguely. For our purpose it does not matter if the distribution refers to the covariates x or the propensity score distribution. Slightly abusing the notation, we denote e_1 to be the distribution of propensity scores of treated units $e_1 = \Pr(z = 1|x), x \sim p_1$ and e_0 as the distribution of propensity scores of control groups $e_0 = \Pr(z = 1|x), x \sim p_0$. It is easy to show that KL divergence between $p_1(x)$ and $p_0(x)$ (as two densities) is the same as the KL divergence between the distribution of the propensity score e_1 and e_2. It is equivalent to conduct importance sampling on x, or on propensity scores $e(x)$, or on any of their bijective transformations. The following descriptions are equivalent:

1. k_1 is large
2. The covariate distribution in the treated group X_t has a heavier tail than the control group X_c;
3. The propensity score distribution in the treated group has a heavier tail than the control group;
4. The propensity score ratio $1/(1-e)$ has a heavy right tail;
5. The matching ratio $e/(1-e)$ has a heavy right tail.

4.4 Related work

There is a rich literature on imbalance assessment (for a review, see Imbens and Rubin, 2015, Chapter 13). Many existing methods that are aimed to assess the overlap in covariate distributions largely rely on normal approximation of $x|z$. For individual predictors, it is common in applied

research to report the sample mean and standard deviation of each coordinate of x in the treated and control group, denoted by μ_t, μ_c, σ_t, σ_c respectively, which further computes the normalized difference

$$\Delta = \frac{\mu_t - \mu_c}{\sqrt{(\sigma_t^2 + \sigma_c^2)/2}}$$

Simply running a t test is misleading as we know a priori that the control and treated groups do not have the same distributions. Other discrepancies include the ratio of the standard deviation

$$\Gamma = \log(\sigma_t/\sigma_c),$$

and the tail-quantile difference,

$$\pi_c^{0.05} = 1 - \hat{F}_c\left(\hat{F}_t(0.95)\right) + \hat{F}_c\left(\hat{F}_t(0.05)\right), \quad \pi_t^{0.05} = 1 - \hat{F}_t\left(\hat{F}_c(0.95)\right) + \hat{F}_t\left(\hat{F}_c(0.05)\right)$$

where $\hat{F}_c$ and $\hat{F}_t$ are the empirical distribution functions in control and treated group respectively.

For multivariate distribution, Imbens and Rubin (2015) recommend to examine the imbalance of the log odd of propensity ratios in the treated and control group

$$l(x) = \log \frac{e(x)}{1 - e(x)},$$

for the reason that the log odds are "more likely to be approximately normal distributed." Imbens and Rubin (2015) further introduce an overlap measure by simply counting the number of controlled (treated) units within a small log propensity score ratio neighbor of every treated (controlled) units. Again, the choice of distance measure and the construction of nearest neighbors can be ad hoc.

Shalit et al. (2017) and Sundin et al. (2019) quantified the effect of imbalance on individual treatment effect by the integral probability metric, a generalization of f-divergence,

$$\sup_{g \in G} \left| \int_{\mathcal{X}} g(x)\big(p(x|z=1) - p(x|z=0)\big)dx \right|,$$

where G is some function family consisting of functions $g : \mathcal{X} \to \mathbb{R}$, in part because this metric can be computed directly using the empirical distributions (Sriperumbudur et al., 2012).

The main bound their approaches address is

$$|| \int_{\mathcal{X}} \mathbb{E}_{y|x,z} L(y, f(x,z)) p(x|1-z) dx - \int_{\mathcal{X}} \mathbb{E}_{y|x,z} L(y, f(x,z)) p(x|z) dx ||.$$

From importance sampling perceptive, this is the difference between the target integral and the Monte Carlo integral, $\mathbb{E}_p f - \mathbb{E}_q f$, which is only relevant if we ignore all the casual context and treat analysis as if in randomized trials. Furthermore, this metric varies by feature representation, over which the method also optimize. As a comparison, the proposed $\hat{k}$ is a intrinsic quantity in the population, not depending on feature transformation. Besides, a directed-metric is more useful than an undirected-distance because we can sepearte the task of ATT and ATC.

In short, there are a few reasons why existing imbalance metrics are not perfect for practical data analysis. First, the scale of divergence measures are hard to interpret. It is often subjective to determine how large the discrepancy in the mean difference or the integral probability metric is practically too big to disable causal estimates. Second, unlike a hypothesis where it is enough to find *a* test statistic along which the covariate distribution exhibits difference, here we know that they are different but want to quantity *the* imbalance that has impact on the treatment effect estimate. For example, the mean difference or integral probability metric is not invariant under an affine transformation. Third, practical data analysis often include as many pre-treatment covariates as possible (or at least attempt to if the imbalance if not a concern). Some inputs could be weakly related to the outcome. If we know some dimension of x is independent of the outcome y given z, its imbalance is irreverent to causal estimate. In practice we do not know how predictive each input is in advance, hence the imbalance quantification becomes misleading when it includes all weakly predictive inputs, and weight them equally in the sum.

Regularized importance sampling and trimming. In causal inference, various methods has been introduced to stabilize weighted estimates. Trimming (Crump et al., 2009; Lee et al., 2011)

discard all units with estimated propensity score outside the "rule of thumb" range $[0.1, 0.9]$, which is computationally equivalent to truncated importance sampling (Ionides, 2008) with a fixed truncation threshold $r > 10$. Subclassifications (Rosenbaum and Rubin, 1983, 1984) re-estimates the propensity score in a small interval, and it is equivalent to replacing the importance ratio by a piece wise constant function that also trims the tail.

Rubin and Thomas (1992, 1996, 2000) point out that the estimated propensity score rather than the exact true propensity score leads to "better balance and better causal inference estimates." Analogously, researchers in machine learning propose to re-estimate the sampling density even if it is known and such re-estimation result in "surprising" superefficiency (Liu and Lee, 2016). From our point of view, to estimate the propensity scores when they are unknown, or to re-estimate sampling distribution even when it is known, should be viewed as implicit regularization that comes from the smoothness in regression or kernel density estimation. The superefficiency is just the usual bias-variance tradeoff, and can also be achieved by other regularization forms of importance ratios (e.g., Vehtari et al., 2019b).

4.5 Examples

The first example illustrates why the intuitive visual check and commonly used divergence metrics are often not enough for imbalance assessment. The univariate covariate in control group $x|z = 0$ is generated from normal$(0, 1)$. In order to compare the difference effect of bulk and tail, we consider five data generating mechanisms for the treated covariate $x|z = 1$. The sample size is balanced such such that $n_{\text{treated}} = n_{\text{control}} = n$. In the first three cases, the treated group has the same normal tail distribution outside the $[0.05, 0.95]$ quantiles, with the bulk modified to be uniform or piece wise uniform distribution. In the fourth and fifth cases, the bulk density is unchanged but the tail is modified to be a normal$(0, 2)$ outside the $[0.1, 0.9]$ quantiles. Similarly in the fifth case, the tail outside $[0.05, 0.95]$ quantiles is a student-t distribution with the degree of freedom 3. We generate the outcome from $y = |x|z + x +$ normal$(0, 0.1)$ that is unknown to the modeler, and compute the analytic average treatment effect in the population.

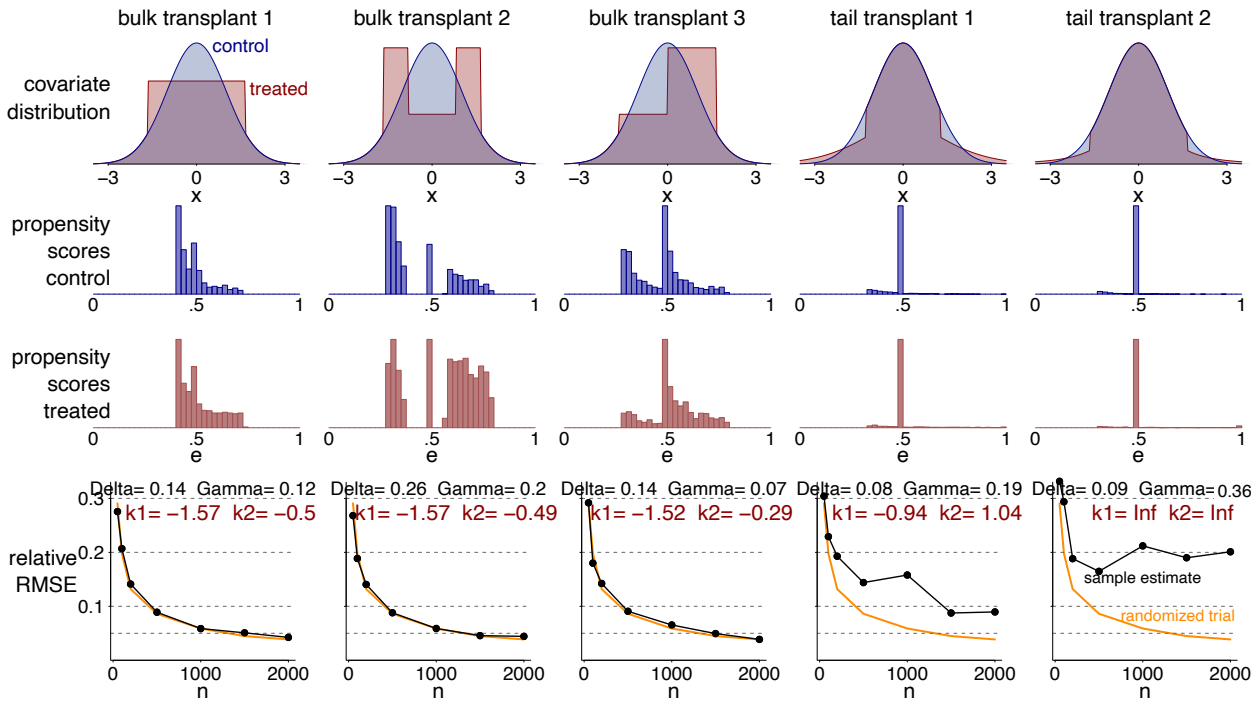


Figure 4.1: *In each data generating mechanism, we generate n control units (densities in blue) with covariate x from standard normal and n treated units (densities in red) from a slightly different distribution. In the first three manipulation, we keep the tail unchanged and transplant the bulk part from uniform and piece wise uniform distribution. In the last two cases we change the tail to an inflated normal or a student t tail. Rows 2–3 are the histograms of the propensity scores in the treated and control group. On the last row, we vary the sample size n from* 50 *to* 2000*, report the root mean square errors (averaged over 200 repetitions) of the ATE estimates, and print the estimated* $\hat{k}$*, standardized mean difference* Δ*, and the standard deviation difference* Γ*.* $\hat{k}$ *is a good indicator of the finite sample error and the convergence rate. For* $\hat{k} < 0.5$*, the convergence rate (black dots) is nearly as quick as a randomized trial (orange curve).*

The first row of Figure 4.1 illustrate the density of covariate distribution in the treated and control group in cases 1–5. In Rows 2–3, we project the the covariate distribution into propensity scores, and visualize the histogram of the propensity scores in the treated and control group with $n = 1000$ randomly sampled units. A careful visual check can reveal the potential nonoverlapping, but it is not clear which one is "more balanced" or "closer." For example, the histogram of bulk transplant 2 seems to have different shapes, while the tail transplants seemingly overlap enough except for a tiny right tail region.

To separate the propensity score estimation noise, we use the exact propensity scores in this subsection. Accordingly the error of the inverse probability weighting estimate constitute the oracle

learning bound given the population imbalance. In the last row of Figure 4.1, we compute the relative root mean square error of the inverse probability weighting estimates (RMSE of ATE estimate/ true ATE, averaged over 200 independent repetitions) for each experiment with varying sample size. For comparison, we also compute the errors of the randomized trails that are generated using the $x|(z=1) \stackrel{d}{=} x|(z=0) \sim \text{normal}(0,1)$. Besides the PSIS $\hat{k}$ in treated and control groups separately, we also compute the divergence measure on absolute moment $\Delta = (\mathbb{E}_n |x_c| - \mathbb{E}_n |x_t|)/\sqrt{(\sigma_t^2 + \sigma_c^2)/2}$, and standard deviation difference $\Gamma = \log(\sigma_t/\sigma_c)$ since the density is symmetric. The finite sample estimates at $n = 1000$ are printed above each panel. $\hat{k}$ captures the actual divergence we are interested. When only the bulk is manipulated, $\hat{k}$ is small. The corresponding ATE estimate error exhibits similar and quick convergence rate—almost identical to the randomized trail result at the same same size. This static asymptotic behavior is in agreement with square root convergence bound as long as $\hat{k} < 0.5$. In contrast, when the tail is modified, the PSIS $\hat{k}$ is 1.0 in the inflated normal tail and infinity in the student-t tail. They are good indicators for the actual large and slowly-converging ATE error. In contrast, other divergence measures are not informative for the ATE estimation accuracy. In particular, the f-divergence can large by modifying the buck for its contribution to the integral, but it is also the region where imbalance is less harmful for there are enough sample units in both treated and control groups.

Misspecification of propensity scores. So far we have only considered the exact propensity score, which will rarely be known in practice. The error of the ATE estimates, can then be decomposed into two terms

$$1/n \sum r_i f_i - \text{true ATE} = \left(1/n \sum r_i f_i - 1/n\, \mathbb{E} \sum r_i f_i\right) \tag{4.5}$$

$$+ \left(1/n\, \mathbb{E} \sum (r_i - r_i^*) f_i\right). \tag{4.6}$$

where r_i is the estimated and potentially regularized ratios, and r_i^* is the unknown true importance ratios.

The first term, the Monte Carlo error term (4.5), is the computation instability of the weighting

estimates, and $\hat{k}$ still reflects its convergence rate. The second term (4.6), the model misspecification error, which can be evaluated in the propensity score model.

Nevertheless, the following example shows PSIS diagnostic could still be useful in this model misspecification situation. We consider the same data generating mechanism used in the previous section

$$y = x + z|x| + \text{normal}(0, 0.1).$$

For the control group, we generate the one-dimensional covariate x_c from normal$(0, 1)$, whereas the treated group x_t is from normal$(\theta, 1)$. The sample size in both control and treated group is n. The average treatment effect in the population can be computed explicitly as $\mathbb{E}\,|x|$, which will be compared with the propensity score weighted estimates of ATE.

Firstly, we use the exact propensity score $e(x) = \text{normal}(x|\theta)/(\text{normal}(x|0) + \text{normal}(x|\theta))$. We vary θ from 0.1 to 3, as a larger θ reflects a lager control-treated discrepancy. We also vary the sample size from 100 to 10000. Figure 4.2 shows the relative root mean square error (RMSE of estimated ATE / actual ATE) as a function of PSIS-$\hat{k}$. As expected, a larger $\hat{k}$ reveals the heavier tail of the log propensity score and slower convergence rate and therefore a larger RMSE. Just like the classic divergence measure Δ and Γ, $\hat{k}$ itself does not take into account the sample size directly. However, the distribution with a large $\hat{k}$ has such a slow convergence rate that any practically large number of sample size will yield a unreliable estimate. According to the empirical study in Vehtari et al. (2019b), we recommend to use $\hat{k} < 0.7$ as a threshold, although user should be more conservative if the sample size is small or the propensity score has already been over-smoothed. On the other hand, such diagnostic quantifies the divergence in the population level, and will be useful in the design phase for calculating the minimal sample size needed for next experiment or meta analysis. Vehtari et al. (2019b) provide the empirical formula.

The relative effective sample size is more useful metric for one single trial. It takes into account the sample size used in the experiment and therefore make studies with different sample size comparable. We can read the approximate scale of estimation error by (effective sample size)$^{-1/2}$.

Now, in a more realistic setting we do not know the propensity score. What if we misspecify

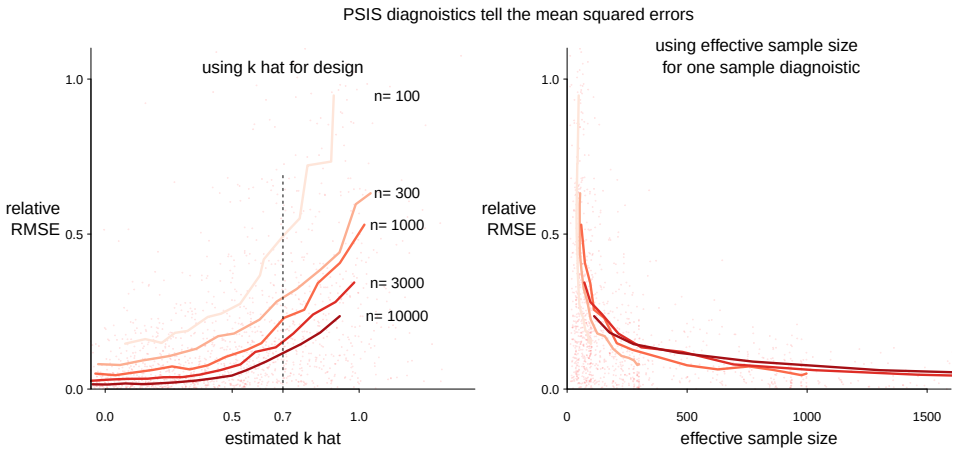


Figure 4.2: *We generate the sample size in control group from normal*$(0,1)$ *and the treated group from normal*$(\theta,1)$*. θ varies from 0.1 to 3 and sample size varies from 100 to 10000. For each setting we repeated the simulation 100 times to compute the relative root mean square error of the inverse probability weighting, and $\hat{k}$, n_{eff} in the proposed PSIS diagnostics. We use the exact propensity score.*

the model $p(z|x)$ with a probit link and use this estimated propensity score in the weighting?

Figure 4.3 shows the result. It has the same pattern as in 4.2, but there is also significant difference. Experiments with same $\hat{k}$ has similar RMSE, nearly independent of sample size.

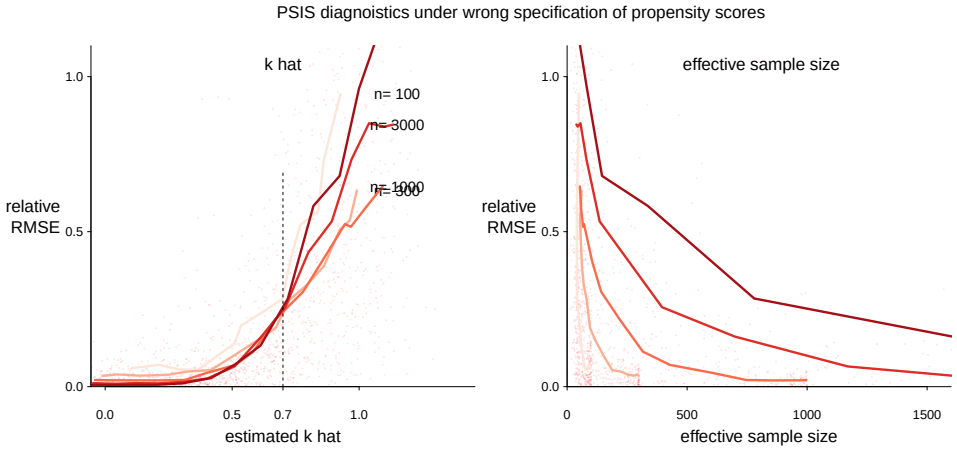


Figure 4.3: *We generate the sample size in control group from normal*$(0,1)$ *and the treated group from normal*$(\theta,1)$*. θ varies from 0.1 to 3 and sample size varies from 100 to 10000. For each setting we repeated the simulation 100 times to compute the relative root mean square error of the inverse probability weighting, and $\hat{k}$, n_{eff} in the proposed PSIS diagnostics. The propensity score is estimated from a misspecified probit model.*

In this example, the model misspecification error term remains a constant (does not decrease with

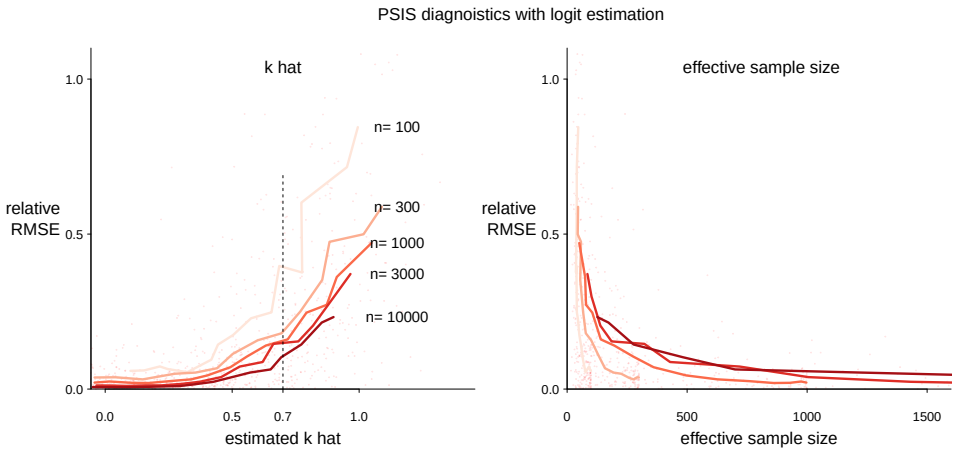


Figure 4.4: *We generate the sample size in control group from normal*$(0,1)$ *and the treated group from normal*$(\theta,1)$. *The parameter* θ *varies from 0.1 to 3 and sample size varies from 100 to 10000. For each setting we repeated the simulation 100 times to compute the relative root mean square error of the inverse probability weighting, and* $\hat{k}$, n_{eff} *in the proposed PSIS diagnostics. The propensity score is estimated from the correct logit model.*

even larger sample size) when there is symmetric bias in propensity score estimates $\mathbb{E}\, r(x) - \mathbb{E}\, r^*(x)$. This constant error offsets the difference in sample size. Furthermore, that bias term is more significant in the tail– simply because we use the inverse propensity score $1/e$, even a constant bias in the logit or probit scale leads a larger bias in the tail of $1/e$. For this concern, $\hat{k}$ quantifies how heavy the tail region is, and a heavy tail *amplifies* the bias for more mass concentration. As a whole, $\hat{k}$ diagnostics appear even more distinguishable for ATE estimates. Again, we observe a sharp increase at $k > 0.7$, consistent with previous recommendation.

Finally, Figure 4.4 presents the result under the correct propensity score model (logit link). It is similar to Figure 4.2. Roughly speaking, for simple logistic regression, the epistemic uncertainty in propensity score shrinks as fast as $n^{-1/2}$, therefore the computation instability term, whose convergence rate is approximately $n^{-(1-k)}$ for k >0.5, dominates the final error.

To sum up, $\hat{k}$ diagnoses the tail shape of weighting ratios. A heavy tail either directly leads to slower convergence rate, or amplifies the model misspecification error, both of which contribute to mean square error of ATE estimates.

Part II

Expanding a model

by tempering and path sampling

Chapter 5. The challenge of estimating the normalizing constant[1]

"If an infinite sum is what awaits us
Of days so white and nights so dark,
Then we already are the past we'll be."

—Jorge Luis Borges

In Bayesian computation, the posterior distribution is often available as an *unnormalized density* $q(\theta)$. The unknown and often analytically-intractable integral $\int q(\theta)d\theta$ is called the *normalizing constant* of q. Many statistical problems involve estimating the normalizing constant, or the ratios of them among several densities. For example, the marginal likelihood of a statistical model with likelihood $p(y|\theta)$ and prior $p(\theta)$ is the normalizing constant of $p(y|\theta)p(\theta)$: $p(y) = \int p(\theta, y)d\theta$. The Bayes factor of two models $p(y|\theta_1), p(\theta_1)$ and $p(y|\theta_2), p(\theta_2)$, requires the ratio of the normalizing constants in densities $p(y, \theta_1)$ and $p(y, \theta_2)$.

Besides, we are often interested in the normalizing constant as a function of parameters. In a posterior density $p(\theta_1, \theta_2, \ldots, \theta_d|y)$, the marginal density of coordinate θ_1 is proportional to $\int \ldots \int p(\theta_1, \theta_2, \ldots, \theta_d|y)d\theta_2, \ldots, d\theta_d$, the normalizing constant of the posterior density with respect to all remaining parameters. Accurate normalizing constant estimation means we have well explored region containing most of the posterior mass, which implies that we can then accurately also estimate posterior expectations of many other functionals.

In simulated tempering and annealing, we augment the distribution $q(\theta)$ with an inverse temperature λ and sample from $p(\theta, \lambda) \propto q(\theta)^\lambda$. Successful tempering requires fully exploring the space of λ, which in turn requires evaluation of the normalizing constant as a function of λ: $z(\lambda) = \int q(\theta)^\lambda d\theta$. Similar tasks arise for model selection and averaging on a series of statistical models indexed by a continuous tuning parameter λ: $p(\theta, y|\lambda)$. In cross validation, we attach to each data point y_i a λ_i and augment the model $p(y_i|\theta)p(\theta)$ to be $q(\lambda, y, \theta) = \prod_i p(y_i|\theta)^{\lambda_i} p(\theta)$, such that the pointwise leave-one-out log predictive density $\log p(y_i|y_{-i})$ becomes the log normalizing

[1]This chapter is a slight modified version of Yao et al. (2020a).

constant $\log \int p(y_i|\theta)p\left(\theta \mid y, \lambda_i = 0, \lambda_j = 1, \forall j \neq i\right) d\theta$. We will revisit some of these problems in more depth in chapter 6 for they stand alone important statistical problems.

In all of these problems we are given an unnormalized density $q(\theta, \lambda)$, where $\theta \in \Theta$ is a multidimensional sampling parameter and $\lambda \in \Lambda$ is a free parameter, and we need to evaluate the integrals at any λ,

$$z : \Lambda \to \mathbb{R},\ z(\lambda) = \int_\Theta q(\theta, \lambda) d\theta,\ \forall \lambda \in \Lambda. \tag{5.1}$$

$z(\cdot)$ is a function of λ. For convenience, throughout the chapter we will call $z(\cdot)$ the *normalizing constant* without describing it is a function. We also use notations q and p to distinguish unnormalized and normalized densities.

In most applications, it is enough to capture $z(\lambda)$ up to a multiplicative factor that is free of λ, or equivalently the ratios of this integral with respect to a fixed reference point λ_0 over any λ:

$$\tilde{z}(\lambda) = z(\lambda)/z(\lambda_0),\ \forall \lambda \in \Lambda. \tag{5.2}$$

5.1 Easy to find an estimate, prone to extrapolation

Two accessible but conceptually orthogonal approaches stand out for the computation of (5.1) and (5.2). Viewing (5.1) as the expectation with respect to the conditional density $\theta|\lambda \propto q(\theta, \lambda)$, we can numerically integrate (5.1) using quadrature, where the simplest is linearly interpolation, and the log ratio in (5.2) can be computed from first order Taylor series expansion,

$$\log \frac{z(\lambda)}{z(\lambda_0)} \approx (\lambda - \lambda_0) \frac{d}{d\lambda} \log z(\lambda)|_{\lambda=\lambda_0} \approx (\lambda - \lambda_0) \frac{1}{z(\lambda_0)} \int_\Theta \left(\frac{d}{d\lambda} q(\theta, \lambda)|_{\lambda=\lambda_0} \right) d\theta. \tag{5.3}$$

In contrast, we can sample from the conditional density $\theta|\lambda_0 \propto q(\theta, \lambda_0)$, and compute (5.1) by importance sampling,

$$\frac{z(\lambda)}{z(\lambda_0)} \approx \frac{1}{S} \sum_{s=1}^{S} \frac{q(\theta_s, \lambda)}{q(\theta_s, \lambda_0)}, \quad \theta_{s=1,\cdots,S} \sim q(\theta, \lambda_0). \tag{5.4}$$

How should we choose between estimates (5.3) and (5.4)? There is no definite answer. For example, in variational Bayes with model parameter θ and variational parameter λ, the gradient part of the (5.3) is the *score function* estimator based on the *log-derivative trick*, while the gradient of (5.4) under a location-scale family in $q(\theta_s|\lambda)$ is called the Monte Carlo gradient estimator using the *reparametrization trick*—but it is in general unknown which is better.

On the other hand, both (5.3) and (5.4) impose severe scalability limitations on the dimension of λ and θ. Essentially they *extrapolate* either from the density conditional on λ_0 or from simulation draws from $q(\theta|\lambda_0)$ to make inferences about the density conditional on λ, hence depending crucially on how close the two conditional distribution $\theta|\lambda_0$ and $\theta|\lambda$ are. Even under a normal approximation, the Kullback-Leibler (KL) divergence between these densities scales linearly with the dimension of θ. Thus it takes at least $O(\exp(\dim(\theta))$ posterior draws to make (5.4) practically accurate. The reliability of (5.3) is further contingent on how flat the Hessian of $z(\lambda_0)$ is, which is more intractable to estimate. In practice, these methods can fail silently especially when implemented as a black-box step in a large algorithm without diagnostics.

Free energy and sampling metastability From the Bayesian computation perspective, the log normalizing constant is interpreted as the analogy of "free energy" in physics (up to a multiplicative constant), in line with the interpretation of $\log q(\theta)$ to be the "potential energy" in Hamiltonian Monte Carlo sampling.

A potential energy is called metastable if the corresponding probability measure has some regions of high probability, but separated by low probability connections. Following are two types of metastability, which both cause Markov chain Monte Carlo algorithms difficulty moving between regions. This pathology is often identifiable by a large $\hat{R}$ between chains and low effective sample size in generic sampling (Vehtari et al., 2020a).

1. In an energetic barrier, the density is isolated in multiple modes, and the transition probability is low between modes. In particular, the Hamiltonian—the sum of the potential and kinetic energy—is preserved in every Hamiltonian Monte Carlo trajectory. Hence, a transition across

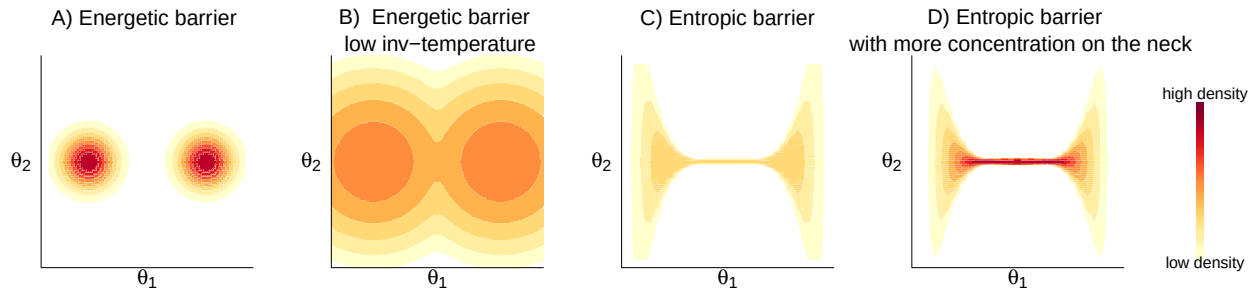


Figure 5.1: An illustration of metastability in a bivariate a distribution $p(\theta_1, \theta_2)$. (A) With a mixture of two Gaussian distributions, the energetic barrier prevents rapid mixing between modes. (B) With a lower inverse temperature λ, the energetic barrier becomes flatter in the conditional distribution $p(\theta|\lambda)$. (C) With an entropic barrier, the left and right part of the distribution is only connected though a narrow tunnel, where the Markov chain will behave like a random walk. (D) Adding more density on the neck increases the transition probability, while leaving $p(\theta_2|\theta_1)$ invariant.

modes is unlikely unless the kinetic energy is exceptionally large.

2. In an entropic barrier, or funnel, the typical set is connected only through narrow and possibly twisted ridges. This barrier is amplified when the dimension of θ increases, in a way that a random walk in high dimensional space can hardly find the correct direction.

As the name suggests, we have the freedom to adaptively tune the free energy of the sampling distribution to remove the metastability therein. The basic strategy is to augment the original metastable density $q(\theta)$ with an auxiliary variable λ, obtaining some density on the extended space $q(\theta, \lambda)$. For energetic barriers, we can take λ to be an inverse temperature variable, a power transformation of the posterior density. The energetic barrier is flattened by a lower inverse temperature. Temperature-based approaches cannot eliminate entropic barriers in the same way, but the transition is boosted by adding probability mass to the neck region. Figure 5.1 gives a graphical illustration.

While the normalizing constant is not itself statistically meaningful in sampling from the augmented density $q(\theta, \lambda)$, computing it serves as an essential intermediate step to constructing the otherwise intractable joint density. Unlike in usual Monte Carlo methods where the target distribution is given and and static, here as in an augmented system, we have the flexibility to either sample θ from some conditional distribution $\theta|\lambda$ at a discrete sequence of λ, or from some joint

distribution of (θ, λ), treating λ continuously. While continuous joint sampling has a finer-grained expressiveness for approximating the normalizing constant, it is harder to access the conditional distribution, which is ultimately what we need when λ is an augmented parameter.

To facilitate the normalizing constant estimation and sampling in metastable distributions, the full problem contains three tasks.

1. Determining what distribution we should sample θ and λ from.

2. Estimating the normalizing constants efficiently with the generated simulation draws.

3. Diagnosing the reliability of the sampling and estimation, particularly distinguishing between an informative extrapolation and a noisy random guess, and deciding when and where to adaptively resample.

In Section 5.2, we introduce a practical solution to all three problems. It extends the idea of path sampling (Gelman and Meng, 1998) to an adaptive design, which performs both the continuously-ranged normalizing constant estimation, and direct sampling of the conditional density. Applying this strategy to metastable sampling, we demonstrate in Sections 6.1 and 6.3 that our proposed adaptive path sampling method enables efficient sampling in both the energetic and entropic bottlenecks, and as a byproduct provides normalizing constant estimation and convergence diagnostics. In Section 5.3, we compare the proposed adaptive sampling to other approaches and show that it is the infinitely dense limit of these basic strategies (5.3) and (5.4). We experimentally illustrate the advantage of the proposed methods in Section 6.5. For the purpose of log normalizing constant estimation and continuous tempering, we have automated our method in the general purpose software Stan (Stan Development Team, 2020), and illustrate the practical implementation in the Appendix.

5.2 The general framework of adaptive path sampling

To begin with, we outline an adaptive path sampling algorithm for the general problem of normalizing constant (ratio) estimation (5.2) in a λ-augmented system $q(\theta, \lambda), \theta \in \Theta, \lambda \in \Lambda$. We further elaborate its application in the context of metastable sampling in Section 6.1 and 6.3.

Instead of sampling λ directly, we consider a transformed sampling parameter a through $\lambda = f(a)$, where the link function $f : \mathcal{A} \to \Lambda$ is continuously differentiable and $\mathcal{A}$ is the support of a. For simplicity, we will use $\mathcal{A} = [0, 1]$ in this section. The actual sampling takes place in the $\mathcal{A} \times \Theta$ space. If there is an interval $\mathcal{I} \subset \mathcal{A}$ which f maps to a fixed value $\lambda_{\mathcal{I}}$, we can directly obtain conditional draws from $\theta|\lambda_{\mathcal{I}}$ by $\{\theta_i : a_i \in \mathcal{I}\}$, while not suffering from discretization errors of the the normalization constant $z(\lambda) = \int_\Theta q(\theta, \lambda) d\theta$. We denote the conditional density $\pi_\lambda := p(\theta|\lambda) = q(\theta, \lambda)/z(\lambda)$.

The general algorithm then iterates the following four steps.

Step 1. Joint sampling with invariant conditional densities. To start, we sample S *joint* simulation draws $(\theta_i, a_i)_{i=1}^S$ from a joint density

$$p(\theta, a) \propto \frac{1}{c(\lambda)} q(\theta, \lambda),\ \lambda = f(a), \tag{5.5}$$

where $c(\lambda)$ is a parametric pseudo prior that is constructed using a series of kernels $\{\gamma_i(\lambda)\}_{i=1}^I$ and regression coefficients $\{\beta_{cj}\}_{j=0}^J$ (which will be updated adaptively throughout the algorithm),

$$\log c(\lambda) = \beta_{c0}\lambda + \sum_{j=1}^{I} \beta_{cj}\gamma_i(\lambda). \tag{5.6}$$

By default, we initialize at a constant function $\beta_c = 0$, i.e., $c(\lambda) \equiv 1$. No matter what the prior $c(\lambda)$ is, the conditional distributions $\theta|a \propto q(\theta, \lambda = f(a))$ in the joint simulation draws are invariant. This motivates to adaptively changing the pseudo-prior $c(\lambda)$.

For the joint sampling task (5.5), we will typically be using dynamic Hamiltonian Monte Carlo (HMC) (Hoffman and Gelman, 2014; Betancourt, 2017) in Stan, which only requires the unnormalized log density $\log q(\theta, \lambda) - \log c(\lambda)$ as input.

Step 2. Estimating the log normalizing constant from joint draws. Thermodynamic integration (Gelman and Meng, 1998) is based on the identity

$$\frac{d}{da}\log z(f(a)) = \mathbb{E}_{\theta|f(a)}\left(\frac{\partial}{\partial a}\log q(\theta, f(a))\right), \tag{5.7}$$

where the expectation is over the invariant conditional distribution $\theta|a \propto q(\theta, f(a))$.

The ratio of the normalizing constant can be computed by integrating both sides of (5.7). To do this, we rank all the sampled draws according to their a coordinate: $a_{(1)} < a_{(2)} < \cdots < a_{(s^*)}$, and compute the pointwise gradients

$$U_{(i)} = \frac{\sum_{a_j=a_{(i)}} \frac{\partial}{\partial a}\log q(\theta, f(a))\big|_{\theta_j, a_j}}{\sum_{j:a_j=a_{(i)}} 1}. \tag{5.8}$$

When there is no tie, the gradient estimate (5.8) essentially approximates the intractable pointwise integral in (5.7), $\mathbb{E}_{\theta|f(a_s)}\left(\frac{\partial}{\partial a}\log\left(q(\theta, f(a))\right)\right)$ by *one* Monte Carlo draw $\frac{\partial}{\partial a}\log\left(q(\theta, f(a))\,|\theta_s, a_s\right.$, a common technique in stochastic approximation.

The integral of the right hand side of (5.7) is then computed from these expectation estimates and the trapezoidal rule. For any $a^* \in \mathcal{A}$, we find its covering interval $a^* \in [a_{(i^*)}, a_{(i^*+1)})$, and compute its normalizing constant with reference to $z(f(0))$ by

$$\begin{aligned}
\log\frac{z(f(a^*))}{z(f(0))} &= \int_0^{a^*}\frac{d}{da}\log z(f(a))da = \int_0^{a^*}\mathbb{E}_{\theta|f(a)}\left(\frac{\partial}{\partial a}\log\left(q(\theta, f(a))\right)\right)da \\
&\approx \frac{1}{2}(a_{(1)} - 0)(U_{(1)} + U_0) + \frac{1}{2}\sum_{j=1}^{i^*-1}(a_{(j+1)} - a_{(j)})(U_{(j+1)} + U_{(j)}) + \frac{1}{2}(a^* - a_{(i^*)})(U_{(i^*)} + U_{a^*}),
\end{aligned} \tag{5.9}$$

where U_{a^*} and U_0 are obtained by extrapolating U.

Step 3. Parametric regularization and adaptive updates. When the normalizing constant $z(\lambda)$ is only required up to a multiplicative factor, we can assume $z(f(0)) = 1$, In Section 6.3, we show how to remove the fixed reference by additional self-normalization when the exact normalizing constant is needed.

Equation (5.9) yields an unbiased estimate of $\log z(\cdot)$. However, due to the stochastic approximation, (5.9) has nonignorable variance in the region where not enough a_i are sampled. For smoothness and regularization, we approximate $\log z(\cdot)$ in some parametric family according to the L_2 distance criterion, $\min \int_0^1 \left(\log z(\lambda) - \left(\beta_0 \lambda + \sum_{j=1}^J \beta_j \gamma_j(\lambda)\right)\right)^2 da$. We compute this objective function on a uniform grid with length I: $\{\lambda_i^* = i/I, 1 \leq i \leq I\}$, compute each $\log z(\lambda_i^*)$ using estimate (5.9), and solve the least squares regression

$$\beta_z = \arg\min_{\beta} \sum_{i=1}^{I} \left(\log z(\lambda_i^*) - \left(\beta_0 \lambda_i^* + \sum_{j=1}^{J} \beta_j \gamma_j(\lambda_i^*)) \right) \right)^2 . \tag{5.10}$$

This parametric estimation serves two goals. First, it provides us with a functional form for the prior which we use in the next step to adaptively modify the sampling distribution. Second, the regression estimate (5.10) smooths finite sample noise in (5.9), which is a bias-variance tradeoff.

We update the functional form of the pseudo-prior $\beta_c \coloneqq \beta_z$, or equivalently $c(\cdot) \coloneqq z(\cdot)$.

Step 4. Diagnostics, stopping condition, and mixing. The marginal distribution of a from the sampling distribution (5.5) satisfies $p(a) = z(\lambda)/c(\lambda), \lambda = f(a)$. If $z(\lambda)$ were accurately computed, one step adaptation $c(a) \coloneqq z(a)$ would result in a uniform marginal distribution on a, which is the basis of diagnostics.

Notably, the sampled marginal density $p(a)$ can be estimated as a normalizing constant $p(a) = \int_\Theta q(\theta, f(a)) c^{-1}(f(a)) d\theta$, thus we use a similar estimate as (5.9), only modifying the gradient $U_{(i)}$ by

$$U_{(i)}^p = \frac{\sum_{j:a_j=a_{(i)}} \frac{\partial}{\partial a} \left(\log q(\theta, f(a)) - \log c(f(a)) \right) \Big|_{\theta_j, a_j}}{\sum_{a_j=a_{(i)}} 1}. \tag{5.11}$$

The sample estimates of $z(\lambda)$ and $p(\lambda)$ possess finite sample Monte Carlo error and are prone to over-extrapolation in regions of few a draws. Therefore, we repeat Steps 1–3 until $p(a)$ is "functionally close enough" to a uniform density. However, running until the complete uniformity is both in practice inefficient and in theory unlikely to be obtained as the actual log normalizing constant $z()$ will not fall into the parametric family exactly.

Our adaptation step $z \to c$ can be viewed as an importance sampling procedure from the joint proposal $c(f(a))^{-1}q(\theta, f(a))$ to the joint target $z(f(a))^{-1}q(\theta, f(a))$. The importance ratio is $r(a) = c(f(a))/z(f(a)) = 1/p(a)$, which only depends on the marginal of a, and the normalizing constant estimate (5.9) can be equivalently expressed by the importance sampling estimate $z(\lambda)^{-1} = c(\lambda)^{-1}r(a)$ when the the marginal $p(a)$ is estimated from path sampling.

To assess the accuracy of the final estimate, we use a Pareto-$\hat{k}$ diagnostic adapted from Pareto smoothed importance sampling (PSIS, Vehtari et al., 2019b). We fit the importance ratio $r_i = 1/p(a_i)$ in a generalized Pareto distribution, estimating its right tail shape parameter $\hat{k}$. As already applied in other computation diagnostics (e.g., Yao et al., 2018b), $\hat{k}$ quantifies the Renyi-divergence between the sampled density $c(a)^{-1}q(\theta, a)$ and the target $z(a)^{-1}q(\theta, a)$ that has a uniform marginal on a.

When $\hat{k} < 0.7$, the normalizing constant $z(\lambda)$, viewed pointwise as an importance sampling estimate, is ensured to be reliable with a practical number of simulation draws, and we terminate sampling. The $\hat{k}$ threshold can be chosen smaller to make the decision more conservative.

Otherwise, we perform further sampling with the updated pseudo prior c. Crucially, the path sampling estimate (5.9) is always unbiased for the log normalization constant under any sampling distribution as long as $\theta|a$ is left invariant. Thus, we save all previously sampled draws $\{a_s, \theta_s\}$, and mix them with the newly sampled draws in the normalizing constant estimation (5.9) during each adaption. This remixing step corresponds to a divide-and-conquer strategy that we will further exploit to sample from a metastabe distribution with entropic barriers (Section 6.3).

5.3 Related work: from importance sampling to bridge sampling to Rao-Blackwellization to path sampling to adaptive path sampling

There is a large literature on methods for computing or approximating normalizing constants that cannot be evaluated in closed form. We refer to Gelman and Meng (1998) and Lelievre et al. (2010) for comprehensive reviews on normalizing constants.

Adaptive importance sampling. The basic importance sampling strategy (5.4) treats the normalizing constant $z(\lambda)$ as a conditional expectation with respect to the random variable θ, whose distribution is parameterized by λ. Chatterjee and Diaconis (2018) proved that under certain conditions that the number of simulation draws required for the importance sampling estimate (5.4) of $z(\lambda)$ to have small error with high probability is roughly $\exp(\mathrm{KL}(\pi_\lambda||\pi_{\lambda_0}))$.

When this KL gap is too large, a remedy is to add more *discrete* ladders $\lambda_0 < \lambda_1 < \ldots < \lambda_K = \lambda$, and use adaptive importance sampling. At the $(j+1)$-th time, we sample $\theta_{j1,\ldots,jS}$ from π_{λ_j}, and the importance sampling estimate gives $z_{\lambda_{j+1}}/z_{\lambda_j} = 1/S \sum_{i=1}^{S} q(\theta_{j,i}, \lambda_{j+1})/q(\theta_{ji}, \lambda_j)$. The final estimation of normalizing constant is

$$\frac{z_{\lambda_K}}{z_{\lambda_0}} = \prod_{j=0}^{k-1} \frac{z_{\lambda_{j+1}}}{z_{\lambda_j}} \approx \prod_{j=0}^{k-1} \left(1/S \sum_{i=1}^{S} q(\theta_{ji}, \lambda_{j+1})/q(\theta_{ji}, \lambda_j) \right). \tag{5.12}$$

Bridge sampling. The importance sampling is restricted to sampling from the conditional density $\theta|\lambda$ for a constant λ at each run—a slice in the (θ, λ) joint space. Bridge sampling simultaneously draws $\theta_{j1,\ldots,jS_j}$ from π_{λ_j} and $\theta_{(j+1)1,\ldots,(j+1)S_{j+1}}$ from $\pi_{\lambda_{j+1}}$. Given a sequence of integrable function $\{\alpha_j(\cdot)\}_{j=1}^{J}$, we estimate the ratio of normalizing constant via the bridge sampling (Meng and Wong, 1996) estimate:

$$\frac{z_{\lambda_{j+1}}}{z_{\lambda_j}} = \frac{\mathbb{E}_{\pi_{\lambda_j}} \left(\alpha_j(\theta) q(\theta, \lambda_{j+1}) \right)}{\mathbb{E}_{\pi_{\lambda_{j+1}}} \left(\alpha_j(\theta) q(\theta, \lambda_j) \right)} \approx \frac{\sum_{i=1}^{S_j} q(\theta_{ji}, \lambda_{j+1}) \alpha_j(\theta_{ji})/S_j}{\sum_{i=1}^{S_{j+1}} q(\theta_{(j+1)i}, \lambda_j) \alpha_j(\theta_{(j+1)i})/S_{j+1}}. \tag{5.13}$$

Under some independence assumptions, the optimal choice of $\alpha_j(\theta)$ to minimize the variance of multistage bridge sampling estimate (5.13) is $\alpha_j^{\mathrm{opt}}(\cdot) = \frac{n_j z_j^{-1}}{\sum_m n_m z_m^{-1} q_m(\cdot)}$ (Meng and Wong, 1996; Shirts and Chodera, 2008).

Multistate Bridge sampling and Rao-Blackwellization. When some λ_k are rarely seen, the inverse probability weighting is unstable. Motivated by the identity $p(\lambda) = \mathbb{E}_\theta\, p(\lambda|\theta)$, Carlson et al. (2016) proposed a Rao-Blackwellized (Robert and Casella, 2013) estimate of the normalizing constant, essentially replacing the empirical marginal probability mass function $\Pr(\lambda)$ by a Rao-

Blackwellized estimate $\Pr(\lambda = \lambda_k) = \sum_{s=1}^{S} (q(\theta_s, \lambda_k)/\sum_{k'} q(\theta_s, \lambda_{k'}))$. We now show that this this estimate is equivalent to multistate bridge sampling:

Theorem 5.1 (equivalence between Rao-Blackwellization and multistate bridge sampling)**.** *The Rao-Blackwellized estimate can be derived from multistate bridge sampling* (5.13) *by choosing*

$$\alpha_{ij}(\theta) = \frac{n_j \hat{z}_j^{-1}}{\sum_m c_m^{-1} q_m(\theta)},$$

which is further an empirical estimate of the optimal bridge sampling functions.

Proof. Let $q_i = q(\theta|\lambda_i)$ be the unnormalized density at a sequence of temperatures λ_i, $i = 1, \dots I$, and $Z = (z_2, \dots, z_I)^T$ the (ratio of) normalizing constants (assuming $z_1 = 1$). With an arbitrary list of $\Theta \to \mathbb{R}$ functions $\alpha_{i,j}$, we define A an $(I-1) \times (I-1)$ matrix, and B an $(I-1)$ vector:

$$A = \begin{bmatrix} a_2 & -b_{23} & \dots & -b_{2I} \\ -b_{32} & a_3 & \dots & -b_{3I} \\ \dots & \dots & \dots & \dots \\ -b_{I2} & -b_{I3} & \dots & a_I \end{bmatrix}, \quad B = \begin{bmatrix} b_{21} \\ b_{31} \\ \vdots \\ b_{I1} \end{bmatrix},$$

with each entry a, b a shorthand for

$$b_{ij} = \mathbb{E}_{\pi_j}(\alpha_{ij} q_i), \quad a_i = \sum_{j=1, j\neq i}^{I} \mathbb{E}_{\pi_i}(\alpha_{ij} q_j).$$

In discrete tempering, each time we sample from $q(\theta, \lambda) = \frac{1}{c(\lambda)} q(\theta, \lambda)$. Since λ is discrete here, we denote $c_m = c(\lambda_m)$ and $q_m = q(\theta, \lambda = \lambda_m)$, both of which are given in each adaptation.

We rearranged the multistate bridge sampling estimates $z_i b_{ji} = z_j b_{ij}$, $\forall i, j$ into a matrix form $AZ = B$. Shirts and Chodera (2008) showed that the optimal sequence of functions that minimizes the variance of the estimated Z is $\alpha_{ij}(\theta) = n_j z_j^{-1} / \sum_m n_m z_m^{-1} q_m(\theta)$, which in practice, starting from some initial guess $\hat{z}$, this can be approximated by $\alpha_{ij}(\theta) = n_j \hat{z}_j^{-1} / \sum_m n_m \hat{z}_m^{-1} q_m(\theta)$.

Denote $S(\theta) = \sum_m c_m q_m(\theta)$, then $\alpha_{ij}(\theta) = \frac{n_j \hat{z}_j^{-1}}{S(\theta)}$. We estimate the matrices A and B by their

empirical means.

$$\hat{a}_i \hat{z}_i = n_i^{-1} \sum_{k=1}^{n_i} \hat{z}_i \sum_{j=1, j\neq i}^{I} \alpha_{ij}(\theta_{i,k}) q_j(\theta_{i,k}) = \hat{z}_i n_i^{-1} \sum_{k=1}^{n_i} \frac{\sum_{j=1,j\neq i}^{I} n_j \hat{z}_j^{-1} q_j(\theta)}{S(\theta_{i,k})} = \hat{z}_i - \sum_{k=1}^{n_i} \frac{q_i(\theta_{i,k})}{S(\theta_{i,k})}.$$

$$\sum_{j=1,j\neq i}^{I} \hat{b}_{ij} \hat{z}_j = \sum_{j=1,j\neq i}^{I} \hat{z}_j n_i^{-j} \sum_{k=1}^{n_j} \frac{n_j \hat{z}_j^{-1} q_i(\theta_{j,k})}{S(\theta_{j,k})} = \sum_{j=1,j\neq i}^{I} \sum_{k=1}^{n_i} \frac{q_i(\theta_{j,k})}{S(\theta_{j,k})}.$$

Combining these two parts, we obtain the final estimate

$$z_i = \sum_{j=1,j\neq i}^{I} \sum_{k=1}^{n_i} \frac{q_i(\theta_{j,k})}{S(\theta_{j,k})} + \sum_{k=1}^{n_i} \frac{q_i(\theta_{i,k})}{S(\theta_{i,k})} = \sum_{m=1}^{N} \frac{q_i(\theta_m)}{\sum_{j=1}^{I} c_j q_j(\theta_m)},$$

which is identical to the Rao-Blackwellized estimates.

Non-equilibrium methods. The importance sampling and bridge sampling estimates requires $n_j > 1$ simulation draws from each $\theta|\lambda_j$. In non-equilibrium methods, we start by a simulation draw θ_0 from equilibrium $\theta|\lambda_0$, evolve it through a sequence of transitions that keeps $\pi_k, k = 1, \ldots, K$ invariant at each step, and collect one non-equilibrium trajectory $(\theta_0, \ldots, \theta_{K-1})$. Notably, we do not draw from π_K directly, and $\theta_j, j \geq 1$ is in general not π_j distributed. We still obtain an unbiased estimate (Jarzynski, 1997; Neal, 2001):

$$\frac{z_{\lambda_K}}{z_{\lambda_0}} = \mathbb{E}\left(\exp \mathcal{W}(\theta_0, \ldots, \theta_{K-1})\right), \quad \mathcal{W}(\theta_0, \ldots, \theta_{K-1}) = \log \prod_{j=0}^{k-1} \frac{q(\theta_j, \lambda_{j+1})}{q(\theta_j, \lambda_j)}. \tag{5.14}$$

where the expectation is over all initial draws and trajectories.

Thermodynamic integration. The path sampling estimate (5.9) is unbiased for the log normalizing constant (ratios) $\log z$, while other importance sampling based algorithms are unbiased in the scale of the normalizing constant z. In our adaptive procedure, since we update the *logarithm* of the joint density and compute its gradient in the next Hamiltonian Monte Carlo run, the unbiasedness of $\log z$ is more relevant. In statistical physics, the $\mathcal{W}$ quantity in (5.14) is interpreted as virtual work induced on the system. Jensen's inequality leads to $\mathbb{E}\,\mathcal{W} \geq \log z(\lambda_K) - \log z(\lambda_0)$. This is

a microscopic analogy of the second law of thermodynamics: the work entered in the system is always larger than the free energy change, unless the switching is processed infinitely slow, which corresponds to the thermodynamic integration.

The thermodynamic integration equality (5.7) was first introduced by Kirkwood (1935) in statistical physics, and further refined or applied by Ogata (1989); Neal (1993); Gelman and Meng (1998); Rischard et al. (2018) in the context of normalizing constant computing. However, the typical use of thermodynamic integration requires discretizing λ into a fixed quadrature ladder $\lambda_1 < \ldots < \lambda_K$, on which the gradient U_j is computed using many draws from $\theta|\lambda_j$. These ladders involve further manual tuning (Schlitter, 1991; Blondel, 2004) to control the variance of U, and the discretization error (bias) in the numerical integration is non-vanishing in a finite discrete ladder, to a large extent compromising the unbiasedness property of $\log z$ estimation.

Our present chapter also extends the general discussion of path sampling in Gelman and Meng (1998), with added steps on parametric regularization, iterative adaptations, and diagnostics. Essentially we use stochastic approximation (Robbins and Monro, 1951) to compute the the pointwise gradient (5.7). We employ a carefully designed link function f that allows direct access to simulation draws from some chosen conditional distributions $\theta|\lambda$, while also eliminates the discretization bias. These extensions facilitate the continuous tempering scheme by providing direct access to draws from the target distribution.

Path sampling as the continuous limit of importance sampling and Taylor expansion. In Section 5.1, we describe two separate approaches: the importance sampling estimate (5.4) and Taylor series expansion (5.3). They reach the same first order limit when the proposal is infinitely close to the target. That is, for any fixed λ_0, as $\delta = |\lambda_1 - \lambda_0| \to 0$,

$$\frac{1}{\delta} \log \mathbb{E}_{\lambda_0} \left(\frac{q(\theta|\lambda_1)}{q(\theta|\lambda_0)} \right) = \int_\Theta \frac{\partial}{\partial \lambda} \log q(\theta|\lambda_0) p(\theta|\lambda_0) d\theta + o(1) = \frac{1}{\delta} \mathbb{E}_{\lambda_0} \left(\log \frac{q(\theta|\lambda_1)}{q(\theta|\lambda_0)} \right).$$

The path sampling estimate $\int_{\lambda_0}^{\lambda_1} \int_\Theta \frac{\partial}{\partial \lambda} \log q(\theta|\lambda_l) p(\theta|\lambda_l) d\theta d\lambda$ that we employ is the integral of the dominate term in the middle. In this sense, path sampling is the continuous limit of both the

importance sampling and the Taylor expansion approach.

More generally, thermodynamic integration (5.7) can be viewed as the $K \to \infty$ limit of the equilibrium bridge sampling, Rao-Blackwellization, and annealed importance sampling, but without having to fit the conditional model infinitely many times. We will further elaborate in the simulating tempering context that such continuous extension is desired, as otherwise the necessary number of interpolating temperatures K soon blows up when the dimension of θ increases.

Theorem 5.2 (Path sampling as the continuous limit of importance sampling and Taylor expansion)**.** *Path sampling can be viewed as the continuous limit of bridge sampling* (5.13) *and annealed importance sampling* (5.14) *when the intermediate states* ($0 = \lambda_0 < \lambda_1 < \ldots < \lambda_{L+1} = 1$) *is infinitely dense, such that* $\max_l \delta_l \to 0$*, where* $\delta_l = \lambda_{l+1} - \lambda_l$ *is the neighboring spacing.*

Proof. Bridge sampling and annealed importance sampling work in the scale of the normalizing constant z, essentially computing $z(1)/z(0)$ by $\prod_{l=0}^{L}(z(l+1)/z(l))$, or equivalently, $\log z(1)/z(0) = \sum_{l=0}^{L} \log z(l) - \log z(l-1)$. Further, both bridge sampling and annealed importance sampling are based on importance sampling identity (with potential refinement of more intermediate states in bridge sampling):

$$\frac{z(l)}{z(l-1)} = \int_\Theta \frac{q(\theta|\lambda_{l+1})}{q(\theta|\lambda_l)} p(\theta|\lambda_l) d\theta.$$

In general, $\log \mathbb{E}\left(q(\theta|\lambda_{l+1})/q(\theta|\lambda_l)\right) \neq \mathbb{E}\left(\log q(\theta|\lambda_{l+1}) - \log q(\theta|\lambda_l)\right)$, where the expectation is taken over $\theta \sim p(\theta|\lambda_l)$. However, such difference will be be approaching zero when we have fine ladder.

For a fixed λ_l, let

$$G_l(\xi) = \log \int_\Theta \frac{q(\theta|\lambda_l + \xi)}{q(\theta|\lambda_l)} p(\theta|\lambda_l) d\theta.$$

It satisfies $G_l(0) = 0$, and its derivative is

$$G_l'(\xi) = \frac{d}{d\xi}\left(\log \int_\Theta \frac{q(\theta|\lambda_l + \xi)}{q(\theta|\lambda_l)} p(\theta|\lambda_l) d\theta\right) = \frac{z(l)}{z(l+\xi)}\left(\int_\Theta \frac{\partial}{\partial \xi} \frac{q(\theta|\lambda_l + \xi)}{q(\theta|\lambda_l)} p(\theta|\lambda_l) d\theta\right).$$

At $\xi = 0$, $G_l'(0)$ becomes identical to the path sampling gradient in (5.7), as

$$G_l'(0) = \int_\Theta \frac{\partial}{\partial \lambda} \log q(\theta|\lambda_l) p(\theta|\lambda_l) d\theta.$$

By Taylor series expansion, $G_l(\xi) = G_l(0) + \xi \int_\Theta \frac{\partial}{\partial \lambda} \log q(\theta|\lambda_l) p(\theta|\lambda_l) d\theta + o(\xi)$, Hence, in the limit as $\max_l \delta_l \to 0$, the importance sampling based estimate can be rearranged into

$$\begin{aligned}
\log z(1)/z(0) &= \sum_{l=0}^{L} G_l(\delta_l) \\
&= \sum_{l=0}^{L} \left(\delta_l \int_\Theta \frac{\partial}{\partial \lambda} \log q(\theta|\lambda_l) p(\theta|\lambda_l) d\theta + o(\delta_l) \right) \\
&= \int_0^1 \int_\Theta \frac{\partial}{\partial \lambda} \log q(\theta|\lambda_l) p(\theta|\lambda_l) d\theta d\lambda + o(1),
\end{aligned}$$

where the dominant term equals the path sampling estimate, and the remainder approaches 0 in the dense limit $\delta_l \to 0, \forall l$ since $\sum_l \delta_l = 1$.

5.4 Marginal density estimation: No, we do not need kernel density estimation for MCMC samples

Markov chain Monte Carlo (MCMC) methods in Bayesian computation are often framed as a method for computing integrals, so that the simulation draws are merely an intermediate step to evaluate some expectation with respect to the posterior distributions, whose convergence and mixing behavior is well studied in various ergodic theories.

For statisticians there is an extra step to extrapolate sample to the population, as adjusting for finite sample variation (in data) is standard practice: We fit hierarchical models to grouped data even when the sample average within each group is unbiased and consistent. We use regression to adjust covariate mismatch in randomized trials even when treatment is randomly assigned. Likewise, here we would like to make a robust inference on the posterior distribution, with a hope to reduce variance from finite Monte Carlo draws (O'Hagan, 1987).

In this present chapter, we focus on one particular quantity, the marginal density: the normalization constant of the joint density, integrating out all remaining dimensions. Given simulation draws $(\theta_1, \ldots, \theta_S)$, where each draw $\theta_s \in \mathbb{R}^d$ is a d-dimension parameter $\theta_s = (\theta_{s1}, \ldots, \theta_{sd})$, we would like to know the marginal density of a particular parameter. Without loss of generality we consider the first coordinate

$$\pi(\theta_1) = \int_{\mathbb{R}_{d-1}} p(\theta) d\theta_2 \ldots d\theta_d.$$

We have suppressed the notation dependence on data y. It is possible to extend the discussion to a low dimensional joint density but we will start by one dimension. We often know the joint density $q(\theta_1, \ldots, \theta_d) \propto p$ up to a normalization constant.

Related methods. *Kernel methods* are widely used non-parametric density estimate of the form

$$\hat{\pi}_1(\theta^*) = \frac{1}{Sh} \sum_{s=1}^{S} K(\frac{\theta^* - \theta_{s1}}{h}).$$

The kernel methods are applicable even when the density form of q is not known. On the other hand, it involves intensive manual tuning in the kernel choice K and bandwidth h. In addition, it ignores all other dimensions $\theta_2, \ldots, \theta_d$ unused.

Rao-blackwised kernel density estimation is based on the identity $p(\theta_1) = \mathbb{E}\, p(\theta_1|\theta_{-1})$. If we know $p(\theta_1|\theta_{-1})$ exactly–including the normalization constant, we can compute the *conditional marginal density estimator* (Gelfand et al., 1992b),

$$\hat{\pi}_1(\theta^*) = \frac{1}{S} \sum_{s=1}^{S} p(\theta^*|\theta_{-1s})$$

When the conditional density is unknown, Gelfand and Smith (1990) suggested to estimated it first from kernel densities.

$$\hat{\pi}_1(\theta^*) = \frac{1}{S} \sum_{s=1}^{S} \mathbb{E}\, \pi_{\text{knrnel}}(\theta_1|\theta_{-1})$$

Importance weighted conditional density estimator (Chen, 1994) does not require the closed form

conditional density. Using the identity

$$\pi(\theta_1^*) = \int \frac{p(\theta_1^*, \theta_2, \ldots, \theta_S) w(\theta_1|\theta_{-1})}{p(\theta_1, \ldots, \theta_d)} p(\theta_1, \ldots, \theta_d) d\theta 1 \ldots d\theta_d$$

where $w(\theta_1^*|\theta_{-1})$ is an arbitrary conditional density that we can evaluate exactly. The sample estimator reads

$$\hat{\pi}(\theta_1^*) = 1/S \sum_{s=1}^{S} w(\theta_{s1}|\theta_{s,-1}) \frac{q(\theta_1^*, \theta_{s2} \ldots, \theta_{sd})}{q(\theta_{s1}, \theta_{s2}, \ldots, \theta_{sd})}$$

In practice, $w(\theta_1^*|\theta_{-1})$ is often chosen to approximate the the unknown conditional density $p(\theta_1^*|\theta_{-1})$.

Proposed method: path sampling estimates. Applying the general path sampling estimate to the marginal density is straightforawd. We summarize the method in Algorithm 5. Here we assume $\theta_1^* > \theta_{(s/2)1}$ and the other half is symmetric. The output is the unnormalized density, which is itself useful for many tasks. When we need the exact marginal density, there is an extra step of self-normalization.

Algorithm 5: Estimate marginal density from MCMC draws and path sampling

Input: S simulation draws $(\theta_s d)_{s=1}^S d = 1^D$, unnormalied joint desntiy $p(\theta_1, \ldots, \theta_D)$, the location at which to evaluate the marginal density θ_1^*

Result: unnormalized marginal density $q(\theta_1)$

1 *sorting.* Sort simulation draws according to the coordinate that we want to evaluate the marginal density. Denote $\theta_{(1)1} < \theta_{(2)1} \cdots < \theta_{(S)1}$ to be the sorted unique value of θ_1;

2 *stochastic approximation.* Compute gradients $U_i(\theta_{i1}) = \frac{\partial}{\partial \theta_1} \log p(\theta_1, \theta_2, \ldots, \theta_d)|_{\theta_{(s)1}, \ldots, \theta_{(s)d}}$;

3 *reference point.* Compute $\theta_0 :=$ Median $(\theta_{s1}) = (\theta_{(s/2)1})$;

4 *quadrature.* Locate θ_1^* in the samples: $\theta_{(i^*)} \leq \theta_1^* < \theta_{(i^*+1)}$ and compute $\log q(\theta_1^*) := \log \frac{\pi(\theta_1^*)}{\pi(\theta_0)} = \sum_{i=s/2}^{i^*-1} \frac{U_i + U_{i+1}}{2} (\theta_{(i+1)1} - \theta_{(i)1}) + \frac{U_{i^*} + U_{i^*+1}}{2} (\theta_1^* - \theta_{(i^*)1})$.

Examples To verify the methods, we start by a multivariate normal example. Suppose the posterior θ is from $\theta \sim \text{MVN}(\mu, \Sigma)$, whose unnormalzied density is $\log q(\theta) = -\frac{1}{2}(\theta - \mu)^T \Sigma^{-1} (\theta - \mu)$. The true marginal distribuion of the first dimension $\pi(\theta_1)$ is $\text{N}(\mu_1, \Sigma_{11})$. In the simulation we fix $\sigma_{11} = 1$ and $\mu = 0$. What enters the path sampling computation is the distribution of $\sum_{i=1}^d \tau_{1i}(\theta_{si})$. Increasing the dimension d is operationally equivalent to relabeling $\sum_{i=1}^d \tau_{1i}(\theta_{si})$ as θ_2, which still

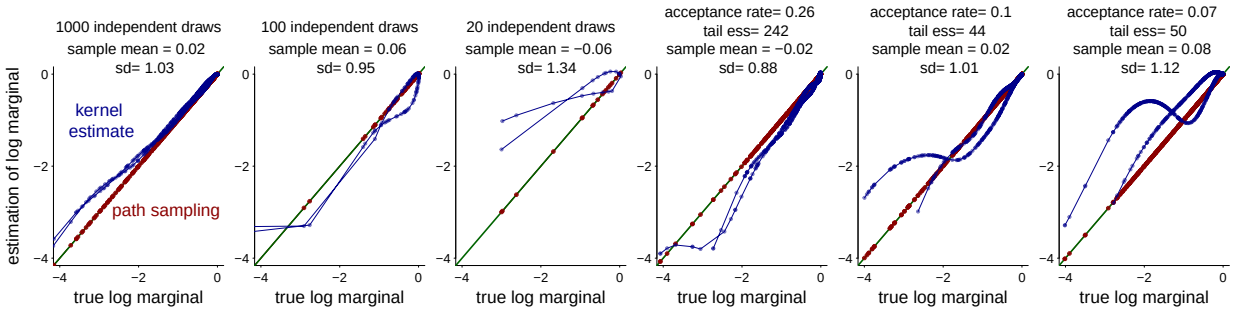


Figure 5.2: When the margin of interest is independentg of others, the path sampling estimation of log marginal density is exact even with a few posterior draws. The empirical density estimation from kernel methods can have various pitfalls even in one dimension, especially in the tail regions and when the Markov chain acceptance rate is low.

admits a jointly multivariate normal density. Therefore, in the next experiment, we fix $d = 2$ and vary σ_2: the marginal standard deviation of θ_2 to simulate the dimension effect. Evidently, this experiment setting is equivalently for path sampling to have an effective dimension $d = 1 + \sigma_2^2$.

When θ_1 is uncorrelated with other dimensions, $u_s = \sigma_1^{-2}(\theta_{s1} - \mu_1)$. Further assuming $\sigma_1 = 1$ and $\mu_1 = 0$, the path sampling estimate becomes

$$-1/2x^2 = \int x dx \approx \sum_s (x_s - x_{s-1}) \frac{x_{s+1} + x_{s+1}}{2}.$$

In contrast, the empirical density estimation from kernel methods has various pitfalls even in one dimension. In the left three panels of Figure 5.2, we vary the independent sample size S from 1000 to 20, and compare the estimated log density with the true log marginal density. Even when $S = 1000$ the tail estimation in kernel methods is far from accurate, while the path sampling estimate remains exact. Furthermore, the empirical density estimation suffers from auto correlation in Markov chains. In panels 4–6 in Figure 5.2, we generate $S = 1000$ draws from random walk Metropolis Hastings (warm-up removed). When the acceptance rate is low, there are many unmoved points in the simulation draws.

Next, we change the variance of θ_2 and the correlation ρ between θ_1 and θ_2. It is equivalent to increase the effective dimension to $1 + \mathrm{Var}(\theta_2)$. We repeat this experiment 50 times and compute the mean squares error (MSE) of the log marginal density estimation $\int |\log \pi_1 \theta - \log \hat{\pi}_1(\theta)|^2 \pi_1(\theta) d\theta$,

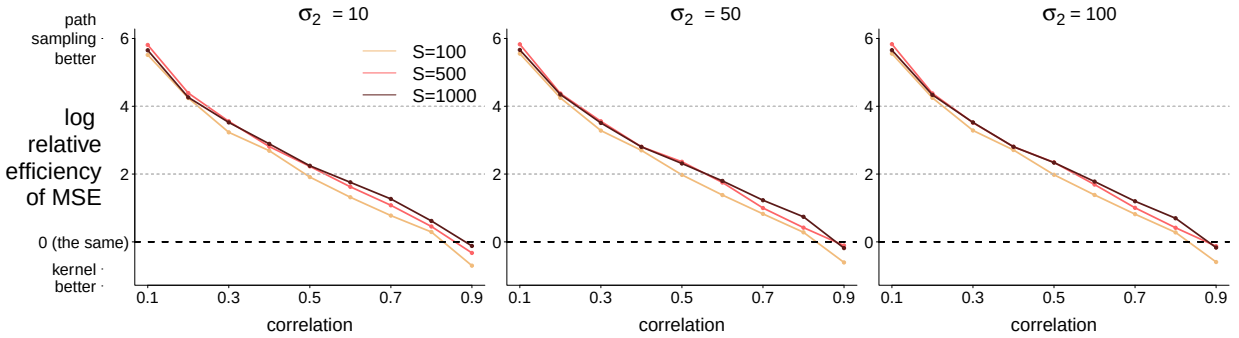


Figure 5.3: We monitor the log relative efficiency (log ratio of MSE of the log marginal densities in empirical and path sampling). Path sampling is more efficient until $\rho = 0.9$. The effective dimension ($d_{\text{eff}} = \sigma_2^2$) plays little role in this comparison.

or the exponential divergence between the true target and the estimated marginal distribution. This error term varies in order of magnitude when we vary σ_2, ρ, and S, hence we define the relative efficiency:

$$\text{relative efficiency} = \frac{\text{MSE of kernel estimate of } \log \pi_1}{\text{MSE of path sampling of } \log \pi_1}.$$

We present the log relative efficiency in Figure 5.3. The path sampling estimate of the marginal density is more accurate until the correlation is too high. The log relative efficiency of 2 is a seven-fold decrease in mean squared error.

In the next chapter, we will apply this better marginal density estimate in the context of simulated tempering and multimodal sampling, and we will further see its benefit over kernel density estimate.

Chapter 6. Continuous tempering using adaptive path sampling[1]

"Our growing freedom from and control over nature rely on a series of stable natural parameters that we tend to take for granted: temperature for example, . . . we can do what we want only so long as we remain marginally enough."

—Slavoj Žižek

6.1 Adaptive continuous tempering: Sample from a multimodal distribution

Given a statistical model, we can evaluate the posterior joint density $p(\theta, y)$= prior×likelihood. In this section, we suppress the dependence on data y and denote the unnormalized posterior distribution from which we want to sample as $q(\theta) \coloneqq p(\theta, y), \theta \in \Theta$. When $q(\theta)$ exhibits severe multimodality, Markov chain Monte Carlo (MCMC) algorithms have difficulty moving between modes. The state-of-the-art dynamic Hamiltonian Monte Carlo sampler for a bimodal density has a mixing rate as slow as random-walk Metropolis (Mangoubi et al., 2018), and even optimal tuning and Riemannian metrics do not help. Although it is possible to evade multimodal sampling using other post-processing and re-weighting strategies (e.g., Yao et al., 2020b), we aim here to sample from the exact posterior density.

To ease the energetic barrier between modes, we consider a distribution bridging between the target $q(\theta)$ and a base distribution $\psi(\theta)$ through a geometrically tempered path:

$$p(\theta|\lambda) = \frac{1}{z(\lambda)} q(\theta)^{\lambda} \psi(\theta)^{1-\lambda},\ \lambda \in [0, 1],\ \theta \in \Theta,$$

where λ is the augmented inverse temperature, $z(\lambda)$ is the normalizing constant $z(\lambda) = \int_{\Theta} \psi(\theta)^{\lambda} q(\theta)^{1-\lambda} d\theta$, and $\psi(\theta)$ is a proper base probability density, typically a simple initial guess or the prior that is easy to sample from. When ψ is known exactly, $z(0)$ is 1. We will discuss the

[1]This chapter is a slight modified version of Yao et al. (2020a).

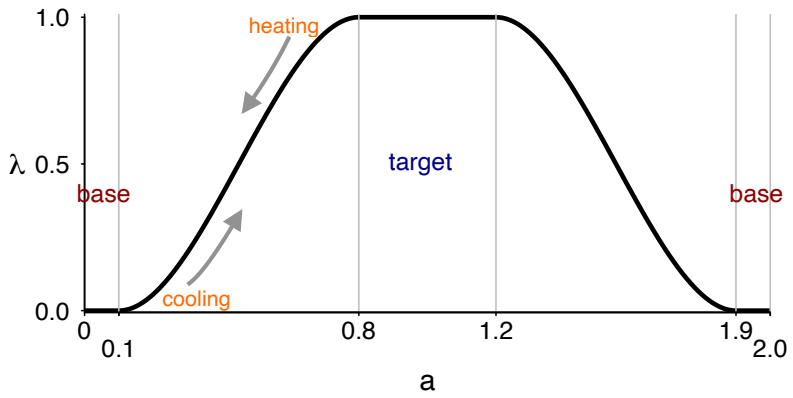


Figure 6.1: *The link function $\lambda = f(a)$. The flat area between 0.8 and 1.2 allows the continuous sampler to have a region where there are exact draws from the target distribution.*

choice of base distribution later. $p(\theta|\lambda)$ is the target density when $\lambda = 1$, and becomes "flattened" for a smaller λ.

To ensure that the joint sampler can access the base and target distributions with nonzero probabilities, we define a link function $\lambda = f(a) : [0,2] \to [0,1]$, that is symmetric $f(a) = f(2-a)$, and flat at two ends: $f(a) = 0$ when $0 \le a \le a_{\min}$ or $2 - a_{\min} \le a \le 2$, and $f(a) = 1$ when $a_{\max} \le a \le 2 - a_{\max}$. This is easy to satisfy using a piecewise polynomial, see Figure 6.1 for an illustration.

$$f(a) = \begin{cases} 0, & 0 \le a < a_{\min}, \\ -2(\frac{a-a_{\min}}{a_{\max}-a_{\min}})^3 + 3(\frac{a-a_{\min}}{a_{\max}-a_{\min}})^2, & a_{\min} \le a < a_{\max}, \\ 1, & a_{\max} \le a < 2 - a_{\max}, \\ -2(\frac{2-a_{\min}-a}{a_{\max}-a_{\min}})^3 + 3(\frac{2-a_{\min}-a}{a_{\max}-a_{\min}})^2, & 2 - a_{\max} \le a < 2 - a_{\min}, \\ 0, & 2 - a_{\min} \le a \le 2. \end{cases} \tag{6.1}$$

It has a continuous first order derivative. In experiments and default software implementation, we set $a_{\min} = 0.1$ and $a_{\max} = 0.8$.

An a-trajectory from 0 to 2 corresponds to a complete λ tour from 0 to 1 (cooling) and back down to 0 (heating). This formulation allows the sampler to cycle back and forth through the space

of λ continuously, while ensuring that some of the simulation draws (those with a between 0.8 and 1.2) are drawn from the exact target distribution with $\lambda = 1$.

To run simulated tempering, we apply the adaptive path sampling (Steps 1–4 in Section 5.2) to the joint distribution,

$$p(\theta, a) \propto \frac{1}{c(f(a))} q(\theta)^{f(a)} \psi(\theta)^{1-f(a)},\ a \in [0,2],\ \theta \in \Theta.$$

During each adaptation, we sample from this joint density, use all existing draws (including from previous adaptions) to obtain the path sampling estimated log normalization constant $\log z$, parametrically regularize it, and update $\log c$ by $\log \hat{z}$. Because $f()$ is designed symmetric, $z(f(a)) = z(f(2-a))$, we flip all a_s to be $2 - a_s$ for all $a_s > 2$ during $\log z$ estimation (5.9).

In addition, the pointwise gradient U in (5.8) is further simplified to be

$$U_{(i)} = \frac{\sum_{j:a_j = a_{(i)}} f'(a_{(i)}) \Big(\log q(\theta_j) - \log \psi(\theta_j) \Big)}{\sum_{j:a_j = a_{(i)}} 1}.$$

If the base density $\psi(\theta)$ is chosen to be the prior in the model, this gradient is simply the product of $f'(a_{(i)})$ and the log likelihood.

When the marginal distribution of a is close to uniform, which is monitored by the Pareto-$\hat{k}$ diagnostic, a path has been constructed from the base to the target. We then collect all draws in the final adaptation with temperature $\lambda = 1$, i.e. $\{\theta_i \mid f(a_i) = 1\}$. These are the desired draws from the target distribution $q(\theta)$. As a byproduct, we obtain the log normalization constant estimate $\log z$, and $z(1)$ equals the marginal likelihood if ψ is chosen as the prior. The full tempering method is summarized in Algorithm 6.

6.2 On the choice of temperature prior margin

In path sampling, the final marginal distribution of a relies on user specification. By default we use $z(\cdot) \to c(\cdot)$, which enforce a uniform marginal distribution. More generally, by updating

Algorithm 6: Continuous tempering with path sampling.

Input: $\psi(\theta), q(\theta)$: the base and (unnormalized) target density;
Output: draws from the target distribution; $\log z(\cdot)$: log normalizing constant.

Initialize pseudo-prior $\log c(\cdot) = 0$;
repeat
  Sample $\{a_s, \theta_s\}_{s=1,\ldots,S}$ from the joint density $q(\theta, a) = \frac{1}{c(a)}\psi(\theta)^{f(a)}q(\theta)^{(1-f(a))}$;
  Flip $a_s := 2 - a_s$ all $a_s > 1$;
  Estimate $\log z(\cdot)$ by path sampling (5.9), from draws in all adaptations;
  Estimate $\log p(\cdot)$ by path sampling (5.9) and gradients (5.11), from the current adaptation;
  Update $\log c(\cdot) \leftarrow \log z(\cdot)$;
until *Pareto-$\hat{k}$, the estimated tail shape parameter of the ratios $1/p(a_s)$, is smaller than 0.7.*;
Collecting sample $\{\theta_i | f(a_i) = 1\}$ as posterior draws of target distributions.

$c(\cdot) \leftarrow z(\cdot)/p^{\text{prior}}(\cdot)$, the final marginal distribution of a will approach p^{prior}. In this section we discuss the choice of p^{prior} beyond uniformity.

The choice of p^{prior} is subject to a efficiency-robustness trade-off. There are three separate goals to pursue via prior tuning:

1. **Robustness.** Because a uniform a ensures the Markov chain has explored the whole temperature space, it is a conservative choice and we use for default in adaptations.

$$p(a) \propto 1.$$

2. **Minimal variance of log z.** On then other end of the spectrum, we can ask for efficiency. Gelman and Meng (1998) proved that the generalized Jeffreys prior minimizes the variance of estimated log normalizing constant.

$$p^{\text{opt}}(\lambda) \propto \sqrt{\mathbb{E}_{\theta|\lambda}\, U^2(\theta, \lambda)}.$$

   where $U(\theta, \lambda) = \frac{\partial}{\partial \lambda} \log q(\theta, \lambda)$.

3. **Smooth transition in the joint sampling.** From the perspective of successful sampling, we

could ask for

$$\mathrm{KL}\Big(\pi_a, \pi_{a+\delta a}\Big) \approx \text{constant},$$

which is related to the constant acceptance rate in discrete Gibbs update (when the discrete temperature update is restricted to either a one-step jump $\lambda_k \to \lambda_{k\pm1}$ or remain unchanged, This constant acceptance rate is often used as a tuning target in discrete tempering (Geyer and Thompson, 1995). The next proposition gives an closed form optimal prior that ensures this constant KL gap.

Theorem 6.1 (the optimal temperature margin that ensures smooth transition)**.** *The desired prior to achieve a smooth KL gap is*

$$p^{\mathrm{opt}}(a) \propto \frac{1}{f'(a)}\sqrt{\mathrm{Var}_{\theta\sim p(\theta|a)}U(\theta,a)},\ \forall a_{\min} < a < a_{\max}.$$

Proof. It is easy to verify that

$$\mathrm{KL}\Big(\pi_a, \pi_{a+\delta a}\Big) = \int \log\big(\frac{\pi_a}{\pi_{a+\delta a}}\big)d\pi_a = \frac{1}{2}(\delta a)^2\Big(\frac{d^2}{da^2}\log z(a) - \frac{f''(a)}{f'(a)}\frac{d}{da}\log(z(a))\Big) + o(\delta a)^2.$$

Assuming we have already sampled from the joint stationary distribution, the gap between two neighboring order statistics reflects how dense the local density is, i.e., $\delta a \propto 1/p(a)$.

Further, the two derivative terms can be expressed by expectations,

$$\frac{d}{da}\log(z(a)) = f'(a)\mathrm{E}_{\theta\sim p(\theta|a)}\big[\log(\psi) - \log(\phi)\big].$$

$$\begin{aligned}\frac{d^2}{da^2}\log(z(a)) =& f''(a)\mathrm{E}_{\theta\sim p(\theta|a)}\big[\log(\psi) - \log(q)\big] + f'(a)^2\mathrm{E}_{\theta\sim p(\theta|a)}\Big(\big(\log(\psi) - \log(q)\big)^2\Big)\\ &- \Big(f'(a)\mathrm{E}_{\theta\sim p(\theta|a)}\left(\log(\psi) - \log(q)\right)\Big)^2,\end{aligned}$$

which further simplifies to

$$\begin{aligned}
&\frac{d^2}{da^2}\log(z(a)) - \frac{f''(a)}{f'(a)}\frac{d}{da}\log(z(a)) \\
=&f'(a)^2\mathrm{E}_{\theta\sim p(\theta|a)}\left(\left(\log(\psi)-\log(q)\right)^2\right) - \left(f'(a)\mathrm{E}_{\theta\sim p(\theta|a)}\left(\log(\psi)-\log(q)\right)\right)^2.
\end{aligned}$$

Put all together, the constant KL gap will be achieved by

$$\begin{aligned}
p(a) \propto&\left(\frac{d^2}{da^2}\log z(a) - \frac{f''(a)}{f'(a)}\frac{d}{da}\log(z(a))\right)^{\frac{1}{2}} \\
=&\frac{1}{f'(a)}\left(\mathrm{Var}_{\theta\sim p(\theta|a)}\left(\log(\psi)-\log(q)\right)\right)^{\frac{1}{2}}.
\end{aligned}$$

It is also evident that under this prior, both $\mathrm{KL}\left(\pi_a, \pi_{a+\delta a}\right)$ and the reserve jump $\mathrm{KL}\left(\pi_{a+\delta a}, \pi_a\right)$ approximates (different) constants along the trajectory.

Due to dependence on the unknown normalizing constant (and higher orders), these two efficiency-optimal priors require additional tuning and adaptations. In general, we still prefer to use the simple uniform margin for robustness. Nevertheless, our method enables any user specific-prior choice, and in Section 6.5, we illustrate that this adaptively estimated and assigned efficiency-optimal prior further reduces the variance in implicit divide and conquer scheme.

6.3 Implicit divide and conquer in a metastable distribution

The proposed continuous simulated tempering algorithm in Section 6.1 alleviates metastablility in energetic barriers. However, tempering is not effective at overcoming purely entropic barriers. In such cases, instead of augmenting the density with an additional temperature variable, we increase the density in the bottleneck region to encourage transitions between metastable regions.

In many models, it is known that certain marginal distributions are problematic. For example, in hierarchical models, the centered parameterization effectively creates left truncation on the group level standard deviation τ, as the sampler hardly enters the $\tau \approx 0$ region. We denote the joint distribution $q(\theta, \tau)$, where τ is the targeted problematic margin and θ is all remaining parameters.

In other cases, these problematic marginals can be identified by various MCMC diagnostics such as trace plots.

A conservative solution is to sample τ first from some "wider" proposal distribution, then sample θ given τ in a Gibbs fashion, and finally adjust for the extra wide proposal by importance sampling. Here we propose an alternative strategy based on path sampling that does not require the Gibbs step or any closed form conditional density. The method is readily available using the joint sampler in Stan.

We first sample *some* simulation draws from the joint posterior distribution of all parameters $\tilde{q}(\theta, \tau) = q(\theta, \tau)$, without requiring a complete exploration. The marginal density of τ in the original model, $p(\tau)$, is the normalization constant $p(\tau) \propto \int_\Theta q(\theta, \tau) d\theta$. Hence, we compute $\log p(\tau)$ over all sampled τ using path sampling formula (5.9), and modify the sampling distribution by adding the bias term $\log \tilde{q}(\theta, \tau) \coloneqq \log q(\theta, \tau) + \log \left(p^{\text{targ}}(\tau)/p(\tau)\right)$, where $p^{\text{targ}}(\tau)$ is a desired marginal density. It can be fixed at any distribution that covers the true posterior marginal of τ, such as its prior. we discuss other choices in the end of this section.

We then sample new draws (θ, τ) from this adapted sampling distribution. Since we only change the marginal distribution of τ between adaptations, the conditional densities $\theta|\tau$ remain invariant. We mix the new sample with the ones from all previous adaptations and use this cumulative sample to estimate the aggregated marginal density $p(\tau)$ using path sampling at each adaptation. We iterate the above procedure until we reach approximate marginal convergence to $p^{\text{targ}}(\tau)$, which is further quantified by the Pareto-$\hat{k}$ diagnostic.

τ is often undersampled in some regions in early iterations. In these regions, the path sampling estimated $p(\tau)$ is less reliable, and the $\log \left(p^{\text{targ}}(\tau)/p(\tau)\right)$ term will cause the sampler to focus in these undersampled regions on the next iteration (and mostly ignore those regions that were already sufficiently sampled). This adaptive behavior is what makes this algorithm an implicit divide-and-conquer-type procedure. Algorithm 7 shows the complete setup for this procedure.

Once we have obtained our estimate of $p(\tau)$ from the divide-and-conquer algorithm, we can use importance sampling to compute marginal posterior expectations and quantiles. Alternatively,

the posterior expectation of any integrable $h(\tau)$ is the normalizing constant of $h(\tau)p(\tau)$, which we can evaluate using path sampling and (one-dimensional) quadrature (5.9). In our experiments, we find the latter approach to be more robust. Furthermore, as a byproduct of our marginal density estimate, we can evaluate the marginal distribution function out to arbitrary distances. This allows us to estimate quantiles with extremely small tail probability that may be more difficult to estimate with Monte Carlo draws.

In addition, the path sampling estimate of $p(\tau)$ can be used to diagnose poor sampling behavior in standard HMC by comparing this estimate with the obtained empirical distribution.

Algorithm 7: Implicit divide-and-conquer scheme for metastable distributions

Input: $q(\tau, \theta)$: the (unnormalized) joint density; τ: the problematic margin; θ: all remaining parameters; $p^{\text{targ}}(\tau)$: the targeted marginal of τ.

Output: $p(\tau)$: marginal density of τ in the original joint density;

Initialize sampling distribution $\tilde{q} = q(\theta, \tau)$; j=1;

repeat

  Generate sample $\{\tau_s, \theta_s\}$ from $\tilde{q}(\tau, \theta)$;

  Mix these draws with all previous adaptations $\{\tau_s, \theta_s\}_{s=1}^{S}$;

  Compute $p(\tau)$, the marginal density of $q(\tau, \theta)$, using path sampling (5.9), gradients (5.11) and all draws;

  Smooth estimated $p(\tau)$ by regression (5.10), and record $p_j(\cdot) \coloneqq p(\cdot)$;

  Update sampling density $\tilde{q}(\tau, \theta) \coloneqq q(\tau, \theta)p^{\text{targ}}(\tau)/p(\tau)$;

  $j \coloneqq j + 1$;

until *The ratios* $r = \{p_j(\tau)/p_{j-1}(\tau)\}$ *have* $\hat{k} < 0.7$.;

The optimal marginal distribution of a. Lastly, the adaptive path sample estimate does not depend on the marginal distribution of a and λ, so that this marginal distribution is determined by user specification. By default in (5.9) we use the update rule $z(\cdot) \to c(\cdot)$, which enforces a uniform marginal distribution on a. More generally, by updating $c(f(\cdot)) \leftarrow z(f(\cdot))/p^{\text{prior}}(\cdot)$, the final marginal distribution of a will approach p^{prior}. The choice of p^{prior} is subject to a *efficiency-robustness* trade-off. Gelman and Meng (1998) showed that the generalized Jeffreys prior $p^{\text{opt}}(\lambda) \propto \sqrt{\mathbb{E}_{\theta|\lambda}\, U^2(\theta, \lambda)}$ minimizes the variance of the estimated log normalizing constant, where $U(\theta, \lambda) = \frac{\partial}{\partial \lambda} \log q(\theta, \lambda)$. With a slight twist, in continuous tempering (Section 6.1), we can prove that another optimal prior $p^{\text{opt}}(a) \propto \frac{1}{f'(a)} \sqrt{\text{Var}_{\theta \sim p(\theta|a)} U(\theta, a)}$ ensures a smooth KL gap

between two adjacent tempered distribution in posterior sample $\mathrm{KL}(\pi_a, \pi_{a+\delta a}) \approx$ constant for $a_{\min} < a < a_{\max}$ (Appendix 6.2). In discrete tempering, this constant KL gap is related to a constant acceptance rate in the neighboring Gibbs update. However, due to its dependence on the unknown normalizing constant (and higher orders), these two efficiency-optimal priors require additional tuning and adaptations. In continuous tempering, we prefer the simple uniform a margin for robustness, as it guarantees a complete λ tour in the joint path. In implicit divide and conquer, if the efficiency is more of a concern, we recommend to adaptively update the target marginal to match the efficiency-optimal one $p^{\text{targ}}(\tau) \leftarrow p^{\text{opt}}(\tau) \propto \sqrt{\mathbb{E}_{\theta\sim q(\theta,\tau)}(\frac{\partial}{\partial\tau}\log q(\theta,\tau))^2}$, which can be further stochastically approximated by joint Monte Carlo draws. This additional adaptation is optional and the estimation of p^{opt} is not required to be precise.

6.4 Related literature

Simulated tempering. Simulated tempering and its variants provide an accessible approach to sampling from a multimodal distribution. We augment the state space Θ with an auxiliary inverse temperature parameter λ, and employ a sequence of interpolating densities, typically through a power transformation $p_j \propto p(\theta|y)^{\lambda_j}$ on the ladder $0 < \lambda_1 < \ldots \lambda_K = 1$, such that p_K is the distribution we want to sample from and p_0 is a (proper) base distribution. At a smaller λ, the between-mode energy barriers in $p(\theta|\lambda)$ collapse and the Markov chains are easier to mix. This dynamic makes the sampler more likely to fully explore the target distribution at $\lambda = 1$.

Discrete simulated tempering (Marinari and Parisi, 1992; Neal, 1993; Geyer and Thompson, 1995) samples from the joint distribution $p(\lambda, \theta) \propto 1/c(\lambda)q(\theta)^\lambda$ using a Gibbs scheme, where $c(\lambda)$ is a pseudo-prior that is often iteratively assigned to be $\hat{z}(\lambda)$: an estimate of the normalizing constant of $q(\theta)^\lambda$ which may be obtained by any of the methods discussed above. Each Gibbs swap involves sampling $\theta|\lambda$ with a one- or multi-step Metropolis update that keeps $p(\theta|\lambda)$ invariant, and a random walk in λ that leaves $\lambda|\theta$ invariant. The number of Metropolis updates, the number of temperature samples, and the temperature spacing all involve intensive user tuning.

In simulated annealing (Neal, 1993; Morris et al., 1998), we evolve λ through a fixed schedule,

and update θ using a Markov chain targeting the current conditional distribution $p(\theta|\lambda)$. The annealed importance sampling (Neal, 2001) adjusts the non-equilibrium in the $\theta|\lambda$ update by the importance weight defined in (5.14).

However, as we mentioned above, discrete tempering schemes are sensitive to the choice of λ ladders. The Markov chain must usually proceed by making small changes between the neighboring distributions. Following the discussion in Betancourt (2015), if some pair of π_{λ_j} and $\pi_{\lambda_{j-1}}$ have a large KL divergence, the error in importance sampling (5.12) based algorithms will be dominated by the j-th term. Under the optimal design, the Kullback–Leibler (KL) divergence between neighboring distributions should be roughly constant:

$$\text{constant} \;\approxeq\; \mathrm{KL}\Big(\pi_\lambda, \pi_{\lambda+\delta\lambda}\Big) = \int \log\big(\frac{\pi_\lambda}{\pi_{\lambda+\delta\lambda}}\big) d\pi_\lambda = \log\big(\frac{z(\lambda+\delta\lambda)}{z(\lambda)}\big) - (\delta\lambda)^2 \frac{d}{d\lambda} \log z(\lambda), \qquad (6.2)$$

which can hardly be achieved even adaptively due to the reliance on the unknown $\log z$ and its derivative.

Further, even with a constant KL gap, the discrete ladder imposes dimension limitations. For example, in simulated tempering, a transition $(\theta, \lambda_i) \to (\theta, \lambda_j)$ between two temperatures λ_i and λ_j in the Gibbs update would only be likely if there is significant overlap between the potential energy distributions $\log q(\theta|\lambda_i)$ and $\log q(\theta|\lambda_j)$. Under a normal approximation with $\dim(\Theta) = d$, the width of the energy distribution scales by $O(d^{1/2})$, while the distance between two adjoining energy distributions is $O(d/K)$. This leads to the best case bound on the necessary number of interpolating densities $K = O(d^{1/2})$. In practice, K is recommended to grow like $O(d)$ (Madras and Zheng, 2003). More theoretical discussion shows that $K = O(d)$ is often needed to ensure a polynomial bound on the adjacent temperature overlap and rapid mixing (Woodard et al., 2009). Since the update on $(\lambda_k)_{k=1}^K$ behaves as a random walk, even when the $\theta|\lambda$ update takes zero time, the relaxation time of a diffusion process with discrete state space $(\lambda_k)_{k=1}^K$ often scales by $O(K^2)$, soon becoming unaffordable as K grows.

Toward continuous tempering. Gobbo and Leimkuhler (2015) designed a continuous tempering

scheme by adding a single auxiliary variable $a \in R$ to the system. The inverse temperature $f(a)$ is defined by a smooth function such that $f(a) = 1$ for $|a| < \Delta$ and $f(a) = 0.15$ for $|a| > \Delta^*$. The Hamiltonian of the system is modified to be $\hat{H}(\theta, p, a, p_a) = f(a)H(\theta, p) + p_a^2/m_a + \log z(a)$, where p_a is the momentum of a, and $\log z(a)$ is adaptively updated using importance sampling, in order to force a to be uniformly distributed in the interval $[\Delta^*, \Delta^*]$. Similarly, Betancourt (2015) introduced adiabatic Monte Carlo, where a contact Hamiltonian is defined on the augmented Hamilton system $\hat{H}(\theta, p, \lambda) = -\lambda \log q(\theta) + 1/2 p^T M^{-1} p + \log z(\lambda)$. These methods are shown to outperform discrete tempering, but they require a modified Hamiltonian and are sensitive to task-specific implementation and tuning.

Graham and Storkey (2017) formulated a continuous tempering on the joint density $p(\theta, \lambda) \propto \zeta^{-\lambda} q(x)^{\lambda} \psi(x)^{(1-\lambda)}$. This can be viewed as a special case of our proposed method by restricting the parametric form of our normalizing constant estimate to a single parameter exponential function $z(\lambda) = \zeta^{\lambda}$. This method does not directly acquire simulation draws from $\theta|\lambda = 1$ or 0, and integrals under target distribution are evaluated through importance weighting.

6.5 Experiments

To manifest the advantage of the proposed method, we present a series of experiments. In Section 6.5, we use a conjugate model to compare the accuracy of normalizing constant estimation. In Section 6.5, we show that the continuous tempering with path sampling is scalable to high dimensional multimodal posteriors. In Section 6.5 and 6.5, we highlight the computational efficiency from the proposed method, including higher effective sample size and quicker mixing time. Lastly, we validate the implicit divide and conquer in a funnel shaped posterior in Section 6.5.

Beta-binomial example: comparing accuracy of estimates of the normalizing constant We start with an example adapted from Betancourt (2015), in which the true normalizing constant can be evaluated analytically. Consider a model with a binomial likelihood $y \sim \text{Binomial}(\theta, n)$ and a Beta prior $\theta \sim \text{Beta}(\alpha, \beta)$. Along a geometric bridge between the prior and posterior, the

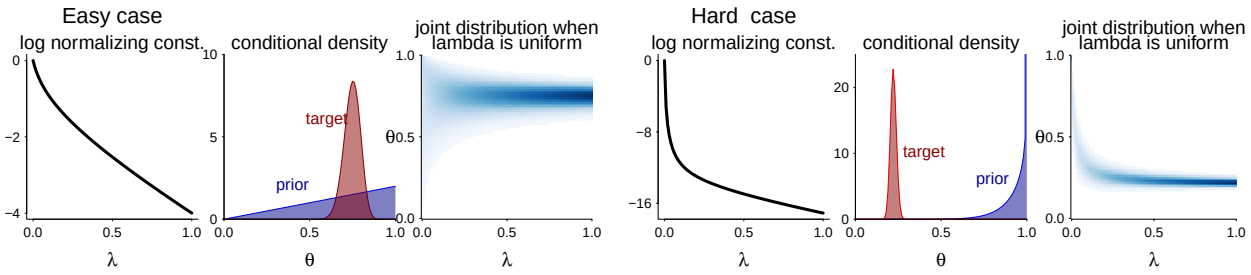


Figure 6.2: *(a) The analytic log normalizing constant. (b) the base and the target (c) the joint of* (λ, θ) *when the marginal of* λ *is uniform. In the hard case the target and base is separated and the log normalizing constant changes rapidly.*

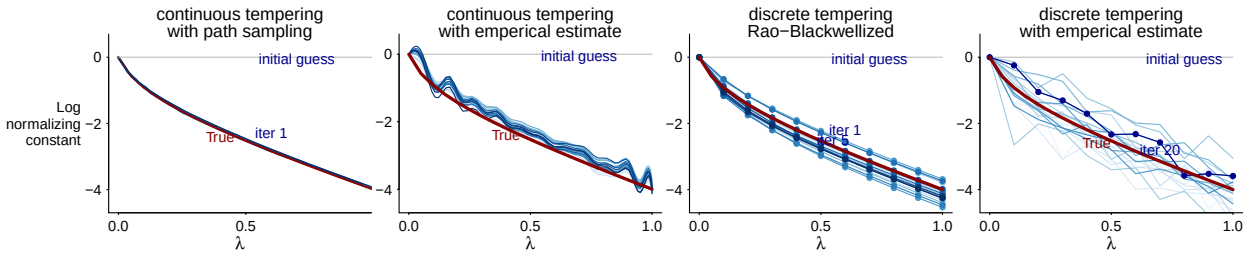


Figure 6.3: *Estimation of the log normalizing constant in the easy Beta-binomial example among first 20 adaptations. All methods eventually approximate the true function, while the proposed continuous tempering with path sampling has the fastest convergences rate.*

conditional unnormalized tempered posterior density at an inverse temperature λ is

$$q(\theta|\lambda) = \text{Binomial}(y|\theta, n)^{\lambda}\text{Beta}(\theta|\alpha, \beta), \quad \theta \in [0,1],\ \lambda \in [0,1].$$

Due to conjugacy, the normalized distribution has a closed form expression $p(\theta|\lambda) = \text{Beta}(\lambda y + \alpha, \lambda(n-y) + \beta)$, and the true normalizing constant z is

$$z(\lambda) = \int_0^1 q(\theta|\lambda)d\theta = \left(\frac{\Gamma(n+1)}{\Gamma(y+1)\Gamma(n-y+1)}\right)^{\lambda} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\Gamma(\lambda y+\alpha)\Gamma(\lambda(n-y)+\beta)}{\Gamma(\lambda n+\alpha+\beta)}.$$

We compare four sample-based estimates of the log normalizing constant: (i) continuous tempering with adaptive path sampling (the proposed method) with joint density $p(a, \theta) \propto 1/c(f(a))q(\theta|f(a))$; (ii) continuous tempering, but replacing the path sampling estimate of marginal density $p(a)$ by an empirical estimate and using importance sampling $\hat{z}(a) = p(a)c(a)$ to compute and update z; (iii) discrete simulated tempering with importance sampling estimation and estimat-

ing the marginal probability mass function $\Pr(\lambda = \lambda_i), 0 = \lambda_0 < \lambda_1 < \ldots < \lambda_K = 1$ by Monte Carlo average; (iv) discrete tempering with a Rao-Blackwellized (Carlson et al., 2016) estimate of the probability mass function.

In the first setting (left half of Figure 6.2), we set $\alpha = 2$, $\beta = 1$, $y = 60$, $n = 80$. Figure 6.3 presents the log normalizing constant estimates in the first 20 adaptations. All methods start with a flat initial guess $\log z(\lambda) = 0$. In continuous tempering, we draw 3000 joint (a, θ) draws in each adaptation. The first half of the draws are treated as warm-up and discarded (which also do in the discrete setting). We choose a length $I = 100$ approximation grid in the parametric adaptation step (5.6). For discrete tempering, we use an evenly spaced ladder $\lambda_i = (i-1)/10, i = 1, \ldots, 11$ and draw 150 λ draws per adaptation, each followed 100 HMC updates in θ. These numbers ensure the continuous tempering has a smaller computation cost (3000 joint draws) than in discrete implementations (100 HMC updates on $\theta \times 150$ updates on λ_i) per adaptation, so as to make the efficiency comparison convincing.

In this easy case, all methods eventually approximate the truth(red curve), among which our proposed continuous tempering with path sampling has the quickest convergence rate, almost recovering the truth within the first adaptation.

In the hard case (right half of Figure 6.2), $\alpha = 9$, $\beta = 0.75$, $k = 115$, $n = 550$. The base and target are two isolated spikes, and the log normalizing constant changes rapidly especially for small λ . Figure 6.4 compares four log normalizing constant estimation methods in the first 20 adaptations. Continuous tempering with path sampling nearly converges to the true value after the 8th adaptation. Rao-Blackwellization achieves a reasonable discrete approximation at the 13-th adaptation, but the result is unstable due to discretization error. In fact, Figure 6.5 displays the L_2 error of $\log z$. Only the proposed method has a monotonically decreasing error with more adaptations. The two empirical importance sampling based methods are both too noisy to be useful.

Rows 2–4 in Figure 6.4 show the first 150 joint draws in adaptation 3, 6, and 10 from all four methods. Our proposed method fully explores the joint space in adaptation 10. Here, the joint HMC

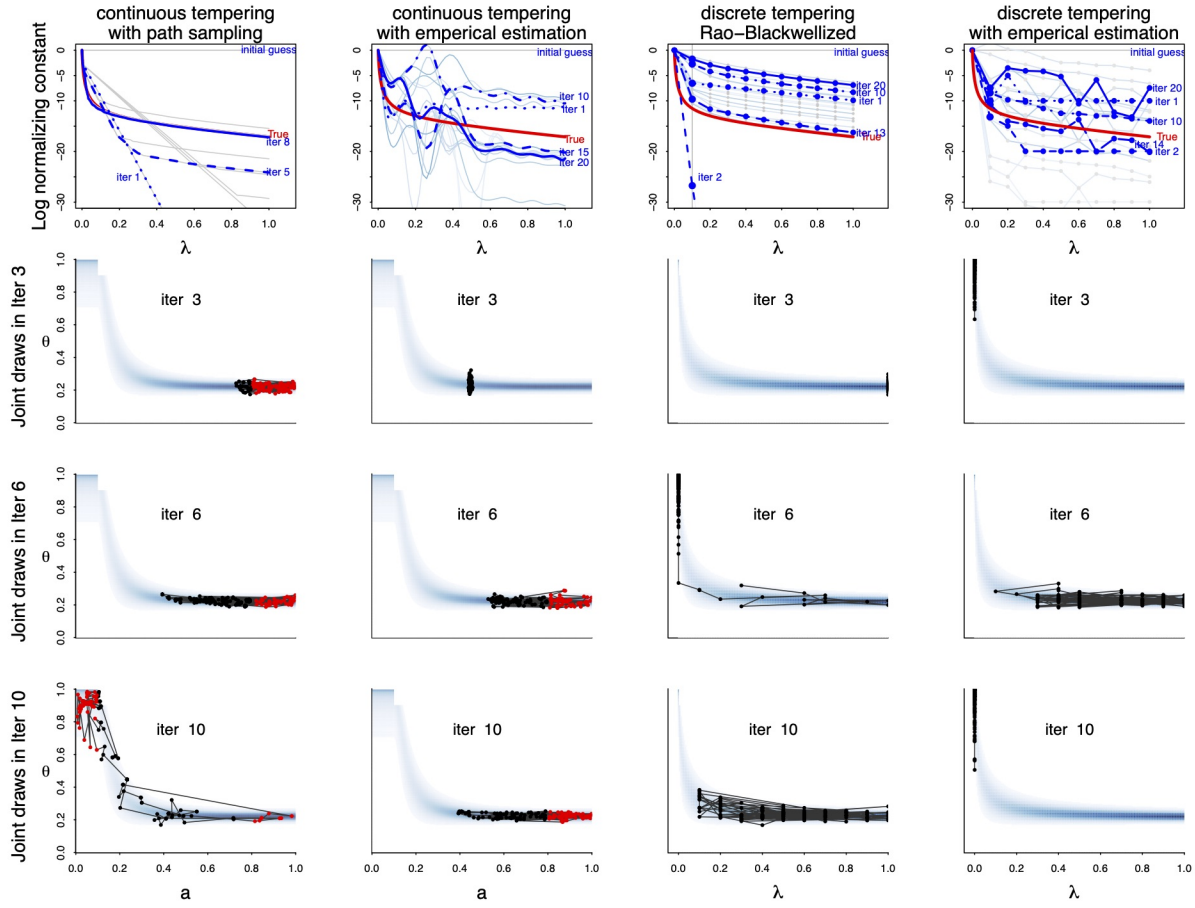


Figure 6.4: *Comparison of four tempering methods in the hard beta-binomial example. Row 1: Starting from a flat guess, only the proposed method converges to the the true (red curve) value of the log normalizing constant after 8 adaptations. Rows 2–4 compare the first 150 draws of the joint distribution of parameters and temperatures in adaptations 3, 6, and 10. The proposed method fully explores the the joint space efficiently, while the two discrete schemes exhibit random walk behaviour in temperatures updates, and cannot adapt to the rapid changing regions near* $\lambda = 0$.

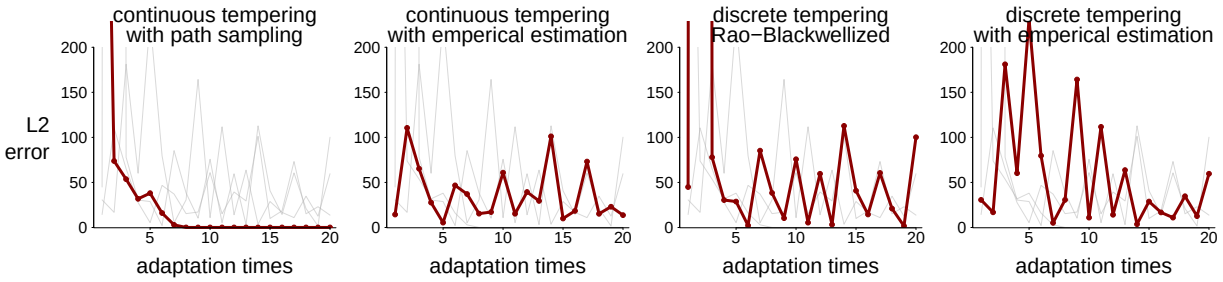


Figure 6.5: *L_2 errors of the log normalizing constant estimate* $\log z(\lambda)$ *from four methods during the first 20 adaptations. Only continuous tempering with path sampling gives a monotonically decreasing error which shrinks to zero in practical amount of time.*

jumps are gradient-guided and and make subtle jumps in regions of rapid change (where $\lambda \approx 0$). In contrast, since the conditionals at $\theta|\lambda = 0$ and $\lambda = 0.1$ differ significantly, the discrete tempering procedure cannot automatically impute more ladders among them, and always over-weighs one end.

Sampling from Gaussian mixtures: compare accuracy of the multimodal sampling Next we consider the problem of sampling from Gaussian mixtures. For visual illustration we begin with the simple case of a mixture of two one-dimensional Gaussians.

Figure 6.6 shows five out of ten total iterations for which we ran our tempering algorithm. At initialization, the joint sampler can only see a thin slice of a in the region around the base, with a large resulting Pareto-$\hat{k}$ diagnostic. With more adaptations, more temperatures are sampled, and the path sampling estimate relies less on extrapolation. After four adaptations, the sampler has fully explored $a \in [0, 2]$, and completed a base-target bridge in the augmented space. Collecting simulation draws with $f(a) = 1$ retrieves the target distribution. The accuracy is confirmed by the $\hat{k} < 0.7$, or a visual check of a nearly uniform a marginal distribution.

Next we consider a 10-component mixture of Gaussians. We generate the individual Gaussian components with variance a tenth of the minimum distance between any two of the mode centers to ensures separation between the modes. We perform this sampling in a range of dimensions from 10 to 100 (with the number of components fixed). We compare the proposed tempering with path sampling with two benchmark algorithms (i) Rao-Blackwellized discrete tempering (Carlson et al., 2016), and (ii) continuous tempering with a log linear normalizing constant assumption (Graham and Storkey, 2017). We do not include other empirical importance sampling based tempering as they are strictly worse than these two benchmarks. For all methods, we use an over-dispersed normal base distribution.

In order to assess whether a given tempering procedure has succeeded, we count the proportion of draws in each target mixture component for each tempering method, and compare this distribution to the actual equal weight target. The results are averaged over five independent runs. Figure 6.7

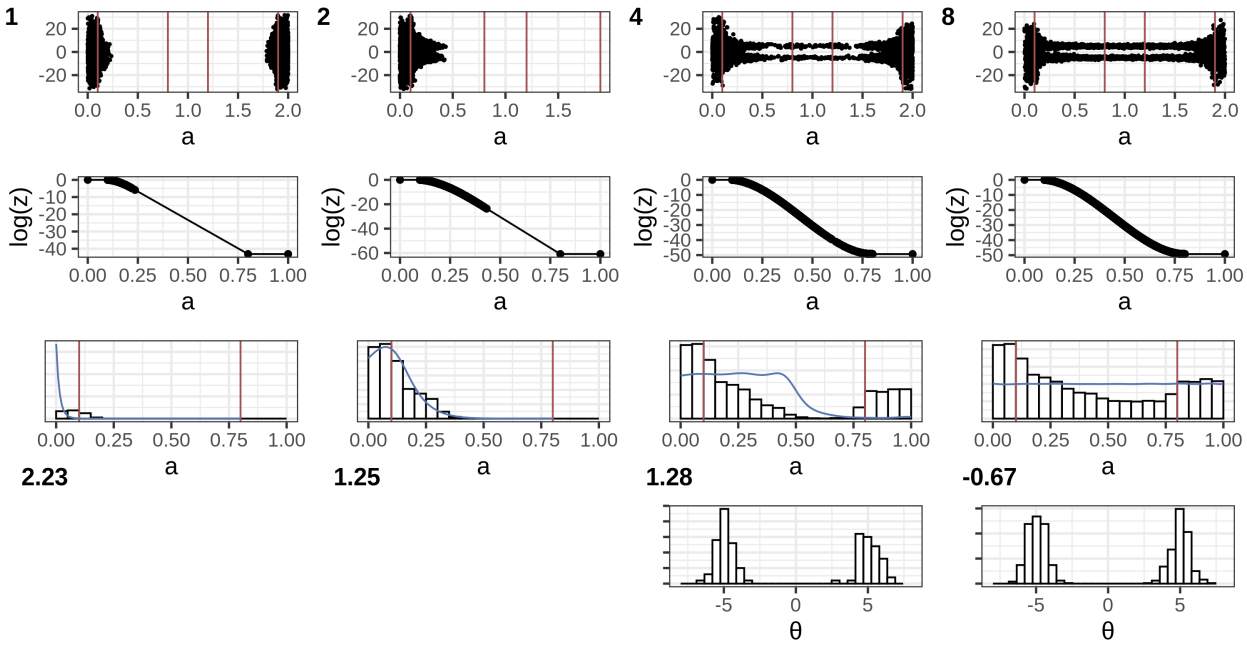


Figure 6.6: *Path sampling-based tempering for the target a Gaussian mixture, plotting adaptation 1, 2, 4, 6, and 10. The first row shows the joint simulation draws; the second row displays the estimate of the log normalizing constant; the third row displays the marginal density on the temperature and cumulative draws of the temperature variable, with Pareto-$\hat{k}$ diagnostics printed at the bottom left of each panel (good if $\hat{k} < 0.7$); and the fourth row is the draws from the target density.*

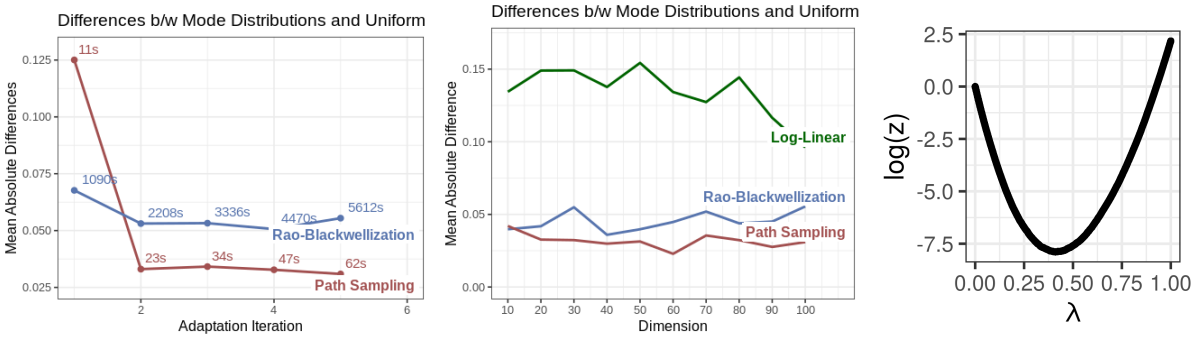


Figure 6.7: *Mean absolute error of target simulation draws from adaptive path sampling based tempering (proposed), and two benchmarks: log-linear continuous tempering and Rao-Blackwellization. Results are averages over 5 repeated runs. Left: Sampling error and time at each adaptation for a Gaussian mixture target with 100 dimensions and 10 components. Numerical labels indicate the number of seconds of CPU time before each adaptation completed. Middle: Comparison of sampling errors as dimensions range from $d = 10$ to 100. The log-linear tempering is supplied with true normalization constant while the other two are with uniform initialization. Right: The estimated log normalizing constant $\log z(\lambda)$ from path sampling.*

displays the evolution of the mean absolute difference between the sampled mode proportions in each adaptation of each algorithm and the target in the 100-dimensional case. The left panel of Figure 6.7 labels each point with the total CPU time needed to complete each adaptation iteration. Not only does the path sampling algorithm achieve more uniform mode exploration, but it does so in significantly less computation time than the Rao-Blackwellized procedure. This is partly a symptom of the HMC-in-Gibbs implementation of the Rao-Blackwellized algorithm.

The middle panel of Figure 6.7 displays the same mean absolute difference between sampled and actual target distribution against the dimension for continuous tempering with path sampling, the Rao-Blackwellized procedure, both with a uniform initialization, and the log linear tempering initialized at the true normalizing constant. For path sampling and Rao-Blackwellized sampling, we plot the results after five iterations of adaptation. In all cases, the final in-target sample from the proposed method is closest to the actual target, with the the smallest mean absolute differences. As shown in the right panel, the normalization constant $\log z(\lambda)$ is not monotone or linear, which explains the undesired performance of the log linear tempering even when it is supplied with the ground truth normalizing constant.

In Figure 6.7, we do not see the mode exploration degrade with increasing dimension for path sampling. This is because the log normalizing constant itself scales mildly with the dimension in Gaussian mixtures. It general, the number of dimensions is not the limiting factor of path sampling, but it could inflate the log normalizing constant in a severe base-target mismatch. We further discuss dimension scalability in Appendix C.

Flower target: the gain on computation efficiency Next we consider sampling from a flower shaped distribution as previously used in Sejdinovic et al. (2014) and Nemeth et al. (2019). This is a two-dimensional distribution with probability density spread throughout multiple "petals". Its probability density function is given by

$$p(x, y \mid \sigma, r, A, \omega) \propto \exp\left(-\frac{1}{2\sigma^2}\left(\sqrt{x^2 + y^2} - r - A\cos(\omega \arctan(y, x))\right)\right).$$

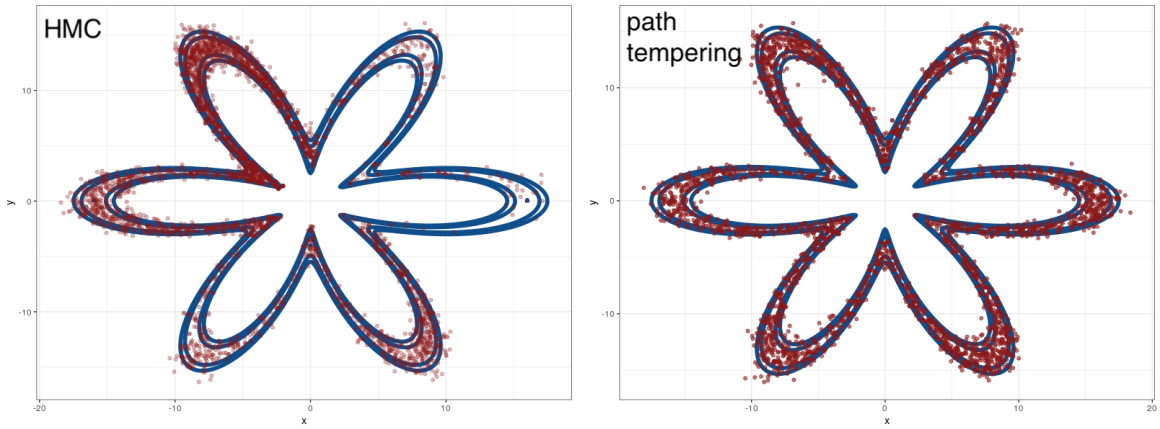


Figure 6.8: *Draws (red dots) from the flower target generated by plain HMC (left) and continuous tempering with path sampling (right). The blue contours represent the underlying target density.*

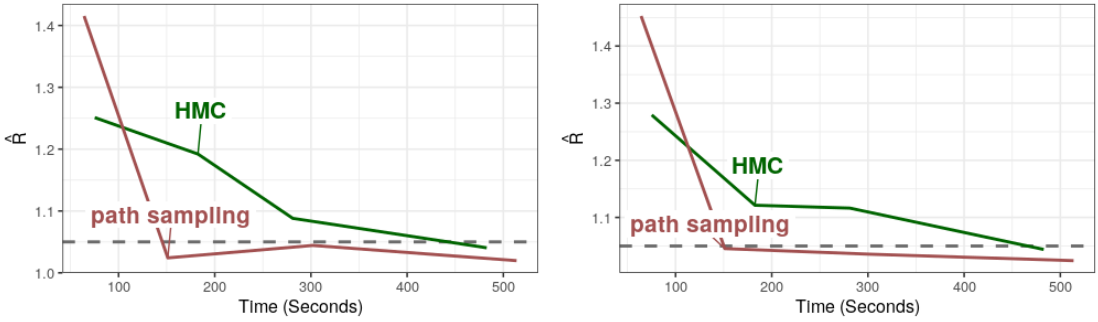


Figure 6.9: *Comparison of computational cost of path sampling versus standard HMC for a flower distribution with 40 petals. The x-axis displays the time in seconds, and the y-axis shows the achieved $\widehat{R}$-value for the x coordinate (left) and y coordinate (right) of the flower distribution. The dashed grey line displays the 1.05 cutoff below which we consider our chains to have mixed sufficiently. All results are averaged over 15 replications.*

The probability mass is pinched into narrow regions between petals, slowing exploration of the target. For this reason, the flower distribution is challenging to sample from using standard HMC.

Figure 6.8[2] plots draws from the flower distribution with 6 petals using plain HMC and using continuous tempering with path sampling. As before, we use a normal base distribution. Both algorithms were run for enough time to generate a similar number of draws from the target. Standard HMC clearly fails to adequately explore each of the petals of the flower distribution. Path sampling-based continuous tempering succeeds in generating draws from each of the petals in roughly equal proportions.

[2]Figure 6.8, 6.9, 6.11, 6.12, and the underlying experiments were conducted by Collin Cademartori.

We further compare the total mixing time. Figure 6.9 plots the computed $\hat{R}$ (Vehtari et al., 2020a) for the x and y coordinates of a challenging flower distribution with 40 petals against computational time. Continuous tempering with path sampling crosses the $\hat{R} = 1.05$ threshold for adequate mixing in about a quarter of the time (already having included all adaptations) required for standard HMC to reach that threshold, although the latter can eventually approximate the target with many more draws. Thus, despite the much lower time cost per sample for standard HMC, path sampling still manages to be more efficient in terms of total mixing time. This disparity will only get more apparent as the difficulty of the target density increases.

Regularized horseshoe regression: expand models continuously and efficiently The proposed path sampling and continuous tempering can enhance computational efficiency even when there is no direct posterior multimodality. In particular, the HMC mixing time scales polynomially with dimensions, hence fitting a slightly larger model can be cheaper than fitting two models separately. Consider a sparse regression with regularized horseshoe prior (Piironen and Vehtari, 2017b,c), an effective tool for Bayesian sparse regression. Letting $y_{1:n}$ be a binary outcome and $x_{n\times D}$ be predictors, the regression coefficient β is equipped with a regularized horseshoe prior: $\beta_d \mid \tau, \zeta, \gamma \sim \text{normal}\left(0, \gamma\zeta_d(\gamma^2 + \tau^2\zeta_d^2)^{-1/2}\right)$, $\tau \sim \text{Cauchy}^+\left(0, 2/\left(D-1\right)\sqrt{n}\right)$, $\zeta_d \sim \text{Cauchy}^+(0,1)$, $d = 1, \ldots, D$. To take into account the model freedom between the logit link $\Pr(y_i = 1 \mid \beta, \text{logit}) = \text{logit}^{-1}\left(\sum_{d=1}^{D} \beta_d x_{id}\right)$ and probit link $\Pr(y_i = 1 \mid \beta, \text{probit}) = \Phi\left(\sum_{d=1}^{D} \beta_d x_{id}\right)$ in the likelihood, we construct a tempered path between the logit and probit model

$$p(a, \beta, \tau, \zeta, \gamma) \propto \frac{1}{c(a)} \prod_{i=1}^{n} \left(\Pr(y = y_i|\beta, \text{logit})^{1-\lambda} \Pr(y = y_i|\beta, \text{probit})^{\lambda}\right) p^{\text{prior}}(\beta, \tau, \zeta, \gamma),\ \lambda = f(a),$$

where $p^{\text{prior}}(\beta, \tau, \zeta, \gamma)$ encodes the regularized horseshoe prior.

In the first experiment, we generate $n = 40$ data points and $D = 100$ covariates with a maximum pairwise correlation 0.5. Among all covariates, only the first three have nonzero coefficients $\beta_{1,2,3}$. Furthermore, y is chosen to have a logit link in the true data generating process. We vary $a_{\min}$ in the link function $\lambda = f(a)$ as defined in Equation (6.1), such that when $0 < a < a_{\min}, \lambda = 0$, the

path sampling draws from the logit model, and when $1 > a > a_{\max} = 1 - a_{\min}, \lambda = 1$, it draws from the probit model. We run path sampling with two adaptations, $S = 500$ draws in the first adaptation, and $S = 3000$ in the second. We then compute the tail effective sample size (tail-ESS, Vehtari et al., 2020a) from the the probit and logit model in the aggregated draws divided by the total sampling time (sum of 4 chains). For compassion, we also fit these two models separately and count their effective sample size. To reflect the bottleneck of the computation, we monitor the smallest tail effective sample size among all regression coefficients β_d. Path sampling between two models expands the model continuously such that both individual models are special cases of the augmented model. Because the posterior distributions $\beta|y$ under the probit and logit links are similar, a connected path between them stabilizes the tail of posterior sampling, resulting in a larger unit time tail effective sample size, as shown in the left two panels of Figure 6.10.

In the second experiment, we fix $a_{\min} = 0.35, a_{\max} = 0.65$, and vary the input dimension $D = n\rho, 2 \leq \rho \leq 8$. Given that $a_{\max} - a_{\min} = 30\%$ of the sampling time is spent on intermediate models, it is remarkable that most times the joint sample from path sampling renders a tail effective sample size from individual models no slower than fitting them separately, as verified in the right two panels in Figure 6.10. The simulation results are averaged over 10 repeated runs. As a caveat, a joint path of two arbitrary models is not always more efficient than separate fits. Figure 6.10 displays the minimal effective sample size among all regression coefficients β_d. When averaged over all β_d, the mean tail effective sample size in path sampling is smaller than individual fits in this example, which can be viewed as a parameter-wise efficiency-robustness trade-off.

Sampling from funnel shaped posteriors by implicit divide-and-conquer We apply the implicit divide-and-conquer algorithm to the hierarchical model and eight-school dataset (Gelman et al., 2013). In this problem, we are given eight observed means y_i and standard deviations s_i which are related to each other through the following hierarchical model with unknown parameters θ, μ, τ,

$$\text{centered parameterization}: \qquad y_i \sim \text{normal}(\theta_i, s_i), \quad \theta_i \overset{iid}{\sim} \text{normal}(\mu, \tau).$$

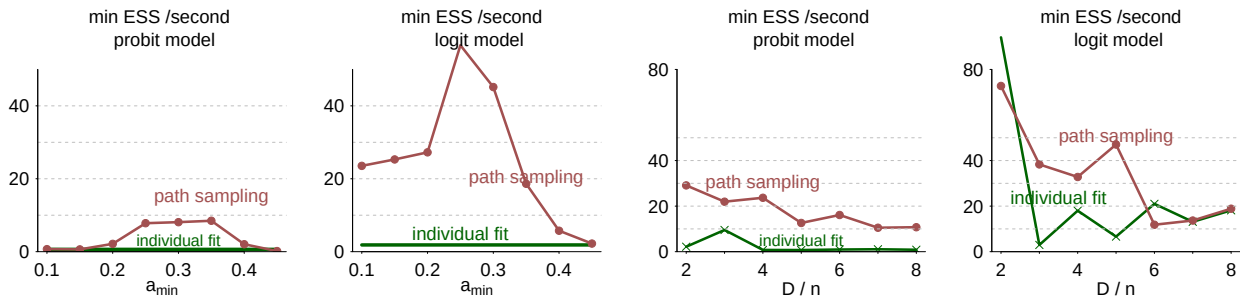


Figure 6.10: *The smallest unit time tail effective sample size among all regression coefficients in the horseshoe regression with both Bernoulli logit and probit likelihood. The green line refers to two separate model fits and the red line corresponds to a joint tempered path sampling counting all adaptation time. Left two panels: we vary* $a_{\min}$*: the proportion of logit and probit sampling in the final joint sample. Right two panels: we fix* $a_{\min} = 0.35, a_{\max} = 0.65$ *and vary the input-dimension/sample-size from 2 to 8. In most cases, the joint path sampler generates a larger tail ESS /s for both probit and logit sample, in addition to all intermediate models fitted simultaneously.*

The centered parametrization in hierarchical models often behaves as implicit left-truncation due to entropic barriers. In particular, as $\tau \to 0$, the mass of the conditional posterior $p(\theta \mid \tau, y, s)$ concentrates around μ, creating a funnel-shape in the posterior of θ that is challenging to traverse. In this dataset, the left truncation can be overcome by switching to a non-centered parametrization that introduces auxiliary variables $\tilde{\theta}$,

$$\text{non}-\text{centered parameterization}: \qquad \theta_i = \mu + \tau \times \tilde{\theta}_i, \quad \tilde{\theta} \overset{iid}{\sim} \text{normal}(0, 1).$$

However, reparametrization is often restrited to location scale families, and choosing between centered and non-centered parametrization remains difficult for real data (Gorinova et al., 2020).

We apply the proposed implicit divide-and-conquer algorithm on the sample drawn from the original centered parametrization. It successively pushes the marginal of τ towards a desired target marginal $p^{\text{targ}}(\tau)$. In our case, we use the efficiency-optimal prior of the form $p(\tau) \propto \sqrt{\mathbb{E}_\tau U^2(\theta, \mu, \tau)}$, where the $U(\theta, \mu, \tau)$ functions are the gradients of the log posterior given in (5.8). This target cannot be computed explicitly, so we also estimate it at each adaptation using a simple window-based averaging scheme over the τ draws. We adaptively repeat the algorithm until we meet the convergence criterion.

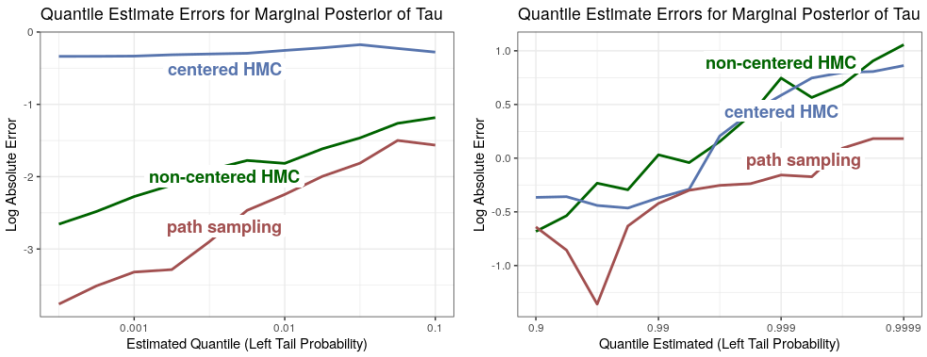


Figure 6.11: *Comparison of log estimation error of the left and right tail quantile estimations using $S = 4000$ posterior draws from HMC (centered and non-centered parametrizations) and implicit divide-and-conquer. The x-axis displays the left tail probability. The y-axis displays the log absolute error of the quantile estimate. The proposed method (red curve) achieves the highest accuracy.*

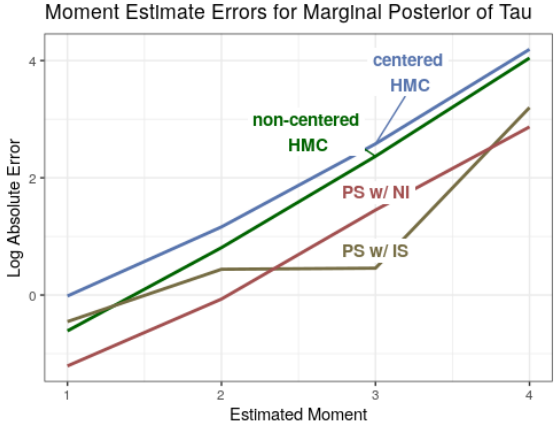


Figure 6.12: *Comparison of log estimation errors of the first to fourth posterior moment of τ using draws $S = 4000$ from HMC (centered and non-centered parametrizations) or implicit divide-and-conquer. The latter returns the marginal density estimation, which is further equipped with either importance sampling or numerical integration. The x-axis displays the estimated moment, and the y-axis displays the log absolute error of the moment estimate.*

Figure 6.11 displays the log errors for quantile estimates in the right and left tails of the marginal posterior of τ for HMC with (a) centered and (b) non-centered parametrizations, and (c) the implicit divide-and-conquer. We set posterior draw size $S = 4000$ (after thinning) in all three samples. In these experiments, we use estimates from the non-centered parametrization with 10^6 draws as our ground truth. In the right tail, the implicit divide-and-conquer runs out-perform the Monte Carlo estimates across all but the smallest quantile. For the left tail, we estimate quantiles with left tail probabilities between 0.001 and 0.1. In this case, path sampling dominates all other methods, with a significant improvement in the extreme quantiles.

The estimated marginal density $p(\tau)$ enables expectation computation. We examine the performance for marginal moment estimation $\mathbb{E}(\tau^m)$ using both importance sampling and numerical integration (the same trapezoidal rule as in (5.9)). Figure 6.12 displays the log errors of estimates for the first four moments of the marginal posterior distribution of τ. The moments computed from simulation draws of the implicit divide-and-conquer, using either numerical integration scheme or importance sampling, have a lower error than Monte Carlo estimates from both HMC parametrizations.

Overall, we see from Figures 6.11 and 6.12 that the proposed implicit divide-and-conquer provides good performance for a range of posterior estimation tasks, often out-performing all other approaches and never generating estimates with errors large enough to be unusable in any of the tested problems. This is remarkable since the implicit divide-and-conquer scheme generates its underlying HMC samples using the centered parametrization during each adaptation.

The implicit divide-and-conquer scheme demands more computation time than one-run HMC, but it avoids the need for a problem-specific reparametrization, so it can be applied in cases where other approaches may not be available. Furthermore, because it comes with a stopping criterion, we can ensure that the extra computation time arrives at a sufficiently accurate result by termination.

6.6 Discussion

Initialization and base measurement In continuous simulated tempering, we initialize the pseudo prior at $c = 1$. When the underling normalizing constant $z(\lambda)$ is several orders of magnitude smaller than $z(0)$ for all $\lambda > 0$, the sampler starting from this non-informative initialization will get stuck in $\lambda = 0$. Because $f'(a) = 0$ in the base, path sampling will fail to update. In this situation, we update the slope b_0 in (5.6) to be the importance sampling estimate (5.4): $b_0 \leftarrow \log \hat{z}^{\text{IS}}(1) = \log\left(\frac{1}{S}\sum_{s=1}^{S} q(\theta_s, \lambda = 1)q^{-1}(\theta_s, \lambda = 0)\right)$. This estimate $\hat{z}^{\text{IS}}$ is unlikely to be accurate, thus we only use for initialization to avoid local convergence.

In Section 6.1, we merely require the base measure ψ to have the same support as the target q. A natural candidate for ψ is the prior as we have used throughout the experiments. Ideally, the base measurement should balance between being easy to sample from, and close enough to the target distribution. This is not a unique challenge in our method, as all (discrete) simulated tempering and, more generally, importance sampling methods need to construct a good base measurement. The current chapter treats the base measurement as an extra input that user has to specify. Based on the discussion on how to construct and optimize the proposal distribution in adaptive importance sampling (Geweke, 1989; Owen and Zhou, 2000; Bugallo et al., 2017; Paananen et al., 2021), it may be possible to obtain better performance by optimizing the choice of base measurement, which itself is often an iterative problem.

That said, we are unlikely to be able to automatically construct a good base density for importance sampling in high dimensional problems such that the KL divergence between the base and the target is bounded. Otherwise other advanced sampling method would not be required. Fortunately, the adaptive path sampling estimate is less sensitive to base-target discrepancy. As we have seen in experiments, our path sampling-based method still yields accurate log normalizing constant estimates and smooth joint sampling even starting from a crudely chosen base distribution when importance sampling and bridge sampling have failed.

Dimension limitations In accordance with the error analysis in Section 5.3 and the simulation results, the proposed adaptive path sampling is more scalable to higher dimensions compared with existing counterparts. However, path sampling-based continuous tempering can still fail in high dimensional posteriors.

In essence, simulated tempering depends on, but can be more difficult than normalizing constant estimation. On one hand, simulated tempering does not require a precise estimation of $z(\lambda)$ (left half of Figure 6.13, see also Geyer and Thompson, 1995), as long as there is enough posterior marginal density $p(a)$ or $p(\lambda)$ everywhere—but this will only happen when the scale of $z(\lambda)$ is small.

In Appendix C, we provide a failure mode example in a latent Dirichlet allocation model, where the log normalizing constant estimation has a scale $\sim 10^4$, and path sampling estimation manages to estimate it with pointwise errors $\sim 10^2$. For fitting a curve, a 1% error is accurate enough. But a pointwise 1% error in the log normalizing constant amounts to inflating the marginal density $p(a)$ in the joint sampling (5.5) by a factor $\exp(10^2)$ at that point, which effectively becomes a point mass and makes path tempering get stuck in one region. Such failures happen in discrete tempering too, and is identifiable by our $\hat{k}$ diagnostics. In other words, successful simulated tempering requires the estimation of the pointwise normalizing constant with multiplicative precision, see Figure 6.13 for an illustration.

The absolute scale of the log normalizing constant is comparable to the log KL divergence between the base and the target. In a prior-posterior tempering path, the log normalizing constant at $\lambda = 1$ is the log marginal likelihood. It grows linearly with both the sample size and how closely the model fits the data (the log likelihood). When the model is poor in predication (as in the latent Dirichlet allocation example), the log normalizing constant soon escapes from the estimation accuracy that any estimate can achieve, an analogy of the "folk theorem of statistical computing": When you have computational problems, often there's a problem with your model (Gelman, 2008).

Furthermore, we present a geometric path for continuous tempering. It is not clear if to modify the free energy by a density-power-transformation is the best way to remove metastability, although

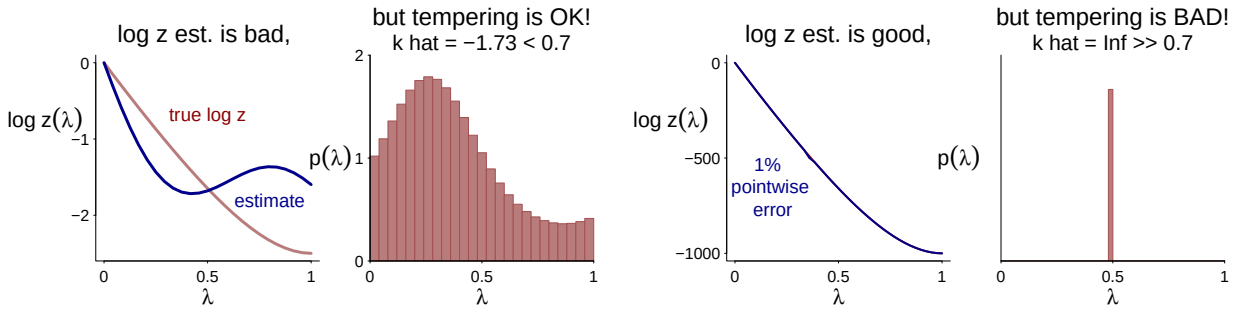


Figure 6.13: *An illustration of the different orientations in normalizing constant estimation and simulated tempering. In the left two panels, the blue curve poorly fits the true log normalizing constant, but there is enough mass everywhere in the resulting marginal density to ensure a complete path in simulated tempering. The right two panels: with a much larger scale, when log z estimation is off by 1% at a single point, the resulting path becomes a point mass in* λ*. In continuous tempering, such failure is diagnosed by a large* $\hat{k}$*.*

most existing tempering methods adopt this form. If there is rapid phase transition at some critical temperature, any power-transformation tempering will be prohibitively slow (Bhatnagar and Randall, 2004). Fortunately, the general framework in Section 5.2 permits an arbitrary path formulations, and is easily implemented in Stan by replacing the closed form gradient $U=$ log likelihood by automatic differentiation. We leave this more flexible tempering path for future research.

General recommendations We have developed a method that integrates adaptive path sampling and tempering and have applied it to several examples. The procedure is highly automated, and we have implemented it in an R package using the general-purpose Bayesian inference engine Stan, returning both the desired posterior sample and the estimated log normalizing constant at convergence. See Appendix B for software details.

In Bayesian computation, the ultimate goal is not to stop at the posterior simulation draws, but to use them to check and improve the model in a workflow. If data come from an identifiable model, then with reasonable sample size we can expect to distinguish among parameters and obtain a well behaved posterior distribution (eventually achieving asymptotic normality). From this perspective, multimodal posteriors should be unlikely with large sample size and may represent data that do not fit the model. Hence, it is crucial to check the model fit even after the target posterior is obtained

from our proposed sampling algorithms.

Finally, although we have argued its relative advantage over existing methods, we do not think the proposed method based on adaptive sampling and tempering can solve all metasable sampling problems, either because of a badly-chosen base measurement, or the dimension and sample size limitation imposed by tempering itself. In this case, the Pareto-$\hat{k}$ diagnostic is still useful to understand why the method fails. Besides refining the base measurement and potentially modifying the model, another alternative strategy to metastable sampling is to use cross validation and multi-chain stacking (Yao et al., 2020b) to combine the non-mixed simulation draws, which in effect changes the target distribution.

6.7 Software implementation

To provide an `R` (R Core Team, 2020) interface of path sampling and continuous tempering, we create a package `pathtemp`, with the underlying execution inside the general-purpose Bayesian inference engine `Stan` (Stan Development Team, 2020). The source code is available at `https://github.com/yao-yl/path-tempering`. The procedure is highly automated and requires minimal tuning.

To install the package, call

```
devtools::install_github("yao-yl/path-tempering/package/pathtemp",upgrade="never")
```

We demonstrate the practical implementation of continuous tempering on a Cauchy mixture example. Consider the following Stan model:

```
data {
real y;
}
parameters {
real theta;
}
model{
y ~ cauchy(theta, 0.2);
-y ~ cauchy(theta, 0.2);
}
```

To run continuous tempering, a user can specify any base model, say $\theta \sim \text{normal}(0, 5)$, and list it in an `alternative model` block as if it is a regular model.

```
...
model{     // keep the original model
y ~ cauchy(theta,0.2);
-y ~ cauchy(theta,0.2);
}
alternative model{    // add a new block of the base measure (e.g., the prior).
theta ~ normal(0,5);
}
```

After saving this code to a stan file `cauchy.stan`, we run the function `code_temperature_augment()`, which automatically constructs a tempered path between the orginal model and the alternative model, and generates a working model named `cauchy_augmented.stan`:

```
library(pathtemp)
update_model <- stan_model("solve_tempering.stan")
file_new <- code_temperature_augment("cauchy.stan")
> output:
> A new stan file has been created: cauchy_augmented.stan.
```

We have automated path sampling and its adaptation into a function `path_sample()`. The following two lines realize adaptive path sampling.

```
sampling_model <- stan_model(file_new) #  translated to C++ code
path_sample_fit <- path_sample(data=list(gap=10), # data list in original model
sampling_model=sampling_model)
```

The returned value `path_sample_fit` provides access to the posterior draws θ from the target density and base density, the join path in the (θ, a) space in the final adaptation, and the estimated log normalizing constant $\log z(\lambda)$.

```
sim_cauchy <- extract(path_sample_fit$fit_main)
in_target <- sim_cauchy$lambda==1
in_prior  <- sim_cauchy$lambda==0
# sample from the target
hist(sim_cauchy$theta[in_target])
# sample from the base
hist(sim_cauchy$theta[in_prior])
# the joint "path"
plot(sim_cauchy$a, sim_cauchy$theta)
# the normalizing constant
plot(g_lambda(path_sample_fit$path_post_a), path_sample_fit$path_post_z)
```

Second, this automated procedure enables to fit two models together. The following Stan code fits a regression with both the probit and logit link. A path between them effectively expands the

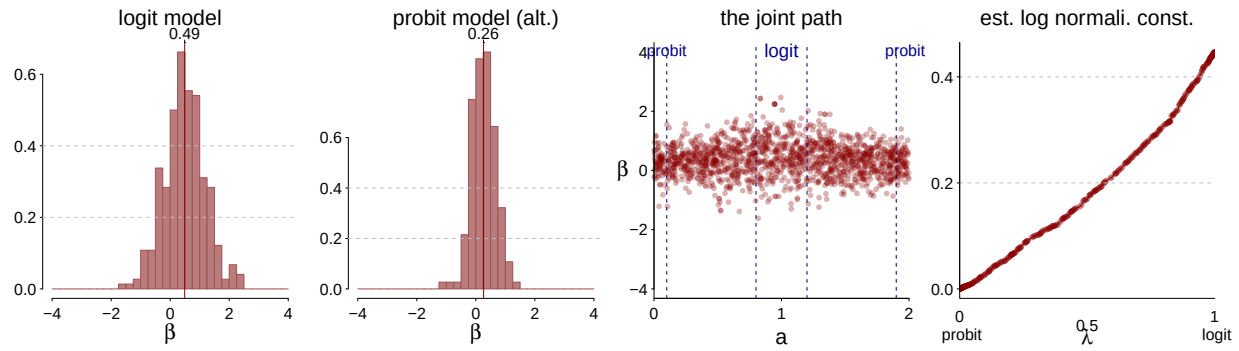


Figure 6.14: Output from the logit-probit code example: the posterior draws β from the probit and logit link, the join path in the (β, a) space in the final adaptation, and the estimated log normalizing constant $\log z(\lambda)$. The visualization is based on 2 adaptations and S=1500 posterior draws.

model continuously such both individual model are special cases of the augmented model. The computational efficiency is enhanced as we are fitting one slightly larger model rather than fitting two models. In addition, the log normalizing constant tells us which models fits the data better, which is related to but distinct from the log Bayes factor in model comparisons. The output in one run is presented in Figure 6.14. The data favor the logit link accordingly.

```
data {
int n;
int y[n];
real x[n];
}
parameters {
real beta;
}
model {
beta ~ normal (0,2);
y ~ bernoulli_logit(beta * x);  // logistic regression
}
alternative model {
beta ~ normal (0,1);  // can be a different prior
y ~  bernoulli(Phi(beta * x)); // probit regression
}
```

Part III

Learning from many models

by Bayesian stacking

Chapter 7. Bayesian model averaging and stacking [1]

"Double, double, toil and trouble; Fire burn and cauldron bubble."
—Wayward Sisters, *Macbeth*

7.1 From model selection to model combination

Bayesian inference provides a coherent workflow for data analysis, parameter estimation, outcome prediction, and uncertainty quantification. However, the model uncertainty is not automatically calibrated: the posterior distribution is always conditioning on the model we use, in which the true data generating mechanism is almost never included. No matter if viewed from the perspective of a group of modelers holding different subjective beliefs, or a single modeler revising belief models through the routine of model check and criticism, or the need of expanding plausible models for flexibility and expressiveness, it is common in practice to obtain a range of possible belief models.

In Section 7.1, we review Bayesian decision theory, through which the model comparison, model selection, and model combination are viewed in a unified framework. The estimation of the expected utility depends crucially on how the true data generating process is modeled, and is described by different $\mathcal{M}$-views. We compare Bayesian model averaging and leave-one-out (LOO) based Bayesian stacking in Section 7.2, which corresponds to the $\mathcal{M}$-closed and $\mathcal{M}$-open view respectively. To explain why these methods work, we discuss related asymptotic theories in Section 7.4. In Section 7.5, we investigate the computation efficiency, and demonstrate an importance-sampling based implementation in `Stan` and `R` package `loo`. We also consider several generalizations in non-iid data.

[1] This chapter is a slight modified version of Yao et al. (2018a) and Yao (2019).

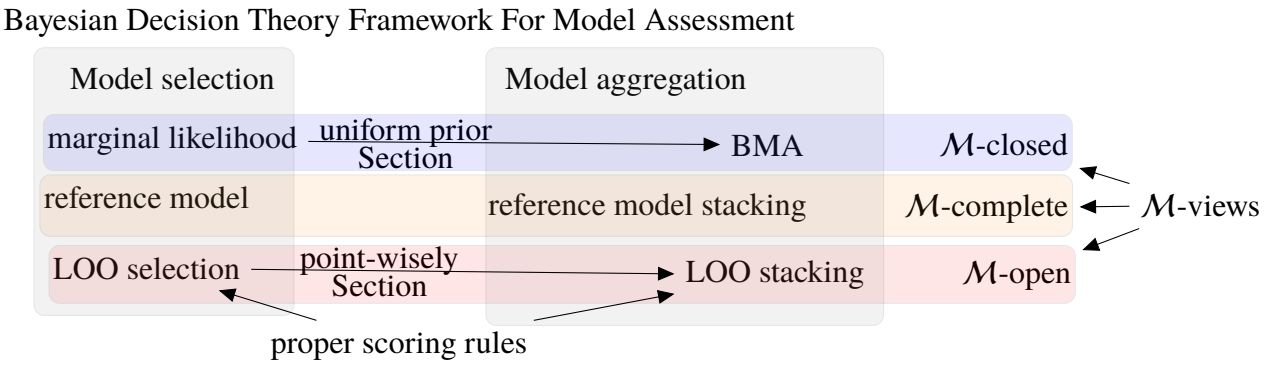


Figure 7.1: *The organization and connections of concepts in this chapter.*

The Bayesian decision framework for model assessment. We denote $D = \{(y_1, x_1), \ldots, (y_n, x_n)\}$ a sequence of observed outcomes $y \in \mathcal{Y}$ and covariates $x \in \mathcal{X}$. The unobserved future observations are $(\tilde{x}, \tilde{y})$. In a predictive paradigm (Bernardo and Smith, 1994; Vehtari and Ojanen, 2012), the statistical inference should be inference on observable quantities such as the future observation $\tilde{y}$, where Bayesian decision theory gives a natural framework for the prediction evaluation. Therefore, we can view model comparison, model selection, as well as model combination as formal Bayesian decision problems. At a higher level, whether to make a single model selection or model combination is part of the decision.

Given a model M with its parameter vector θ, we compute the posterior predictive density $p(\tilde{y}|y, M) = \int p(\tilde{y}|\theta, M)p(\theta|y, M)d\theta$, where we have suppressed the dependence on x for brevity. To evaluate how close the prediction is to the *truth*, we construct the utility function of the predictive performance through *scoring rules*. In general, conditioning on $\tilde{x}$, the unobserved future outcome $\tilde{y}$ is the random variable in sample space $(\mathcal{Y}, \mathcal{A})$. $\mathcal{P}$ is a convex class of probability measure on $\mathcal{Y}$. Any member of $\mathcal{P}$ is called a probabilistic forecast. A scoring rule (Gneiting and Raftery, 2007) is a function $S : \mathcal{P} \times \mathcal{Y} \to [\infty, \infty]$ such that $S(P, \cdot)$ is $\mathcal{P}$-quasi-integrable for all $P \in \mathcal{P}$. In the continuous case, every distribution $P \in \mathcal{P}$ is identified with its density function p.

For two probability measures P and Q, we write $S(P, Q) = \int_{\mathcal{Y}} S(P, \omega)dQ(\omega)$. A scoring rule S is called *proper* if $S(Q, Q) \geq S(P, Q)$ and *strictly proper* if equality holds only when $P = Q$ almost surely. A proper scoring rule defines the divergence $d : \mathcal{P} \times \mathcal{P} \to [0, \infty)$ as

$d(P,Q) = S(Q,Q) - S(P,Q)$. For continuous variables, some popularly used scoring rules include:

- *Quadratic score:* $\mathrm{QS}(p,\tilde{y}) = 2p(\tilde{y}) - ||p||_2^2$ with the divergence $d(p,q) = ||p-q||_2^2$.
- *Logarithmic score:* $\mathrm{LogS}(p,\tilde{y}) = \log p(\tilde{y})$ with $d(p,q) = \mathrm{KL}(q,p)$. The logarithmic score is the only proper local score assuming regularity conditions.
- *Continuous-ranked probability score:* $\mathrm{CRPS}(F,\tilde{y}) = -\int_{\mathbb{R}}(F(\tilde{y}') - 1(y' \geq \tilde{y}))^2 dy'$ with $d(F,G) = \int_{\mathbb{R}}(F(\tilde{y}) - G(\tilde{y}))^2 d\tilde{y}$, where F and G are the corresponding distribution functions.
- *Energy score:* $\mathrm{ES}(P,y) = \frac{1}{2}\mathbb{E}_P||Y-Y'||_2^\beta - \mathbb{E}_P||Y-y||_2^\beta$, where Y and Y' are two independent random variables from distribution P. When $\beta = 2$, this becomes $\mathrm{ES}(P,\tilde{y}) = -||\mathbb{E}_P(\tilde{y}) - \tilde{y}||^2$. The energy score is strictly proper when $\beta \in (0,2)$ but not when $\beta = 2$.
- *Scoring rules depending on first and second moments:* Examples include $S(P,\tilde{y}) = -\log\det(\Sigma_P) - (\tilde{y}-\mu_P)^T\Sigma_p^{-1}(y-\mu_P)$, where μ_P and Σ_P are the mean vector and covariance matrix of distribution P.

In such framework, the expected utility for any posterior predictive distribution $p(\cdot)$ is

$$\mathbb{E}_{\tilde{y}} S\left(p(\cdot),\tilde{y}\right) = \int_{\mathcal{Y}} S(p,\tilde{y}) p_t(\tilde{y}|y) d\tilde{y}, \tag{7.1}$$

where $p_t(\tilde{y}|y)$ is the unknown true data generating density of outcomes $\tilde{y}$ given current observations.

With the widely used logarithm score, the expected log predictive density (elpd) of model M is

$$\mathrm{elpd} = \int_{\mathcal{Y}} \log p(\tilde{y}|y,M) p_t(\tilde{y}|y) d\tilde{y}. \tag{7.2}$$

The general decision problem is an optimization problem that maximizes the expected utility within some decision space $\mathcal{P}$: $p^{\mathrm{opt}} = \arg\max_{p\in\mathcal{P}} \int S(p,\tilde{y}) dp_t(\tilde{y})$. *Model selection* can be viewed as a sub decision space of *model combination*, by restricting model weights to have only one non-zero entry. In such sense, model selection may be unstable and wasteful of information.

The expected scoring rule (7.1) depends on the generating process of $\tilde{y}$, which is unknown in the first place. How we will estimate such expectation depends on how we view the relation between belief models and the true generating process, i.e., three $\mathcal{M}$-views.

Remodeling: $\mathcal{M}$-closed, $\mathcal{M}$-complete, and $\mathcal{M}$-open views. Bernardo and Smith (1994) classified model comparison problems into three categories: $\mathcal{M}$-closed, $\mathcal{M}$-complete and $\mathcal{M}$-open.

- In $\mathcal{M}$-closed problems, the true data generating process can be expressed by one of $M_k \in \mathcal{M}$, although it is unknown to researchers.
- $\mathcal{M}$-complete refers to the situation where the true model exists and is out of model list $\mathcal{M}$. But we still wish to use a model M^* because of tractability of computations or communication of results, compared with the actual belief model.
- The $\mathcal{M}$-open perspective acknowledges the true model is not in $\mathcal{M}$, and we cannot specify the explicit form $p(\tilde{y}|y)$ because it is too difficult conceptually or computationally, we lack time to do so, or do not have the expertise, etc.

Computing the integral (7.1) requires a model for $\tilde{y}$. The inference and model assessment can have different model assumptions, akin to the distinction between estimation and hypothesis testing in frequentist statistics. For $\mathcal{M}$-closed and $\mathcal{M}$-complete problems, we specify a belief model M^* that we believe to be or well approximate the data generate process, and we describe all uncertainty related to future data in the belief model M^* through $p(\tilde{y}|y, M^*)$. The expected utility of any prediction Q is estimated by

$$\mathbb{E}_{\tilde{y}} S(Q, \tilde{y}) \approx \int_{\mathcal{Y}} S(Q, \tilde{y}) p(\tilde{y}|y, M^*) d\tilde{y}. \tag{7.3}$$

$\mathcal{M}$-closed and $\mathcal{M}$-complete are a simplification of reality. No matter how flexible the belief model M^* is, there is little reason to believe it reflects the *truth*, unless in rare situations such as computer simulations. Although such simplification is sometimes useful, the stronger assumption may also

result in an unverifiable and irretrievably bias in (7.1), which will further lead to an undesired performance in model aggregation.

In $\mathcal{M}$-open problems, we still rely on models in $\mathcal{M}$ in inference and prediction. But we make minimal assumptions in the model assessment phase. Cross-validation is a widely used strategy to this end, where we re-use samples $y_1, \ldots, y_n$ as pseudo Monte Carlo draws from the true data generating process without having to model it explicitly. For example, the leave-one-out (LOO) predictive density of a model M is a consistent estimation of (7.2).

$$\text{elpd}_{\text{loo}} = \frac{1}{n}\sum_{i=1}^{n} \log p(y_i|y_{-i}, M) = \frac{1}{n}\sum_{i=1}^{n} \int \log p(y_i|\theta, M) p(\theta|M, y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_n) d\theta.$$

7.2 From Bayesian model averaging to Bayesian stacking

We have a series of models $\mathcal{M} = \{M_1, \ldots, M_K\}$, each having parameter vectors $\theta_k \in \Theta_k$. In general θ_k can have different dimensions and interpretations, and some may be infinite dimensional too. We denote the likelihood and prior in the k-th model by $p(y|\theta_k)$ and $p(\theta_k|M_k)$. The goal is to aggregate all component predictive distributions $\{p(\tilde{y}|y, M), M \in \mathcal{M}\}$. Adopting different $\mathcal{M}$-views, we will solve the problem by various methods as follows.

$\mathcal{M}$-closed: Bayesian model averaging. Bayesian model averaging (BMA) assigns a prior both to the model space $p(M_k)$ and parameters $p(\theta_k|M_k)$. Through Bayes rule, the posterior probability of model k is proportional to the product of its prior and marginal likelihood,

$$p(M_k|y) = \frac{p(y|M_k)p(M_k)}{\sum_{k'=1}^{K} p(y|M'_k)p(M'_k)}.$$

In particular, the aggregated posterior predictive distribution of new data $\tilde{y}$ is estimated by

$$p_{\text{BMA}}(\tilde{y}|y) = \sum_{k=1}^{K} p(\tilde{y}|M_k, y)p(M_k|y).$$

In $\mathcal{M}$-closed cases, BMA is optimal if the method is evaluated based on its frequency properties

assessed over the joint prior distribution of the models and their internal parameters (Madigan et al., 1996; Hoeting et al., 1999). In $\mathcal{M}$-open and $\mathcal{M}$-complete cases, BMA almost always asymptotically *select* the one single model on the list that is closest in Kullback-Leibler (KL) divergence, compromising the extra expressiveness of model aggregation.

Furthermore, BMA is contingent on the marginal likelihood $p(y|M_k) = \int p(y|\theta_k)p(\theta_k|M_k)d\theta_k$, which will be sensitive to the prior $p(\theta_k|M_k)$. A correct specification of the model (an $\mathcal{M}$-closed view) is stronger than the asymptotic convergence to truth in some model, as it also requires the prior to be correctly chosen in terms of reflecting the actual population distribution of the underlying parameter. For example, consider observations $y_1, \ldots, y_n$ generated from $y \sim \text{normal}(0, 0.1^2)$, and a normal-normal model: $y \sim \text{normal}(\mu, 1)$ with a prior $\mu \sim \text{normal}(0, 10^2)$. Such prior is effectively flat on the range of observed y. However, a change of prior to $\mu \sim \text{normal}(0, 100^2)$ or $\text{normal}(0, 1000^2)$ would divide the marginal likelihood, and thereby the posterior probability, by roughly a factor of 10 or 100.

$\mathcal{M}$-open: stacking. Stacking is originated from machine learning for the purpose of pooling point estimates from multiple regression models (Wolpert, 1992; Breiman, 1996b; LeBlanc and Tibshirani, 1996). Clyde and Iversen (2013), Le and Clarke (2017), and Yao et al. (2018a) develop and extend its Bayesian interpretation.

The ultimate goal of stacking a set of K predictive distributions built from the model list $\mathcal{M} = (M_1, \ldots, M_K)$ is to find the predictive distribution with the form of a linear pooling $C = \{\sum_{k=1}^K w_k p(\cdot|M_k) : \sum_k w_k = 1, w_k \geq 0\}$ that is optimal according to a specified utility. The decision to make is the model weights w, which has to be a length-K simplex $w \in \mathbb{S}_1^K = \{w \in [0,1]^K : \sum_{k=1}^K w_k = 1\}$. Given a scoring rule S, or equivalently the divergence d, the optimal stacking weight should solve

$$\max_{w \in \mathbb{S}_1^K} S\Big(\sum_{k=1}^K w_k p(\cdot|y, M_k), p_t(\cdot|y)\Big) \text{ or equvalently } \min_{w \in \mathbb{S}_1^K} d\Big(\sum_{k=1}^K w_k p(\cdot|y, M_k), p_t(\cdot|y)\Big), \tag{7.4}$$

where $p(\tilde{y}|y, M_k)$ is the predictive density of new data $\tilde{y}$ in model M_k that has been trained on

observed data y and $p_t(\tilde{y}|y)$ refers to the true distribution.

With an $\mathcal{M}$-open view, we empirically estimate the optimal stacking weight in (7.4) by replacing the full predictive distribution $p(\tilde{y}|y, M_k)$ evaluated at a new data point $\tilde{y}$ with the corresponding LOO predictive distribution $\hat{p}_{k,-i}(y_i) = \int p(y_i|\theta_k, M_k)p(\theta_k|y_{-i}, M_k)d\theta_k$.

Therefore, it suffices to solve the following optimization problem

$$\hat{w}^{\text{stacking}} = \max_{w \in \mathbb{S}_1^K} \frac{1}{n} \sum_{i=1}^{n} S\Big(\sum_{k=1}^{K} w_k \hat{p}_{k,-i}, y_i\Big). \tag{7.5}$$

The aggregated predictive distributions on new data $\tilde{y}$ is $p_{\text{stacking}}(\tilde{y}|y) = \sum_{k=1}^{K} \hat{w}_k^{\text{stacking}} p(\tilde{y}|y, M_k)$.

In terms of Vehtari and Ojanen (2012, Section 3.3), stacking predictive distributions (7.5) is the M^*-optimal projection of the information in the actual belief model M^* to $\hat{w}$, where explicit specification of M^* is avoided by re-using data as a proxy for the predictive distribution of the actual belief model and the weights w_k are the free parameters.

Choice of utility. The choice of the scoring rule should depend on the underlying application and researchers' interest. Generally we recommend logarithmic score because (a) log score is the only proper local scoring rule, and (b) the easy interpretation of the underlying Kullback-Leibler divergence. When using logarithmic score we name (7.5) as *stacking of predictive distributions*:

$$\max_{w \in \mathbb{S}_1^K} \frac{1}{n} \sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k p(y_i|y_{-i}, M_k). \tag{7.6}$$

$\mathcal{M}$-complete: reference-model stacking. It is possible to replace cross-validation with a non-parametric reference model M^*. Plug it into (7.3) we compute the expected utility and further optimize over stacking weights, which we will call reference-model stacking. We can either stack component models $p(\tilde{y}|M_k)$, or stack the projected component models using a projection predictive approach which projects the information from the reference model to the restricted models (Piironen and Vehtari, 2017a). However in general it is challenging to construct a useful reference model, as then there is probably no need for model averaging.

The connection between BMA and stacking. BMA, and more generally marginal likelihood based model evaluation, can also be viewed as a special case of the utility-based model assessment.

First, under an $\mathcal{M}$-closed view, we believe the data is generated from one of the model $M^* \in \mathcal{M}$ in the candidate model list. We consider a zero-one utility by an indicator function of whether the model has been specified correctly:

$$u(M^*, M_k) = \mathbb{1}(M^* = M_k). \tag{7.7}$$

Then the expected utility M_k is $\int \mathbb{1}(M^* = M_k)p(M^*|y)dM^* = p(M_k|y)$, which is exactly the posterior model probability $p(M_k|y)$ in BMA. Hence the decision maker will pick the model with the largest posterior probability, which is equivalent to the approach of Bayes factor. Interestingly, the model with the largest BMA weight is also the model to be selected under the zero-one utility, whereas in general the model with the largest stacking weight is not necessarily single-model-selection optimal (see discussions in Section 7.4)

Second, under the $\mathcal{M}$-closed view the information about unknownness is contained in the posterior distribution $p(M_k, \theta_k|y)$, and the actual beliefs about the future observations are described by the BMA predictive distribution. Using (7.3) and (7.4), stacking over the logarithmic score reads

$$\max_{w \in \mathbb{S}_1^K} \int_{\mathcal{Y}} \log(\sum_{k\prime=1}^{K} w_{k\prime} p(\tilde{y}|M_{k\prime}, y)) \sum_{k=1}^{K} p(M_k|y)p(\tilde{y}|M_k, y)d\tilde{y},$$

whose optimal solution is always the same as the BMA weight $w_k^{\text{opt}} = p(M_k|y)$, as the logarithmic score is strictly proper.

In practice it is nearly impossible to either come up with an exhaustive list of possible candidate models that encompasses the true data generating process, or to formulate the true prior that reflects the population. It is not surprising that stacking typically outperforms BMA in various prediction tasks (see extensive simulations in Clarke, 2003; Yao et al., 2018a). Notably, in the large sample limit, BMA assigns weight 1 to the closest model to the true data generating process measured in KL divergence, regardless of how close other slightly more wrong models are. It effectively

becomes model selection and yields practically spurious and overconfident results (e.g., Yang and Zhu, 2018) in $\mathcal{M}$-open problems.

7.3 Other related methods and generalizations.

Pseudo-BMA. When the marginal likelihood in BMA is hard to evaluate, it can be approximated by information criterion. In Pseudo Bayes factors (Geisser and Eddy, 1979; Gelfand, 1996), we replace the marginal likelihoods $p(y|M_k)$ by a product of Bayesian leave-one-out cross-validation predictive densities $\prod_{i=1}^{n} p(y_i|y_{-i}, M_k)$. Yao et al. (2018a) propose another information criterion based weighting scheme named Pseudo-BMA weighting. The weight for model k is proportional to the exponential of the model's estimated elpd: $w_k \propto \exp(\widehat{\text{elpd}}_{\text{loo}}^{k})$. Alternatively, such quantity can be estimated using a nonparametric reference model in $\mathcal{M}$-complete views (Li and Dunson, 2019). We may further take into account the sampling variance in cross-validation, and average over weights in multiple Bayesian bootstrap resamples (Yao et al., 2018a). The information criterion weighting is computationally easier, but should only be viewed as an approximation to the more desired stacking weights.

Pseudo-BMA+. The pseudo-BMA estimation doesn't take into account the uncertainty resulting from having a finite number of proxy samples from the future data distribution. Taking into account the uncertainty would regularize the weights making them go further away from 0 and 1. The Bayesian bootstrap (BB) can be used to compute uncertainties related to LOO estimation (Vehtari and Lampinen, 2002). The Bayesian bootstrap (Rubin, 1981) gives simple non-parametric approximation to the distribution. Having samples of $z_1, \ldots, z_n$ from a random variable Z, it is assumed that posterior probabilities for all observed z_i have the distribution Dirichlet$(1, \ldots, 1)$ and values of Z that are not observed have zero posterior probabilities. We define $z_i^k = \widehat{\text{elpd}}_{\text{loo},i}^{k}$, $i = 1, \ldots n$, and sample vectors $(\alpha_{1,b}, \ldots, \alpha_{n,b})_{b=1,\ldots,B}$ from the Dirichlet $(\overbrace{1, \ldots, 1}^{n})$ distribution, and compute the weighted means, $\bar{z}_b^k = \sum_{i=1}^{n} \alpha_{i,b} z_i^k$. Then a Bayesian bootstrap sample of w_k with size

B is,

$$w_{k,b} = \frac{\exp(n\bar{z}_b^k)}{\sum_{k=1}^K \exp(n\bar{z}_b^k)}, \quad b = 1, \ldots, B,$$

and the final adjusted weight of model k is,

$$w_k = \frac{1}{B} \sum_{b=1}^{B} w_{k,b},$$

which we call Pseudo-BMA+ weight.

iBMA. We may combine the cross-validation and BMA. Intrinsic Bayesian model averaging (iBMA, Berger and Pericchi, 1996) enables improper prior, which is not allowed in BMA. It first partitions samples into a small training set $y(l)$ and remaining $y(-l)$, and replaces the marginal likelihood by partial likelihood $\int p(y(-l)|M_k, \theta_k) p(\theta_k|y(l), M_k) d\theta$. The final weight is the average across some or all possible training samples. An alternative is to avoid averaging over all subsets and use the fractional Bayes factor (O'Hagan, 1995). iBMA is more robust for models with vague priors, but is reported to underperform stacking.

All model aggregation techniques introduced so far are two-step procedures, where we first fit individual models and then combine all predictive distributions. It is also possible to conduct both steps jointly, which can be viewed as a decision problem on both the model weights and component predictive distributions. Ideally, we may avoid the model combination problem by extending the model to include the separate models M_k as special cases. A finite-component mixture model is the easiest model expansion, but is generally quite expensive to make inference. Further, if the sample size is small or several components in the mixture could do the same thing, the mixture model can face non-identification or instability. In fact, the immunity to duplicate models is a unique feature of stacking, while many methods including BMA, information criterion weighting and mixture models often have a disastrous performance in face of many similar weak models.

Apart from combining *models*, when we fit one single model but unstable computation, model averaging techniques are also useful to combine *inference* results from multiple non-mixing runs.

This is related to the idea of bagging (Breiman, 1996a). In particular, when the posterior density $p(\theta|y)$ from a model contains multiple isolated modes, Markov chain Monte Carlo (MCMC) algorithms can have difficulty moving between modes. We will explore this idea in the next chapter.

7.4 Properties of stacking

Model aggregation is no worse than model selection. The stacking estimate (7.4) finds the optimal predictive distribution within the linear combination that is the closest to the data generating process with respect to the chosen scoring rule. Solving for the stacking weights in (7.6) is an M-estimation problem. To what extent shall we worry about the finite sample error in leave-one-out cross-validation? Roughly speaking, as long as there is consistency for single model cross-validation, then asymptotically model averaging never does worse than model selection in terms of prediction (Clarke, 2001). Le and Clarke (2017) further prove that under some mild conditions, for either the logarithmic scoring rule or the energy score (negative squared error) and a given set of weights $w_1 \dots w_K$, the weighted leave-one-out-score is a consistent estimate as sample size $n \to \infty$,

$$\frac{1}{n}\sum_{i=1}^{n} S\Big(\sum_{k=1}^{K} w_k \hat{p}_{k,-i}, y_i\Big) - \mathbb{E}_{\tilde{y}|y} S\Big(\sum_{k=1}^{K} w_k p(\tilde{y}|y, M_k), \tilde{y}\Big) \xrightarrow{L_2} 0.$$

In this sense, stacking gives optimal combination weights asymptotically, and is an approximation to the Bayes action.

Stacking viewed as pointwise model selection. Besides justified by the decision theory, stacking weights also have a probabilistic interpretation. To see this, we divide the input-output product space $\mathcal{X} \times \mathcal{Y}$ into K disjoints subsets based on which model performs the best,

$$\mathcal{J}_k \coloneqq \{(\tilde{x}, \tilde{y}) \in \mathcal{X} \times \mathcal{Y} : p(\tilde{y}|M_k, \tilde{x}) > p(\tilde{y}|M_{k\prime}, \tilde{x}), \forall k' \neq k\}.\ k = 1, \dots, K.$$

under some separation condition, the log score stacking weight (7.5) is approximately the probability of the model being the locally best fit: $w_k^{\text{stacking}} \approx \Pr(\mathcal{J}_k)$, where the probability is taken with respect to the joint true data generating process.

Moreover, the advantage of model averaging comes from the fact that model can behave differently in different regions in (x, y) space. Let $\rho = \sup_k \Pr(\mathcal{J}_k)$, then $1 - \rho$ is a rough description of the diversity of models. In terms of the expected log predictive density (elpd), under the separation condition, the gain from the optimally weighted models (against model selection) is lower bounded by $\text{elpd}_{\text{stacking}} - \sup_k \text{elpd}_k \geq L(1 - \rho)(1 - \epsilon) - \log K$.

We omit details here and will visit this perspective in Chapter 10.

Selection or averaging? One practical difficulty in model comparison is to determine how large the difference between model performance is "significant" and whether to discard bad models (Sivula et al., 2020). The probabilistic approximation in the previous subsection suggests that an overall weak model can still be useful in the aggregation. As long as a model is better than all remaining models in some subset of data, this model possesses a non-zero stacking weight no matter how poorly it fits everywhere else.

Lastly, a model with the largest BMA weight (assuming equal prior) is optimal under marginal likelihood model selection. In contrast, a model with the largest stacking weight is not necessarily optimal in terms of single model selection: it may outperform other models most of the time but also have arbitrarily low elpd in the remaining areas—stacking is not designed for model selection. Hence, we do not recommend to discarded models with small weights from the average.

7.5 Stacking in practice

Practical implementation using Pareto smoothed importance sampling. Stacking (7.5) requires leave-one-out (LOO) predictive density $p(y_i|y_{-i}, M_k)$, whose exact evaluation needs to refit each model n times. k-fold cross-validation is computationally cheaper but may introduce higher bias. Vehtari et al. (2017) proposed the an approximate method for Bayesian LOO. It is based on

the importance sampling identity:

$$p(\theta|y_{-i}) \propto \frac{1}{p(\theta|y_i)} p(\theta|y_1, \ldots, y_n).$$

In the k-th model, we fit to all the data, obtaining S simulation draws $\theta_k^s (s = 1, \ldots S)$ from the full posterior $p(\theta_k|y, M_k)$ and calculate

$$r_{i,k}^s = \frac{1}{p(y_i|\theta_k^s, M_k)} \propto \frac{p(\theta_k^s|y_{-i}, M_k)}{p(\theta_k^s|y, M_k)}. \tag{7.8}$$

A direct importance sampling often has high or infinite variance and we remedy it by Pareto smoothed importance sampling (PSIS, Vehtari et al., 2019b). For each fixed model k and data y_i, we fit the generalized Pareto distribution to a set of largest importance ratios $r_{i,k}^s$, and calculate the expected values of the order statistics of the fitted generalized Pareto distribution. These values are used to obtain the smoothed importance weight $w_{i,k}^s$, which is used to replace $r_{i,k}^s$. PSIS-LOO importance sampling computes the LOO predictive density as

$$p(y_i|y_{-i}, M_k) = \int p(y_i|\theta_k, M_k) \frac{p(\theta_k|y_{-i}, M_k)}{p(\theta_k|y, M_k)} p(\theta_k|y, M_k) d\theta_k \approx \frac{\sum_{s=1}^S w_{i,k}^s p(y_i|\theta_k^s, M_k)}{\sum_{s=1}^S w_{i,k}^s}.$$

Stacking for multilevel data. Although the illustration in this article is focused on iid data, the leave-one-out consistency only requires the conditional *exchangeability* of outcomes y given x (Bernardo and Smith, 1994, Chapter 6). Roberts et al. (2017) review cross-validation strategies for data with temporal, spatial, hierarchical, and phylogenetic structure. In general, the PSIS-LOO approximation applies to factorizable models $p(y|\theta, x) = \prod_{i=1}^N p(y_i|\theta, x_i)$ such that the pointwise log-likelihood can be obtained easily by computing $\log p(y_i|\theta, x_i)$.

Non-factorizable models can sometimes be factorized by re-parametrization. In a multilevel model with J groups, we denote the group-level and global parameter as θ_m and ψ. The joint

likelihood is

$$p(y|x,\theta,\psi) = \prod_{j=1}^{J}\left[\prod_{n=1}^{N_j} p(y_{jn}|x_{jn},\theta_j)p(\theta_j|\psi)\right]p(\psi), \tag{7.9}$$

where y are partially exchangeable, i.e., y_{mn} are exchangeable in group j, and θ_m are exchangeable. Rearrange the data and denote the group label of (x_i, y_i) by z_i, then (7.9) can be reorganized into the long format $\prod_{i=1}^{N'} p(y_i|x_i, z_i, \theta, \psi)$ so the previous results follow. Depending on whether the prediction task is to predict a new observation within a group, or a new group, we should consider leave-one-point-out or leave-one-group-out cross-validation.

When the future data are known to come from a group j, there are two stacking strategies: (a) apply generic stacking only to observations from the j-th group, which is asymptotically optimal with enough data, but has large variance if the group size is small, and (b) apply stacking to all observations regardless of their group structure, which has smaller variance at the cost of less flexibility. The more preferred hierarchical stacking (Chapter 9) trades off between these two extremes. Its Bayesian hierarchical formulation shares information across groups, stabilizing model weights in small groups while still allowing the flexibility of group-specific weighting.

Stacking for time series data. When observations y_t come in sequence and the main purpose is to make prediction for the next not-yet-observed data, we can use the prequential principle (Dawid, 1984) to factorize the likelihood: $p(y_{1:N}|\theta) = \prod_{t=1}^{N} p(y_t|y_{1:t-1}, \theta)$. In model averaging, we can replace the LOO density $p(y_i|y_{-i})$ in (7.5) by the sequential predictive density leaving out all future data: $p(y_t|y_{<t}) = \int p(y_t|y_{1:t-1}, \theta)p(\theta|y_{1:t-1})d\theta$ in each model, and then stacking follows. The ergodicity of y will yield

$$\lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N} S\left(p(\cdot|y_{<t}), y_t\right) - \lim_{N\to\infty}\frac{1}{N}\mathbb{E}_{Y_{1:N}}\sum_{t=1}^{N} S\left(p(\cdot|Y_{<t}), Y_t\right) \to 0.$$

which implies a similar stacking optimality as discussed in Section 7.4. Geweke and Amisano (2012) investigate this stacking approach in time series data.

When there is a particular horizon of interest for prediction, a model that is good at short

term forecast is not necessarily good for long term forecast. We can extend the one-step ahead $p(y_t|y_{<t})$ to m-step-ahead predictive density $p(y_{t<m}|y_{<t}) = p(y_t, \ldots, y_{t+m-1}|y_1, \ldots, y_{t-1}) = \int p(y_{t<m}|y_{<t}, \theta)p(\theta|y_{<t})d\theta$ in the objective function (Lavine et al., 2021).

In terms of computation, the exact prequential evaluation requires refitting each model for each t, which can be approximated by PSIS as, $p(y_t|y_{<t}) = \int p(y_t|\theta, y_{<t})\frac{p(\theta|y_{<t})}{p(\theta|y)}p(\theta|y)d\theta$. We then start from the full data inference $p(\theta|y)$ and dynamically update $p(\theta|y_{<t})$ using PSIS approximation. When $p(\theta|y_{<t})$ reveals large discrepancy from $p(\theta|y)$ for some small t, we refit the model $p(\theta|y_{<t})$ and update the proposal. Bürkner et al. (2020) verify such approximation gives stable and accurate results with minimal number of refits in time series.

We can further extend the static stacking scheme to a dynamic model weighting, allowing the explanation power of models to change over time. Yao et al. (2021b) present an election forecast example that applies hierarchical stacking to longitudinal polling data. Another flexible model weighting strategy in time series forecasting is Bayesian predictive synthesis (BPS, McAlinn and West, 2019; McAlinn et al., 2020): The predictive density has the form $\int \alpha(y|z) \prod_{k=1:K} h_k(z_k)dz$, where $z = z_{1:K}$ is the latent vector generated from predictive densities $h_k(\cdot)$ in each model and $\alpha(y|z)$ is the distribution for y given z that is designed to calibrate the model-specific biases and correlations.

The choice of model list As we have discussed earlier, BMA and information criterion weighting are undesired against many similar weak models. We may remedy this by a careful construction of priors. For example, George (2010) establishes dilution priors to compensate for model space redundancy in linear models, putting smaller weights on those models that are close to each other. Fokoue and Clarke (2011) introduce prequential model list selection to obtain an optimal model space.

Stacking is prior invariant and immune to model duplication. Nevertheless, all methods discussed in the present chapter fit models separately, and are thereby limited in that they do not pool information between the different model fits. The benefit of stacking depends only on the span of

the model list (Le and Clarke, 2017), and models to be stacked should be as different as possible (Breiman, 1996b). In light of discussion in Section 7.4, the ideal situation of stacking is when models can offer different predictive density pointwisely.

In general, we do not recommend constructing a large list of weak models (e.g., subset regression) and aggregate them in a black box way, as in that setting we would recommend moving to a continuous model space that encompasses all separate models. We prefer to carefully construct component models that would have individually fit the data as much as possible, and all admissible estimators for parameters should be considered before the optimization procedures.

7.6 Examples

Gaussian mixture model. This simple example helps us understand how BMA and stacking behave differently. It also illustrates the importance of the choice of scoring rules when combining distributions. Suppose the observed data $y = (y_i, i = 1, \ldots, n)$ come independently from a normal distribution $\mathrm{N}(3.4, 1)$, not known to the data analyst, and there are 8 candidate models, $\mathrm{N}(\mu_k, 1)$ with $\mu_k = k$ for $1 \leq k \leq 8$. This is an $\mathcal{M}$-open problem in that none of the candidates is the true model, and we have set the parameters so that the models are somewhat separate but not completely distinct in their predictive distributions.

For BMA with a uniform prior $\Pr(M_k) = \frac{1}{8}, k = 1, \ldots, 8$, we can write the posterior distribution explicitly:

$$\hat{w}_k^{\mathrm{BMA}} = P(M_k|y) = \frac{\exp(-\frac{1}{2}\sum_{i=1}^n (y_i - \mu_k)^2)}{\sum_{k'} \exp(-\frac{1}{2}\sum_{i=1}^n (y_i - \mu_{k'})^2)},$$

from which we see that $\hat{w}_3^{\mathrm{BMA}} \xrightarrow{P} 1$ and $\hat{w}_k^{\mathrm{BMA}} \xrightarrow{P} 0$ for $k \neq 3$ as sample size $n \to \infty$. Furthermore,

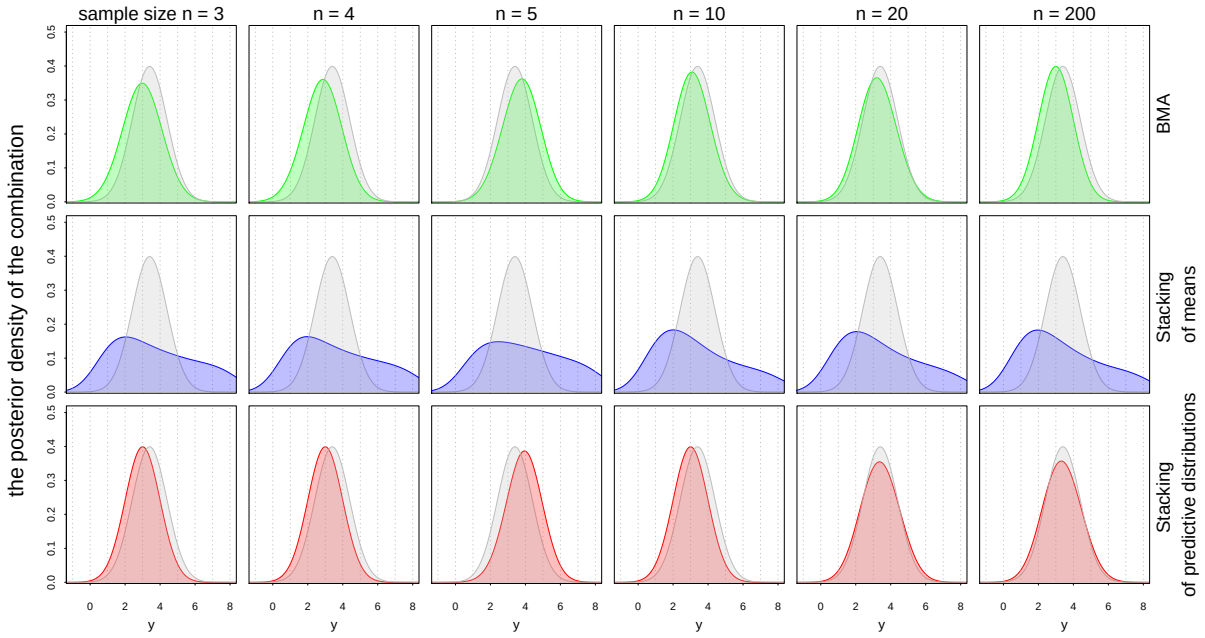


Figure 7.2: *For the Gaussian mixture example, the predictive distribution $p(\tilde{y}|y)$ of BMA (green curve), stacking of means (blue) and stacking of predictive distributions (red). In each graph, the gray distribution represents the true model $N(3.4, 1)$. Stacking of means matches the first moment but can ignore the distribution. For this $\mathcal{M}$-open problem, stacking of predictive distributions outperforms BMA as sample size increases.*

for any given n ,

$$\begin{aligned} \mathrm{E}_{y\sim \mathrm{N}(\mu,1)}[\hat{w}_k^{\mathrm{BMA}}] &\propto E_y\left(\exp(-\frac{1}{2}\sum_{i=1}^{n}(y_i-\mu_k)^2)\right) \\ &\propto \left(\int_{-\infty}^{\infty}\exp\left(-\frac{1}{2}\left((y-\mu_k)^2+(y-\mu)^2\right)\right)dy\right)^n \\ &\propto \exp\left(-\frac{n(\mu_k-\mu)^2}{4}\right), \end{aligned}$$

This example is simple in that there is no parameter to estimate within each of the models: $p(\tilde{y}|y, M_k) = p(\tilde{y}|M_k)$. Hence, in this case the weights from Pseudo-BMA and Pseudo-BMA+ are the same as the BMA weights.

For stacking of means, we need to solve

$$\hat{w} = \arg\min_{w} \sum_{i=1}^{n}(y_i - \sum_{k=1}^{8} w_k k)^2, \quad s.t. \sum_{k=1}^{8} w_k = 1, \quad w_k \geq 0.$$

This is nonidentifiable because the solution contains any vector $\hat{w}$ satisfying

$$\sum_{k=1}^{8} \hat{w}_k = 1, \quad \hat{w}_k \geq 0, \quad \sum_{k=1}^{8} \hat{w}_k k = \frac{1}{n} \sum_{i=1}^{n} y_i.$$

For point prediction, the stacked prediction is always $\sum_{k=1}^{8} \hat{w}_k k = \frac{1}{n} \sum_{i=1}^{n} y_i$, but it can lead to different predictive distributions $\sum_{k=1}^{8} \hat{w}_k \mathrm{N}(k, 1)$. To get one reasonable result, we transform the least squares optimization to the following normal model and assign a uniform prior to w:

$$y_i \sim \mathrm{N}\left(\sum_{k=1}^{8} w_k k, \sigma^2\right), \quad p(w_1, \ldots, w_8, \sigma) = 1.$$

Then we could use the posterior means of w as model weights.

For stacking of predictive distributions,we need to solve

$$\max_{w} \sum_{i=1}^{n} \log\left(\sum_{k=1}^{8} w_k \exp\left(-\frac{(y_k - k)^2}{2}\right)\right), \quad s.t. \sum_{k=1}^{8} w_k = 1, \quad w_k \geq 0$$

In fact, this example is a density estimation problem. Smyth and Wolpert (1998) first applied stacking to non-parametric density estimation, which they called *stacked density estimation*. It can be viewed as a special case of our stacking method.

We compare the posterior predictive distribution $\hat{p}(\tilde{y}|y) = \sum_k \hat{w}_k p(\tilde{y}|y, M_k)$ for these three methods of model averaging. Figure 7.2 shows the predictive distributions in one simulation when the sample size n varies from 3 to 200. Stacking of means (the middle row of graphs) gives an unappealing predictive distribution, even if its point estimate is reasonable. The broad and oddly spaced distribution here arises from nonidentification of w, and it demonstrates the general point that stacking of means does not even try to match the shape of the predictive distribution. The top and bottom row of graphs show how BMA picks up the single model that is closest in KL divergence, while stacking picks a combination; the benefits of stacking becomes clear for large n.

In this trivial non-parametric case, stacking of predictive distributions is almost the same as fitting a mixture model, except for the absence of the prior. The true model $\mathrm{N}(3.4, 1)$ is actually

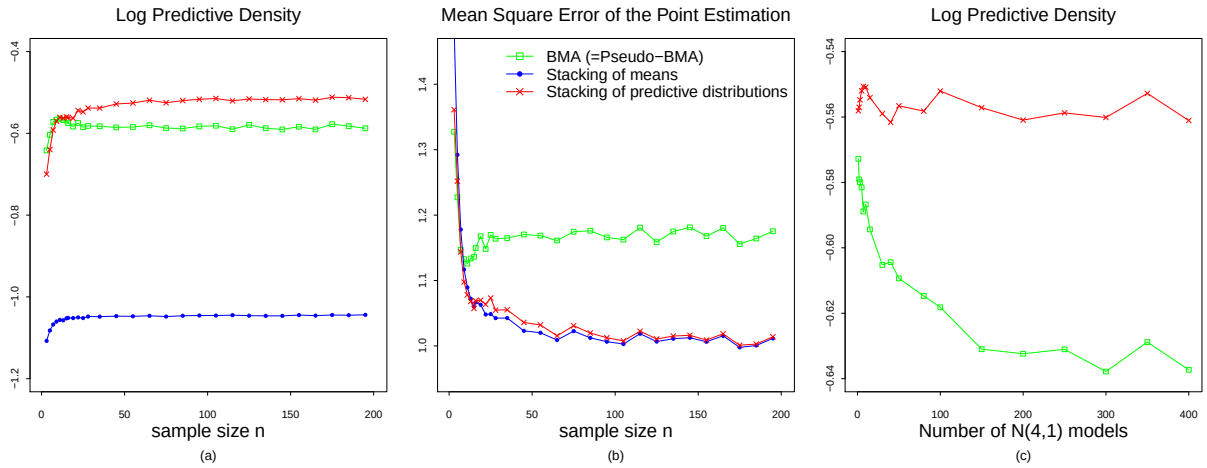


Figure 7.3: *(a) The left panel shows the expected log predictive density of the combined distribution under BMA, stacking of means and stacking of predictive distributions. Stacking of predictive distributions performs best for moderate and large sample sizes. (b) The middle panel shows the mean squared error treating the posterior mean of $\hat{y}$ as a point estimation. Stacking of predictive distributions gives almost the same optimal mean squared error as stacking of means, both of which perform better than BMA. (c) The right panel shows the expected log predictive density of stacking and BMA when adding some more $N(4,1)$ models to the model list, where sample size is fixed to be 15. All average log scores and errors are calculated through 500 repeated simulation and 200 test data generating from the true distribution.*

a convolution of single models rather than a mixture, hence no approach can recover the true one from the model list. From Figure 7.3 we can compare the mean squared error and the mean logarithmic score of these three combination methods. The log scores and errors are calculated through 500 repeated simulations and 200 test data. The left panel shows the logarithmic score (or equivalent, expected log predictive density) of the predictive distribution. Stacking of predictive distributions always gives a larger score except for extremely small n. In the middle panel, it shows the mean squared error by considering the posterior mean of predictive distribution to be a point estimate, even if it is not our focus. In this case, it is not surprising to see that stacking of predictive distributions gives almost the same optimal mean squared error as the stacking of means, both of which are better than the BMA. Two distributions close in KL divergence are close in each moment, while the reverse does not necessarily hold. This illustrates the necessity of matching the *distributions*, rather than matching the *moments*.

Stacking depends only on the space expanded by all candidate models, while BMA or Pseudo-

BMA weighting may by misled by such model expansion. If we add another $\mathrm{N}(4,1)$ as the 9th model in the model list above, stacking will not change at all in theory, even though it becomes non-strictly-convex and has infinite same-height mode. For BMA, it is equivalent to putting double prior mass on the original 4th model, which doubles the final weights for it. The right panel of Figure 7.3 shows such phenomenon: we fix sample size n to be 15 and add more and more $\mathrm{N}(4,1)$ models. As a result, BMA (or Pseudo-BMA weighting) puts larger weight on $\mathrm{N}(4,1)$ and behaves worse, while the stacking is essentially unchanged. It illustrates another benefit of stacking compared to BMA or Pseudo-BMA weights. If the performance of a combination method decays as the list of candidate models is expanded, this may indicate disastrous performance if there are many similar weak models on the candidate list. We are not saying BMA can never work in this case. In fact some other methods are proposed to make BMA overcome such drawbacks. For example, George (2010) establishes dilution priors to compensate for model space redundancy for linear models, putting smaller weights on those models that are close to each other. Fokoue and Clarke (2011) introduce prequential model list selection to obtain an optimal model space. But we propose stacking as a more straightforward solution.

Linear subset regressions. The previous section demonstrates a simple example of combining several different nonparametric models. Now we turn to the parametric case. This example comes from Breiman (1996b) who compares stacking to model selection. Suppose the true model is

$$Y = \beta_1 X_1 + \cdots + \beta_J X_J + \epsilon,$$

where $\epsilon \sim \mathrm{N}(0,1)$. All the covariates X_j are independently from $\mathrm{N}(5,1)$. The number of predictors J is 15. The coefficient β is generated by

$$\beta_j = \gamma \left((1_{|j-4|<h}(h - |j-4|)^2 + (1_{|j-8|<h})(h - |j-8|)^2 + (1_{|j-12|<h})(h - |j-12|)^2 \right),$$

where γ is determined by fixing the signal-to-noise ratio such that

$$\frac{\text{Var}(\sum_j \beta_j X_j)}{1 + \text{Var}(\sum_j \beta_j X_j)} = \frac{4}{5}.$$

The value h determines the number of nonzero coefficients in the true model. For $h = 1$, there are 3 “strong" coefficients. For $h = 5$, there are 15 “weak" coefficients. In the following simulation, we fix $h = 5$. We consider the following two cases:

1. $\mathcal{M}$-open: Each subset contains only one single variable. Hence, the k-th model is a univariate linear regression with the k-th variable X_k. We have $K = J = 15$ different models in total. One advantage of stacking and Pseudo-BMA weighting is that they are not sensitive to prior, hence even a flat prior will work, while BMA can be sensitive to the prior. For each single model $M_k : Y \sim \text{N}(\beta_k X_k, \sigma^2)$, we set prior $\beta_k \sim N(0, 10)$, $\sigma \sim \text{Gamma}(0.1, 0.1)$.

2. $\mathcal{M}$-closed: Let model k be the linear regression with subset $(X_1, \ldots, X_k)$. Then there are still $K = 15$ different models. Similarly, in model $M_k : Y \sim \text{N}(\sum_{j=1}^{k} \beta_j X_j, \sigma^2)$, we set prior $\beta_j \sim N(0, 10), j = 1, \ldots, k$, $\sigma \sim \text{Gamma}(0.1, 0.1)$.

In both cases, we have seven methods for combining predictive densities: (1) stacking of predictive distributions, (2) stacking of means, (3) Pseudo-BMA, (4) Pseudo-BMA+, (5) best model selection by mean LOO value, (6) best model selection by marginal likelihood, and (7) BMA. We generate a test dataset $(\tilde{x}_i, \tilde{y}_i)$, $i = 1, \ldots, 200$ from the underlying true distribution to calculate the out of sample logarithm scores for the combined distribution under each method and repeat the simulation 100 times to compute the expected predictive accuracy of each method.

Figure 7.4 shows the expected out-of-sample log predictive densities for the seven methods, for a set of experiments with sample size n ranging from 5 to 200. Stacking outperforms all other methods even for small n. Stacking of predictive distributions is asymptotically better than any other combination method. Pseudo-BMA+ weighting dominates naive Pseudo-BMA weighting. Finally, BMA performs similarly to Pseudo-BMA weighting, always better than any kind of model

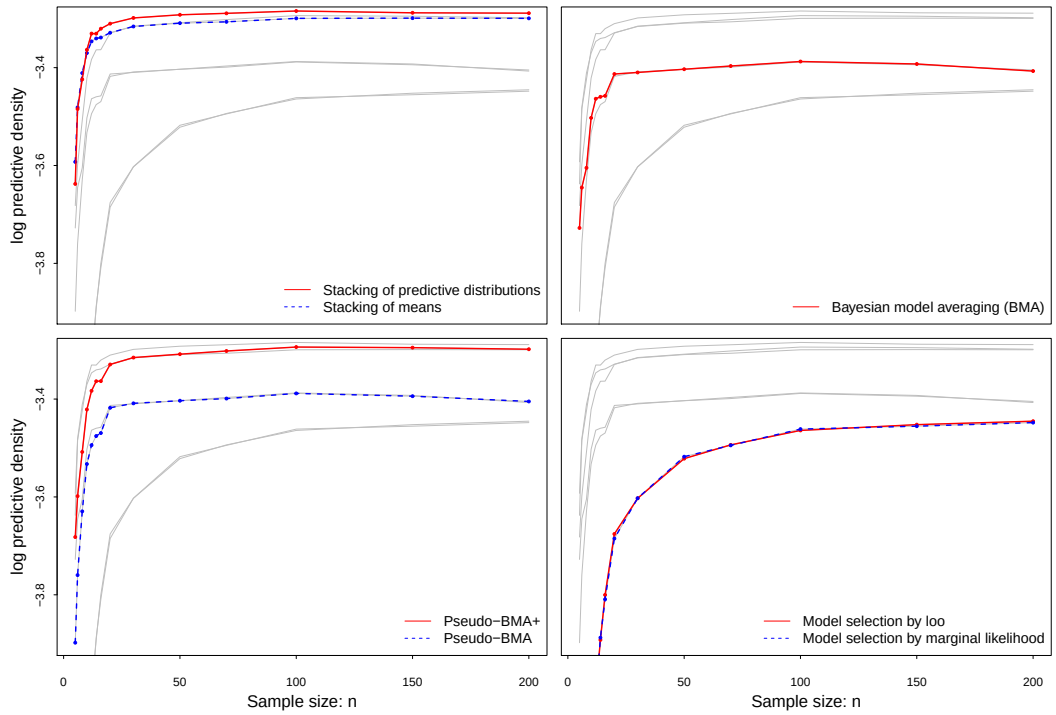


Figure 7.4: *Mean log predictive densities of 7 combination methods in the linear regression example: the k-th model is a univariate regression with the k-th variable ($1 \le k \le 15$). We evaluate the log predictive densities using 100 repeated experiments and 200 test data.*

selection, but that advantage vanishes in the limit since BMA picks up one model. In this $\mathcal{M}$-open setting, model selection can never be optimal.

The results change when we move to the second case, in which the k-th model contains variables $X_1, \ldots, X_k$ so that we are comparing models of differing dimensionality. The problem is $\mathcal{M}$-closed because the largest subset contains all the variables, and we have simulated data from this model. Figure 7.5 shows the mean log predictive densities of the seven combination methods in this case. For a large sample size n, almost all methods recover the true model (putting weight 1 on the full model), except BMA and model selection based on marginal likelihood. The poor performance of BMA comes from the parameter priors: recall that the optimality of BMA arises when averaging over the priors and not necessarily conditional on any particular chosen set of parameter values. There is no general rule to obtain a "correct" prior that accounts for the complexity for BMA in an arbitrary model space. Model selection by LOO can recover the true model, while selection by marginal likelihood cannot due to the same prior problems. Once again, BMA eventually becomes

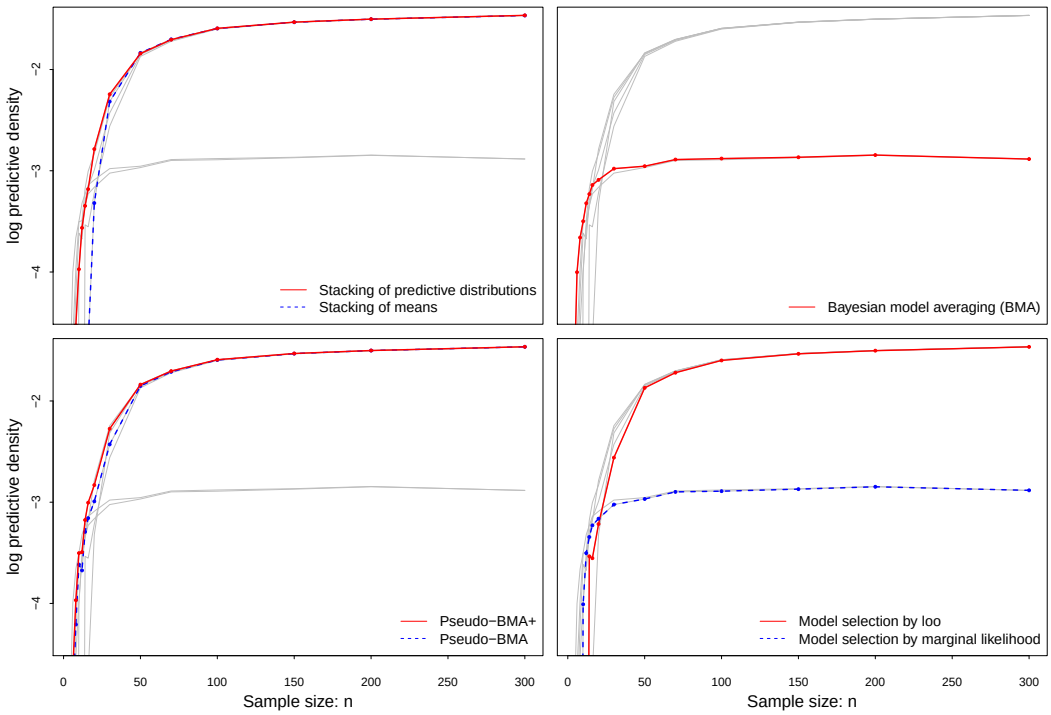


Figure 7.5: *Mean log predictive densities of 7 combination methods in the linear regression example: the k-th model is the regression with the first k variables ($1 \leq k \leq 15$). We evaluate the log predictive densities using 100 repeated experiments and 200 test data.*

the same as model selection by marginal likelihood, which is much worse than any other methods asymptotically.

In this example, stacking is unstable for extremely small n. In fact, our computations for stacking of predictive distributions and Pseudo-BMA depend on the PSIS approximation to $\log p(y_i|y_{-i})$. If this approximation is crude, then the second step optimization cannot be accurate. It is known that the parameter $\hat{k}$ in the generalized Pareto distribution can be used to diagnose the accuracy of PSIS approximation. When $\hat{k} > 0.7$ for a datapoint, we cannot trust the PSIS-LOO estimate and so we re-run the full inference scheme on the dataset with that particular point left out.

Comparison with mixture models. Stacking is inherently a two-step procedure. In contrast, when fitting a mixture model, one estimates the model weights and the status within parameters in the same step. In a mixture model, given a model list $\mathcal{M} = (M_1, \ldots, M_k)$, each component in the mixture occurs with probability w_k. Marginalizing out the discrete assignments yields the joint

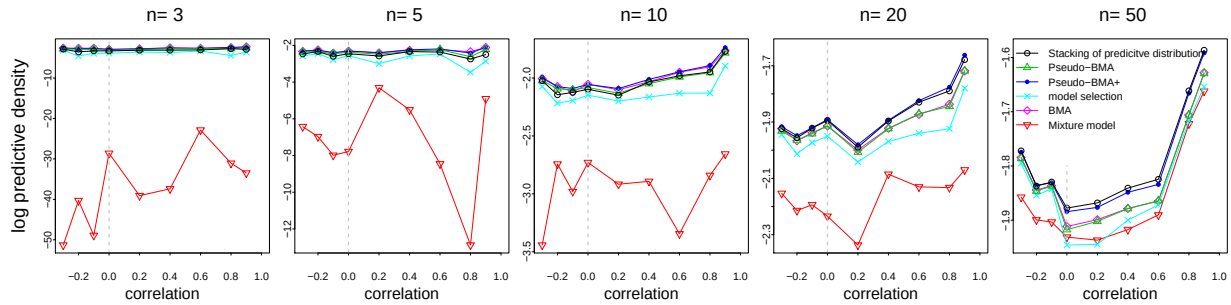


Figure 7.6: *Log predictive densities of the combined distribution obtained by stacking of predictive distributions, BMA, Pseudo-BMA, Pseudo-BMA+, model selection by marginal likelihood, and mixture models. In each case, we evaluate the predictive density by 100 testing data and 100 repeated simulations. The correlation of variables ranges from* -0.3 *to* 0.9*, and sample size ranges from 3 to 50. Stacking of predictive distributions and Pseudo-BMA+ outperform mixture models in all cases.*

likelihood

$$p(y|w_{1:K}, \theta_{1:K}) = \sum_{k=1}^{K} w_k p(y|\theta_k, M_k).$$

The mixture model seems to be the most straightforward continuous model expansion. Nevertheless, there are several reasons why we may prefer stacking to fitting a mixture model. Firstly, MCMC methods for mixture models are difficult to implement and generally quite expensive. Secondly, if the sample size is small or several components in the mixture could do the same thing, the mixture model can face non-identification or instability problem unless a strong prior is added.

Figure 7.6 shows a comparison of mixture models and other model averaging methods in a numerical experiment, in which the true model is

$$Y \sim \mathrm{N}(\beta_1 X_1 + \beta_2 X_2 + \beta_3 X_2, 1), \quad \beta_k \text{ is generated from } \mathrm{N}(0, 1),$$

and there are 3 candidate models, each containing one covariate:

$$M_k : Y \sim \mathrm{N}(\beta_k X_k, \sigma_k^2), \text{with a prior } \beta_k \sim \mathrm{N}(0, 1), \quad k = 1, 2, 3.$$

In the simulation, we generate the design matrix by $\mathrm{Var}(X_i) = 1$ and $\mathrm{Cor}(X_i, X_j) = \rho$. ρ determines how correlated these models are and it ranges from -0.3 to 0.9.

Figure 7.6 shows that both the performance of mixture models and single model selection are worse than any other model averaging methods we suggest, even though the mixture model takes much longer time to run (about 30 more times) than stacking or Pseudo-BMA+. When the sample size is small, the mixture model is too complex to fit. On the other hand, stacking of predictive distributions and Pseudo-BMA+ outperform all other methods with a moderate sample size.

Proximity and directional models of voting. Adams et al. (2004) use US Senate voting data from 1988 to 1992 to study voters' preference for the candidates who propose policies that are similar to their political beliefs. They introduce two similar variables that indicate the distance between voters and candidates. *Proximity voting comparison* represents the i-th voter's comparison between the candidates' ideological positions:

$$U_i(D) - U_i(R) = (x_R - x_i)^2 - (x_D - x_i)^2,$$

where x_i represents the i-th voter's preferred ideological position, and x_D and x_R represent the ideological positions of the Democratic and Republican candidates, respectively. In contrast, the i-th voter's *directional comparison* is defined by

$$U_i(D) - U_i(R) = (x_D - X_N)(x_i - X_N) - (x_R - X_N)(x_i - X_N),$$

where X_N is the neutral point of the ideology scale.

Finally, all these comparison is aggregated in the party level, leading to two party-level variable *Democratic proximity advantage* and *Democratic directional advantage*. The sample size is $n = 94$.

For both of these two variables, there are two ways to measure candidates' ideological positions x_D and x_R, which lead to two different datasets. In the *Mean candidate* dataset, they are calculated by taking the average of all respondents' answers in the relevant state and year. In the *Voter-specific* dataset, they are calculate by using respondents' own placements of the two candidates. In both datasets, there are 4 other party-level variables.

	Full model		***BMA***		***Stacking of predictive distributions***		***Pseudo-BMA+ weighting***	
	Mean Candidate	*Voter-specific*	*Mean Candidate*	*Voter-specific*	*Mean Candidate*	*Voter-specific*	*Mean Candidate*	*Voter-specific*
prox. adv.	-3.05 (1.32)	-2.01 (1.06)	-0.22 (0.95)	0.75 (0.68)	0.00 (0.00)	0.00 (0.00)	-0.02 (0.08)	0.04 (0.24)
direct. adv.	7.95 (2.85)	4.18 (1.36)	3.58 (2.02)	2.36 (0.84)	2.56 (2.32)	1.93 (1.16)	1.60 (4.91)	1.78 (1.22)
incumb. adv.	1.06 (1.20)	1.14 (1.19)	1.61 (1.24)	1.30 (1.24)	0.48 (1.70)	0.34 (0.89)	0.66 (1.13)	0.54 (1.03)
quality adv.	3.12 (1.24)	2.38 (1.22)	2.96 (1.25)	2.74 (1.22)	2.20 (1.71)	2.30 (1.52)	2.05 (2.86)	1.89 (1.61)
spend adv.	0.27 (0.04)	0.27 (0.04)	0.32 (0.04)	0.31 (0.04)	0.31 (0.07)	0.31 (0.03)	0.31 (0.04)	0.30 (0.04)
partisan adv.	0.06 (0.05)	0.06 (0.05)	0.08 (0.06)	0.07 (0.06)	0.01 (0.04)	0.00 (0.00)	0.03 (0.05)	0.03 (0.05)
constant	53.3 (1.2)	52.0 (0.8)	51.4 (1.0)	51.6 (0.8)	51.9 (1.1)	51.6 (0.7)	51.5 (1.2)	51.4 (0.8)

Figure 7.7: *Regression coefficients and standard errors in the voting example, from the full model (columns 1–2), the averaged subset regression model using BMA (columns 3–4), stacking of predictive distributions (columns 5–6) and Pseudo-BMA+ (columns 7–8).* Democratic proximity advantage *and* Democratic directional advantage *are two highly correlated variables.* Mean candidate *and* Voter-specific *are two datasets that provide different measurements on candidates' ideological placement.*

The two variables *Democratic proximity advantage* and *Democratic directional advantage* are highly correlated. Montgomery and Nyhan (2010) point out that Bayesian model averaging is an approach to helping arbitrate between competing predictors in a linear regression model. They average over all 2^6 linear subset models excluding those containing both variables *Democratic proximity advantage* and *Democratic directional advantage*, (i.e., 48 models in total). Each subset regression is with the form

$$M_\gamma : y|\left(X, \beta_0, \beta_\gamma\right) \sim N(\beta_0 + X_\gamma \beta_\gamma, \sigma^2).$$

Accounting for the different complexity, they used the hyper-g prior (Liang et al., 2008). Let ϕ to be the inverse of the variance $\phi = \frac{1}{\sigma^2}$. The hyper-$g$ prior with a hyper-parameter α is,

$$\begin{aligned} p(\phi) &\propto \frac{1}{\phi}, \\ \beta|\left(g, \phi, X\right) &\sim \mathrm{N}\Big(0, \frac{g}{\phi}(X^T X)^{-1}\Big), \\ p(g|\alpha) &= \frac{\alpha - 2}{2}(1+g)^{-\alpha/2},\ g > 0. \end{aligned}$$

The first two columns of Figure 7.7 show the linear regression coefficients as estimated using

least squares. The remaining columns show the posterior mean and standard error of the regression coefficients using BMA, stacking of predictive distributions, and Pseudo-BMA+ respectively. Under all three averaging strategies, the coefficient of *proximity advantage* is no longer statistically significantly negative, and the coefficient of *directional advantage* is shrunk. As fit to these data, stacking puts near-zero weights on all subset models containing *proximity advantage*, whereas Pseudo-BMA+ weighting always gives some weight to each model. In this example, averaging subset models by stacking or Pseudo-BMA+ weighting gives a way to deal with competing variables, which should be more reliable than BMA according to our previous argument.

Predicting well-switching behavior in Bangladesh. Many wells in Bangladesh and other South Asian countries are contaminated with natural arsenic. People whose wells have arsenic levels that exceed a certain threshold are encouraged to switch to nearby safe wells (for background details, see e.g., Yao et al., 2021a). We are analyzing a dataset including 3020 respondents to find factors predictive of the well switching. The outcome variable is

$$y_i = \begin{cases} 1, & \text{if household } i \text{ switched to a safe well.} \\ 0, & \text{if household } i \text{ continued using its own well.} \end{cases}$$

And we consider following input variables:

- `dist`: the distance (in meters) to the closest known safe well,
- `arsenic`: the arsenic level (in 100 micrograms per liter) of the respondent's well,
- `assoc`: whether a member of the household is active in any community association,
- `educ`: the education level of the head of the household.

We start with what we call Model 1, a simple logistic regression with all variables above as well as a constant term,

$$\begin{aligned} y \sim & \text{Bernoulli}(\theta), \\ \theta = & \text{logit}^{-1}(\beta_0 + \beta_1 dist + \beta_2 arsenic + \beta_3 assoc + \beta_4 educ). \end{aligned}$$

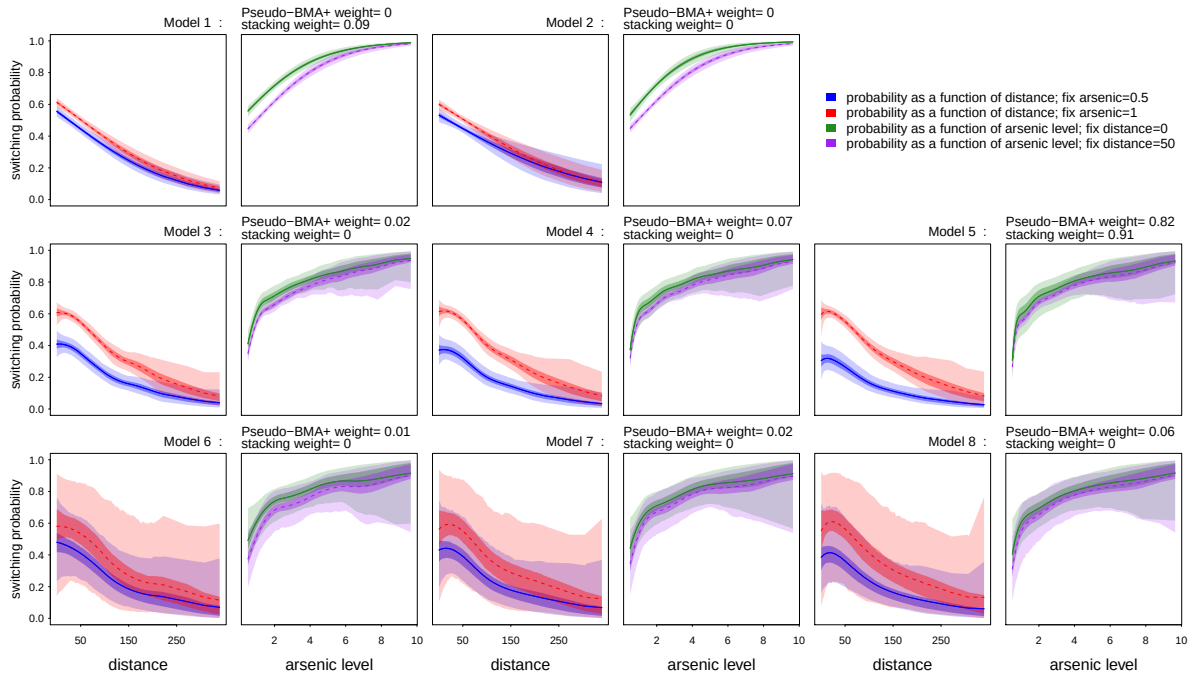


Figure 7.8: *The posterior mean, 50% and 95% confidence interval of the well switching probability in Models 1–8. For each model, the switching probability is shown as a function of (a) the distance to the nearest safe well or (b) the arsenic level of the existing well. In each subplot, other input variables are held constant. The model weights by stacking of predictive distributions and Pseudo-BMA+ are printed above each panel.*

Model 2 contains the interaction between distances and arsenic levels,

$$\theta = \text{logit}^{-1}(\beta_0 + \beta_1 dist + \beta_2 arsenic + \beta_3 assoc + \beta_4 educ + \beta_5 dist \times arsenic).$$

Furthermore, we can use spline to capture the nonlinear relational between the logit switching probability and the distance or the arsenic level. Model 3 contains the B-splines for the distance and the arsenic level with polynomial degree 2,

$$\theta = \text{logit}^{-1}(\beta_0 + \beta_1 dist + \beta_2 arsenic + \beta_3 assoc + \beta_4 educ + \alpha_{dis} B_{dis} + \alpha_{ars} B_{ars}),$$

where B_{dis} is the B-spline basis of distance with the form $\left(B_{dis,1}(dist), \ldots, B_{dis,q}(dist)\right)$ and $\alpha_{dis}, \alpha_{ars}$ are vectors. We also fix the number of spline knots to be 10. Model 4 and 5 are the similar models with 3-degree and 5-degree B-splines, respectively.

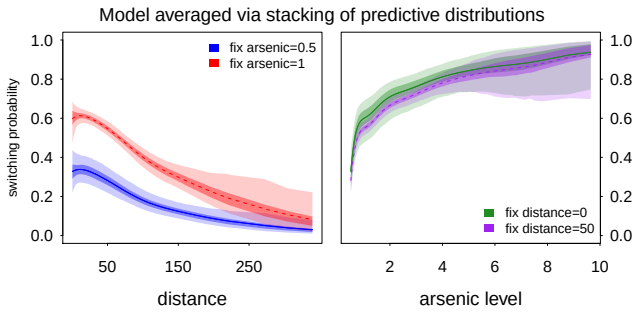


Figure 7.9: *The posterior mean, 50% and 95% confidence interval of the well switching probability in the combined model via stacking of predictive distributions. Pseudo-BMA+ weighting gives a similar result for the combination.*

Next, we can add a bivariate spline to capture nonlinear interactions,

$$\theta = \text{logit}^{-1}(\beta_0 + \beta_1 dist + \beta_2 arsenic + \beta_3 assoc + \beta_4 educ + \beta_5 dist \times arsenic + \alpha B_{dis,ars}),$$

where $B_{dis,ars}$ is the bivariate spline basis with the degree to be $2\times2, 3\times3$ and 5×5 in Model $6, 7$ and 8 respectively.

Figure 7.8 shows the inference results in all 8 models, which are summarized by the posterior mean, 50% confidence interval and 95% confidence interval of the probability of switching from an unsafe well as a function of the distance or the arsenic level. Any other variables `assoc` and `educ` are fixed at their means. It is not obvious from these results which one is the best model. Spline models give a more flexible shape, but also introduce more variance for posterior estimation.

Finally, we run stacking of predictive distributions and Pseudo-BMA+ to combine these 8 models. The calculated model weights are printed above each panel in Figure 7.8. For both combination methods, Model 5 (univariate splines with degree 5) accounts for the majority share. Model 8 is the most complicated one, but both stacking and Pseudo-BMA+ avoid overfitting by assigning it a negligible weight.

Figure 7.9 shows the posterior mean, 50% confidence interval, and 95% confidence interval of the switching probability in the stacking-combined model. Pseudo-BMA+ weighting gives a similar combination result for this example. At first glance, the combination looks quite similar to Model 5, while it may not seem necessary to put an extra 0.09 weight on Model 1 in stacking

combination since Model 1 is completely contained in Model 5 if setting $\alpha_{dis} = \alpha_{ars} = 0$. However, Model 5 is not perfect since it predicts that the posterior mean of switching probability will decrease as a function of the distance to the nearest safe well, for small distances. In fact, without further control, it is not surprising to find boundary fluctuation as a main drawback for higher order splines. This decreasing trend around the left boundary is flatter in the combined distribution since the combination contains part of straightforward logistic regression (in stacking weights) or lower order splines (in Pseudo-BMA+ weights). In this example the sample size $n = 3020$ is large, hence we have reasons to believe stacking of predictive distributions gives the optimal combination.

7.7 Discussion

Sparse structure and high dimensions. Yang and Dunson (2014) propose to combine multiple point forecasts, $f = \sum_{k=1}^{K} w_k f_k$, through using a Dirichlet aggregation prior, $w \sim \text{Dirichlet}\left(\frac{\alpha}{K^\gamma}, \ldots, \frac{\alpha}{K^\gamma}\right)$, and the adaptive regression. Their goal is to impose the sparsity structure (certain models can receive zero weights). They show their combination algorithm can achieve the minimax squared risk among all convex combinations,

$$\sup_{f_1,\ldots f_K \in F_0} \inf_{\hat{f}} \sup_{f_\lambda^* \in F_\Gamma} E||\hat{f} - f_\lambda^*||^2,$$

where $F_0 = (f : ||f||_\infty \leq 1)$.

The stacking method can also adapt to sparsity through stronger regularizations. When the dimension of model space is high, we can use a hierarchical prior on w in estimation (7.5) to pull toward sparsity if that is desired.

Constraints and regularity. In point estimation stacking, the simplex constraint is the most widely used regularization so as to overcome potential problems with multicollinearity. Clarke (2003) suggests relaxing the constraint to make it more flexible.

When combining distributions, there is no need to worry about multicollinearity except in degenerate cases. But in order to guarantee a meaningful posterior predictive density, the simplex

constraint becomes natural, which is satisfied automatically in BMA and Pseudo-BMA weighting. As mentioned in the previous section, stronger priors can be added.

Another assumption is that the separate posterior distributions are combined linearly. There could be gains from going beyond convex linear combinations. For instance, in the subset regression example when each individual model is a univariate regression, the true model distribution is a convolution instead of a mixture of each possible models distribution. Both of them lead to the additive model in the point estimation, so stacking of the means is always valid, while stacking of predictive distributions is not possible to recover the true model in the convolution case.

Our explanation is that when the model list is large, the convex span should be large enough to approximate the true model. And this is the reason why we prefer adding stronger priors to make the estimation of weights stable in high dimensions.

General recommendations. The methods discussed in this paper are all based on the idea of fitting models separately and then combining the estimated predictive distributions. This approach is limited in that it does not pool information between the different model fits: as such, it is only ideal when the K different models being fit have nothing in common. But in that case we would prefer to fit a larger super-model that includes the separate models as special cases, perhaps using an informative prior distribution to ensure stability in inferences.

That said, in practice it is common for different sorts of models to be set up without any easy way to combine them, and in such cases it is necessary from a Bayesian perspective to somehow aggregate their predictive distributions. The often-recommended approach of Bayesian model averaging can fail catastrophically in that the required Bayes factors can depend entirely on arbitrary specifications of noninformative prior distributions. Stacking is a more promising general method in that it is directly focused on performance of the combined predictive distribution. Based on our theory, simulations, and examples, we recommend stacking (of predictive distributions) for the task of combining separately-fit Bayesian posterior predictive distributions. As an alternative, Pseudo-BMA+ is computationally cheaper and can serve as an initial guess for stacking. The

computations can be done in R and Stan, and the optimization required to compute the weights connects directly to the predictive task.

Looking forward. Along with an increasing number of statistical models and learning algorithms, ensemble methods have been appealing tools to expand existing models and inferential procedures, and to improve predictive performance. In addition, the popularity of ensemble methods in Bayesian statistics can be viewed as representing a modern shift in Bayesian data analysis: from a static model-based inference to a Bayesian workflow in which we are fitting many models while working on a single problem.

This chapter is mostly about Bayesian model averaging, stacking, and their variants. For these methods, the model weights are trained after model-specific inferences, and the cost of the former is typically much smaller than the latter. Another popular approach to construct ensembles is to train each model and the model weight simultaneously or iteratively, such as in boosting (Freund and Schapire, 1997), gradient boosting (Friedman, 2001), and mixture of experts (Jacobs et al., 1991). These methods are computationally intensive for full-Bayesian inference, but more useful to combine weak learners. On the other hand, different ensemble methods can be further aggregated: for example, to stack fits from BMA and mixture of experts.

Many of these ensemble methods had limited usage until enough computational resources and efficient approximation became available. Conversely, many model averaging strategies also help solve difficulties in statistical computing. For example, bagging stabilizes otherwise non-robust point estimates, and stacking can be used in multimodal posterior sampling.

Looking forward, there are many open questions. To name a few, both BMA and stacking are restricted to a linear mixture form, would it be beneficial to consider other aggregation forms such as convolution of predictions or a geometric bridge of predictive densities? Stacking often relies on some cross-validation, how can we better account for the finite sample variance therein? While staking can be equipped with many other scoring rules, what is the impact of the scoring rule choice on the convergence rate and robustness? Beyond current model aggregation tools, can

we develop an automated ensemble learner that could fully explore and expand the space of model classes—for example, using an autoregressive (AR) model and a moving-average (MA) model to learn an ARMA model? We leave these directions for future investigation.

7.8 Software implementation

An `R` package `loo` (Vehtari et al., 2019a) provides model weights from the PSIS-LOO based stacking and pseudo-BMA. Suppose `fit1`, `fit2` and `fit3` are three models fit objects from the Bayesian inference package Stan (Stan Development Team, 2020), then we can compute their stacking weights as follows.

```
model_list <- list(fit1, fit2, fit3)
log_lik_list <- lapply(model_list, extract_log_lik)
# stacking:
wts <- loo_model_weights( log_lik_list, method = "stacking",
optim_control = list(reltol=1e-10))
```

Chapter 8. Using stacking to combine non-mixing Bayesian computations: The curse and blessing of multimodal posteriors [1]

"They are immortal, all those stars both silvery and golden shall shine out again,
The great stars and the little ones shall shine out again, they endure,
The vast immortal suns and the long-enduring pensive moons shall again shine."

—Walt Whitman

8.1 The curse of multimodal posteriors

Bayesian computation becomes difficult when posterior distributions are multimodal or, more generally, meta-stable, as then it is generally impossible to compute moments analytically or to directly draw simulations. Variational and mode-based approximations can be poor fits to the posterior, and it can be difficult to identify all the modes in the first place. General-purpose Markov chain Monte Carlo algorithms can have problems moving between modes. And even if different modes are found, it is difficult to compute their relative weights in the posterior distribution, as this requires integration over the posterior density within each mode.

Should we just run longer and longer chains? This is inefficient when effective sample size per iteration (Vehtari et al., 2020a) is low. The state-of-the-art Hamiltonian Monte Carlo sampler for a bimodal density mixes as poorly as random-walk Metropolis (Mangoubi et al., 2018), and even optimal tuning and Riemannian metrics do not help. When the posterior distribution has unconnected masses, the Markov chain is not irreducible and will never converge to its target.

In some cases it is possible to collapse multimodality using reparameterization (Papaspiliopoulos et al., 2007; Johnson and Geyer, 2012; Betancourt and Girolami, 2015; Gorinova et al., 2020), but this is not automated for general problems. Several schemes have been proposed for sampling from distributions with isolated modes by increasing the temperature to enhance the transition

[1] This chapter is a slight modified version of Yao et al. (2020b).

probability between modes; these methods include annealing (Kirkpatrick et al., 1983; Bertsimas and Tsitsiklis, 1993), parallel tempering (Hansmann, 1997; Earl and Deem, 2005), simulated tempering (Marinari and Parisi, 1992; Neal, 1993), Wang-Landau algorithm (Wang and Landau, 2001), and path sampling (Gelman and Meng, 1998; Yao et al., 2020a). These merthods enlarge the sampling space by an auxiliary tempeture and often require to estimate the partition function: a high dimensional integral. They are useful for many statistical physics and molecular biology problems, in which the only goal is to sample from a given density. However, tempering-based methods are sensitive to implementation and tuning, and their theoretical mixing rates drop quickly in high dimensions (Bhatnagar and Randall, 2004). Based on simulations in Yao et al. (2020a), methods based on simulated tempering are unlikely to work in large statistical models with the scale that we consider in the present chapter.

Moreover, the metastability of sampling (Figure 8.1) comes from both the energetic (two modes are distinct) and entropic (two regions are connected through a narrow neck) barriers. Increasing the temperature does not ease the entropic barrier, which is a common problem with hierarchical models.

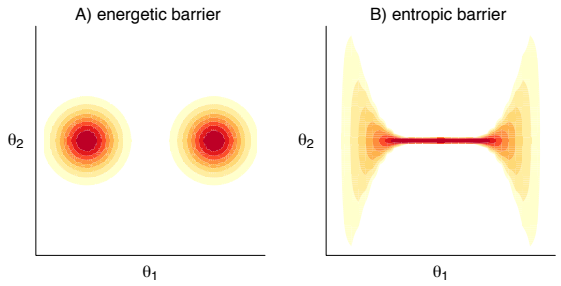


Figure 8.1: *Both the energetic and entropic barriers lead to sampling metastability, and the entropic barrier cannot be eliminated by increasing the temperature.*

Despite these computational difficulties, we would like to aim for full Bayesian inference, or some approximation, because of the general benefits of using probability to quantify uncertainty in inferences.

One way to explore a multimodal space is to run a large number of chains of MCMC or variational inference from dispersed starting points, but then the question arises of how to average over resulting inferences if they have not mixed. There can be benefits to averaging non-mixing MCMC chains even with uniform weighting (Hoffman and Ma, 2020), but it should be possible do better: equal weighting is convenient but is not in general appropriate and can lead to a strong dependence on initial values.

Stacking (Wolpert, 1992; Breiman, 1996b; LeBlanc and Tibshirani, 1996; Clarke, 2003) and its Bayesian variants (Clyde and Iversen, 2013; Le and Clarke, 2017; Yao et al., 2018a) use cross validation to average over a discrete set of fitted models.

The present chapter extends stacking to combine multiple chains fitting the same model. Although this might seem like a small step, the practical implications of this idea are large, because multimodality, meta-stability, and poor mixing are common problems when fitting complex multilevel models.

Applying Bayesian stacking for non-mixing computations involves two challenges. The computational challenge is to approximate the results of cross validation without the need to repeatedly re-fit the model. This can be done using Pareto-smoothed importance sampling (Vehtari et al., 2017, 2019b). The conceptual challenge is that the goal of minimizing prediction error is not the same as minimizing Kullback-Leibler divergence to or approximating a posterior distribution. We can prove some theoretical results regarding the benefits of cross validation in this setting, but ultimately the effectiveness of stacking is best demonstrated by applying it to a series of challenging problems that represent different sorts of posterior distributions that arise in applied statistics.

The contribution of this chapter is to provide a practical solution to yield a combined inference from non-mixing computations, and to evaluate it on a diverse set of statistical examples. Our procedure is scalable with negligible postprocessing cost and constructed to minimize cross-validated prediction error.

In Section 8.2 we discuss various types of posterior multimodality, some of which can arise from model misspecification. Section 8.4 details our method and practical implementation to deal with non-mixing chains for Bayesian computation. We use a theoretical example in Section 8.7 to show that stacked-chain inference can achieve higher predictive efficiency than exact posterior density asymptotically. In Section 8.8 we demonstrate the proposed method in applied examples, including hyperparameter bimodality and slow mixing in Gaussian process regression, latent Dirichlet allocation, unstable variational inference in horseshoe regression, and Bayesian neural networks.

8.2 The folk theorem of statistical computing

When you have computational problems, often there's a problem with your model. This "folk theorem" (Gelman, 2008) can be understood by thinking of a statistical model or family of distributions as a set of possible probabilistic explanations for a dataset. If the data come from some distribution in the model class, then with identification and reasonable sample size we can expect to distinguish among these explanations, and with a small sample size and continuous model, we would hope to find a continuous range of plausible explanations and thus a well behaved posterior distribution. But if the data do not fit the model, so that none of the candidate explanations work, then the posterior distribution represents a mixture of the best of bad choices, and it can have poor geometry in the same way that the seafloor can look rough if the ocean is drained.

Poor data fit, or conflict between the prior and likelihood, do not necessarily lead to awkward computation. For example, the normal-normal model yields a log-concave posterior density with constant curvature for any data. But if a model is flexible enough to fit different qualitative explanations of data, then poorly fitting data can be interpreted by the model as ambiguity, as indicated by posterior multimodality.

The other way a model can be difficult to fit is if its parameters are only weakly constrained by the posterior. With a small sample size (or, in a hierarchical model, a small number of groups), uncertainty in the hyperparameters can yield a posterior distribution of widely varying curvature, which leads to slowly mixing MCMC. In practice, we can often fix the geometry by putting stronger priors on these hyperparameters. However, a strong prior constraint is not always desired—sometimes we are interested in fitting a model that is legitimately difficult to compute, because we want to allow for different possible explanations of the data, and a too strong prior implies an adhoc selection. These are settings where the stacking approach discussed in this chapter can be useful.

Under the correct model and reasonable priors, Bayesian posteriors often attain asymptotic normality and leave little room for several distinct and non-vanishing modes. That ensures rapid

mixing for random-walk Metropolis, scaling as $O(d)$ (Roberts et al., 1997; Cotter et al., 2013; Dwivedi et al., 2018), and Hamilton Monte Carlo, scaling as $O(d^{1/4})$ (Beskos et al., 2013; Bou-Rabee et al., 2020; Mangoubi and Smith, 2017, 2019). From this perspective, multimodal posteriors should be unlikely with a large enough sample size. With smaller sample sizes, lack of convergence to the asymptotic normality can arise from various sources.

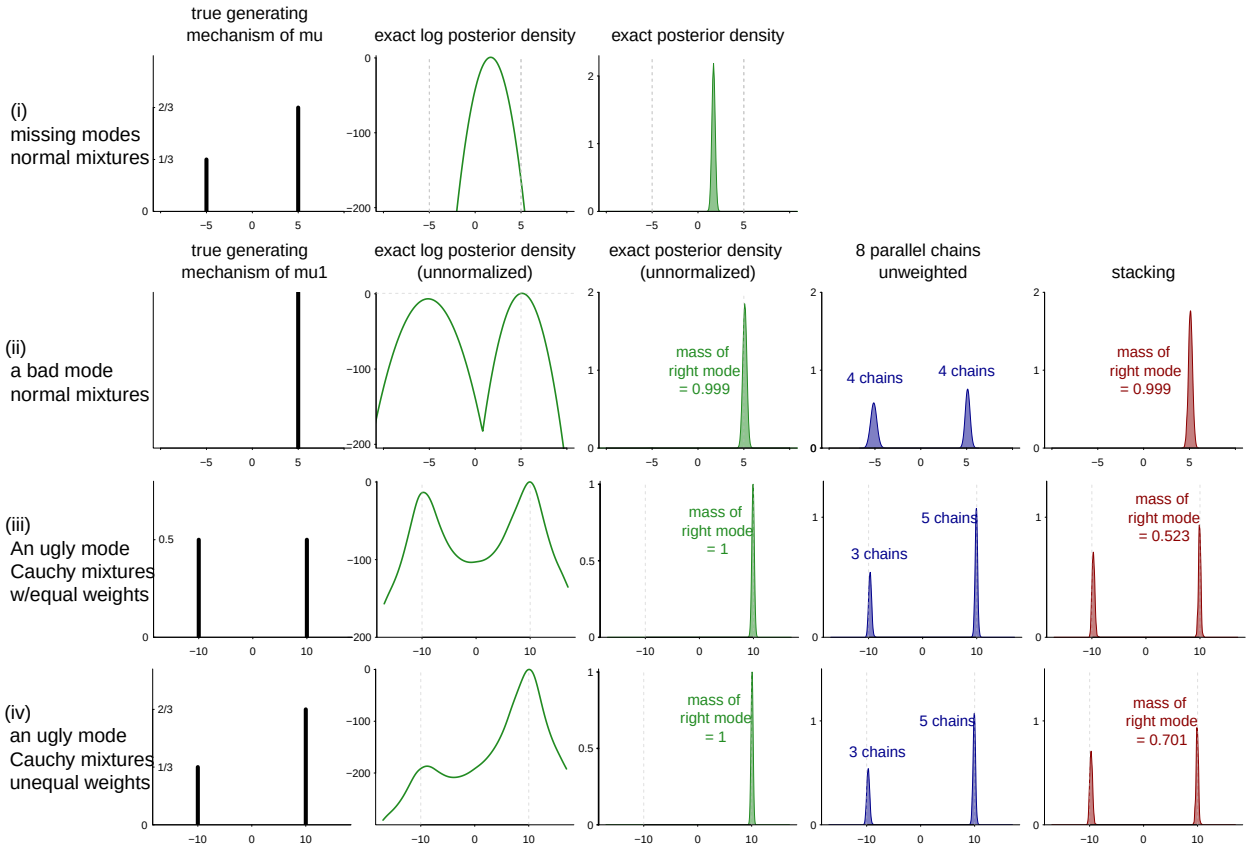
8.3 The general problem of Bayes limiting behavior under model misspecification

Mixture model examples. Before theory discussions, we first design four mixture examples, visualized in Figure 8.2. They demonstrate the behavior of exact inference, uniformly weighted parallel chains, and stacking weighted parallel chains in case of different types posteriors.

(i) A missing mode: We draw n points $y_i, \ldots, y_n$ independently from the mixture, $\frac{2}{3}\text{normal}(5,1) + \frac{1}{3}\text{normal}(-5,1)$. We fit the model $y_i|\mu \sim \text{iidnormal}(\mu,1)$ with a flat prior on μ. The true data generating process (DG) is expressed by $\mu \sim \frac{2}{3}\delta(5) + \frac{1}{3}\delta(-5)$, but the Bayesian posterior $p(\mu|y) = \text{normal}(\bar{y}, 1/\sqrt{n})$ is unimodally concentrated at $\mu = \bar{y} \approx 5/3$ and cannot catch the two modes in data.

(ii) A bad mode: With the same data y above, now we fit a two-component normal model $y \sim \frac{2}{3}\text{normal}(\mu_1,1) + \frac{1}{3}\text{normal}(\mu_2,1)$ with known mixture probability and a flat prior on μ_1, μ_2. The model is identifiable, but the resulting posterior is bimodal, centered around $(\mu_1,\mu_2) = (5,-5)$ and $(-5,5)$ respectively. Asymptotically ($n \to \infty$) the posterior converges to the first mode, thereby the data generating process, but the existence of a second artifact mode both challenges the sampling and compromises the prediction with finite data sample size. In Figure 8.2 we simulate $n = 30$ data points and run eight parallel chains. Four chains are trapped in the "wrong" mode.

(iii) An ugly mode: We generate data $y_1, \ldots, y_n$ iid from $\frac{1}{2}\text{Cauchy}(10,1) + \frac{1}{2}\text{Cauchy}(-10,1)$. We fit a one-component model $y \sim \text{Cauchy}(\mu,1)$ with a flat prior. The true data generating process is expressed by $\mu \sim \frac{1}{2}\delta(10) + \frac{1}{2}\delta(-10)$. In the limit ($n \to \infty$), the posterior density will be



Summary: example	# modes in DG	# modes in posterior	posterior → DG as $n \to \infty$?	unweighted chains approx. DG?	stacking approx. DG?
i	2	1	✗	✗	✗
ii	1	2	✓	✗	✓
iii	2	2	✗	✗	✓
iv	2	2	✗	✗	✓

Figure 8.2: *Under a multimodal data generating mechanism, the exact Bayesian posterior can miss the modes in (i) or over-concentrate at one mode (iii–iv). Stacking, our proposed method, approximates the data generating process well in (ii–iv). The sample size* $n = 30$ *in (i–ii) and* $n = 100$ *in (iii–iv).*

concentrated at one of two points $\mu \approx \pm 9.8$. In the simulation with $n = 100$, the right-side posterior mode contains almost 100% mass (up to the precision 10^{-6}). The induced predictive model then only describes half of data. *Stacking*, as implemented in this chapter, assigns a weight of 0.52 to the right-side mode, achieving a much better prediction compared to the data generating process.

(iv) Another ugly mode: We draw n data points $y_i, \ldots, y_n$ independently from the mixture model, $\frac{2}{3}\text{Cauchy}(10, 1) + \frac{1}{3}\text{Cauchy}(-10, 1)$ and again fit a one-component model $y \sim \text{Cauchy}(\mu, 1)$ with

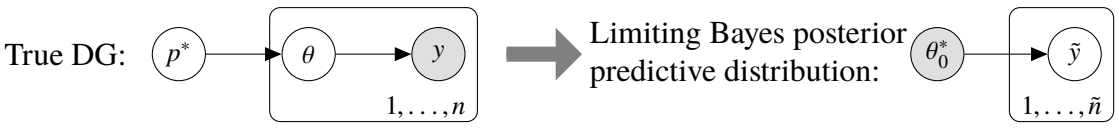


Figure 8.3: *When the parameter θ is randomly drawn from a distribution p^* in the data generating process* (8.3)*, the limiting posterior inference $p(\theta|y)$ almost surely converges to a point estimate θ_0^*.*

a flat prior. The posterior $p(\theta|y)$ carries almost all masses on the right-side mode $p(\theta|y) \approx \delta(\theta - 10)$, while our proposed method still approximates the true data generating process.

These examples represent various sources of posterior multimodality. In example (ii), one of the modes is purely an artifact. Not drawing a posterior sample around it improves finite sample-predictions. Such artifact-type modes are found in cases of prior-data conflict, label-switching, aliasing (Bafumi et al., 2005), mixture and cluster-based models (Stephens, 2000; Blei et al., 2003), and hierarchical models (Liu and Hodges, 2003).

In other examples, the data generating process (DG) can be expressed via a bimodal distribution on μ. In example (i), the Bayesian posterior $p(\mu|y)$ converges to some middle point. In (iii–iv), the posterior overconfidently concentrates at one of the modes and ignores the other, even when these two modes have equal density at the modal point. We will revisit the Cauchy mixture in Section 8.7 and prove its limiting behavior, in which Bayesian inference almost surely occasions overconfident concentration, but the blessing of bimodality enables stacking to recover the *true* data generating process from the *wrong* model and *wrong* inference.

The curse of a multiverse generating process. Given data $y_1, \ldots, y_n$ generated independently and identically distributed from an unknown data generating process: $p_{\text{true}}(y)$, and a potentially misspecified model $y|\theta \sim f(y|\theta)$ and prior $p(\theta), \theta \in \Theta$, when the sample size n goes to infinity and regularization conditions apply, the limiting Bayesian posterior will be almost surely supported on the set of global modes (Berk, 1966; Kleijn and Van der Vaart, 2012): $\mathcal{A} = \left\{\theta^* \in \Theta : \mathbb{E}_{\tilde{y}\sim p_{\text{true}}} \log f(\tilde{y}|\theta^*) = \max_{\theta\in\Theta} \mathbb{E}_{\tilde{y}\sim p_{\text{true}}} \log f(\tilde{y}|\theta)\right\}$.

Such limiting behavior has two undesired properties. First, when data are generated from one parameter θ_0 in the model (an $\mathcal{M}$-closed view), $p_{\text{true}} = f(\cdot|\theta_0)$, the posterior will be asymptotically

concentrated at θ_0. But otherwise, the limiting predictive distribution should be interpreted as the closest distribution to data generating process in terms of Kullback–Leibler (KL) divergence, as we can rewrite $\mathcal{A}$ as

$$\mathcal{A} = \arg\min_{\theta\in\Theta} \mathrm{KL}\Big(p_{\text{true}}(\cdot) \mid\mid f(\cdot|\theta)\Big), \quad \forall\eta > 0,\ \Pr_{\text{bayes}}(||\theta - \mathcal{A}|| < \eta \mid y_{1\ldots,n}) \xrightarrow{n\to\infty,\ a.s.} 1, \quad (8.1)$$

The asymptotic predictive distribution is equivalent to some *point* estimate $f(\cdot|\theta^*), \theta^* \in \mathcal{A}$. What makes a method Bayesian is the use of probability to quantify uncertainties. Ideally we would fully use the expressibility of the model and find the optimal *probabilistic* inference $p_{\text{optimal}}(\theta)$ from some space $\mathcal{F}$ that renders the best prediction for future unseen data according to a user-specified divergence $D(\cdot||\cdot)$,

$$p_{\text{optimal}} = \arg\min_{\tilde{p}\in\mathcal{F}} \mathrm{D}\Big(p_{\text{true}}(\cdot) \mid\mid \int_\Theta f(\cdot|\theta)\tilde{p}(\theta)d\theta\Big). \quad (8.2)$$

In particular, if the model is expressive enough (see Figure 8.3), then there is a density $p^*(\cdot) \in \mathcal{F}$ such that

$$p_{\text{true}}(\tilde{y}) = \int_\Theta f(\tilde{y}|\theta)p^*(\theta)d\theta. \quad (8.3)$$

and we can choose any divergence D corresponding to a strictly proper scoring rule (Gneiting and Raftery, 2007). Then the minimum of (8.2) is attained at p^*, hence the "correct inference."

The limiting Bayesian posterior (8.1) is a special solution to (8.2) where we fix D to be KL divergence and restrict the distribution family $\mathcal{F}$ to be formed from Dirac delta functions: $\mathcal{F} = \{\delta(\theta_0) \mid \theta_0 \in \Theta\}$.

To rephrase the folk theorem, challenges in sampling and difficulties in modeling are confounded. There is no general algorithm to sample from truly multimodal distributions, and even if this could be done, the posterior multimodality can signify that the true data are unlikely generated to have been from any single parameter in the model, and so the Bayesian posterior itself, which over-concentrates in the limit, is not appropriate.

With enough data, model misspecifications can be detected using posterior predictive checks (Gelman et al., 1996), and (8.3) can be expanded to a hierarchical model by replacing θ by n copies

$\theta_{1,\ldots,n}$, with hyperpriors $\theta_i|\tau \sim p(\theta_i|\tau)$:

$$y_i|\theta_i = f(y_i|\theta_i), \quad \theta_i|\tau \sim p(\theta_i|\tau), \quad \tau \sim p(\tau), \quad i = 1,\ldots,n. \tag{8.4}$$

But it enlarges the model n times bigger, hurting both computational scalability and finite-sample convergence rate. Meanwhile, the hyperprior $p(\theta_i|\tau)$ can still be misspecified.

8.4 Inference from non-mixed computation: parallel approximation and stacking

To start, we assume we have some existing computer program that attempts to draw samples from a posterior distribution $p(\theta|y)$ but might get trapped in a single mode or, more generally, a small part of the distribution. We will typically be using Hamiltonian Monte Carlo with the no-U-turn sampler (HMC/NUTS) in Stan (Stan Development Team, 2020). For the present chapter, all that is necessary is that the algorithm produces *some* set of posterior draws, which might also be obtained for example by variational inference (Blei et al., 2017) or mode-based approximation such as Laplace's method or expectation propagation (Vehtari et al., 2020b).

Step 1: Parallel evaluation. We run our program M times from different starting points to have a chance to explore many modes or areas of the target distribution. We also recommend an overdispersed initialization. Multiple starting points is not a new idea in statistical computation, but we emphasize that our goal here is *exploration*, without the expectation that the chains will mix with each other, nor that all modes and separated regions are reached. It could, for example, make sense to run the simulation algorithm in parallel on a large number of processors in a cluster. If the algorithm is iterative, follow the usual protocol to discard the initial transient states; for example when running MCMC we typically discard, as warmup, the first half of each simulated chain.

In practice, within-chain convergence is easier than mixing among parallel chains. This is especially true for distribution with isolated modes and high between-mode energy barriers. To monitor the within-chain convergence, we use split-$\widehat{R}$ (Vehtari et al., 2020a). For most simulation we experimented, it is fairly easy to have split-$\widehat{R} \approx 1$ for most chains.

Step 2 (optional): Clustering. We can use a between-chain mixing measure such as $\widehat{R}$ (Gelman and Rubin, 1992; Vehtari et al., 2020a) to partition the M parallel simulations into K clusters, each of which approximately captures the same part of the target distribution. Label the simulations from cluster k as $(\theta_{ki}, i = 1, \ldots, S_k)$, with the total number of draws being $S = \sum_{k=1}^{K} S_k$. This step is optional and recommended if the number of parallel runs M is large.

To keep notation coherent, when the clustering step is skipped, we denote $K = M$ and θ_{ks} as the s-th sample in the k-th chain. Throughout the chapter, we use $1 \leq i \leq n$ to index outcome observations, $1 \leq k \leq K$ to index clusters (chains, optimization runs), and $1 \leq s \leq S$ to index posterior draws.

Step 3: Reweighing non-mixing chains using stacking. From the previous two steps, we assume θ_{ki} come from a stationary distribution $p_k(\theta|y)$, which in general do not mix, nor do they match the exact posterior $p(\theta|y)$.

To take into account between-chain heterogeneity, we consider a generalized form of Monte Carlo estimate for any integral function $h(\theta)$ from chain-wise weights $w_1, w_2, \ldots, w_K$:

$$\mathbb{E}[h(\theta)] \approx \sum_{k=1}^{K} \sum_{s=1}^{S_k} w_k S_k^{-1} h(\theta_{ks}). \tag{8.5}$$

The usual Monte Carlo estimate is a special case with $w_1 = \cdots = w_K = 1/K$.

We optimize weights in (8.5) to maximize the leave-one-out cross validation performance of the distribution formed from the weighted average of the simulation draws. This first requires estimation of the pointwise leave-one-out (loo) log predictive density (lpd, Gelman et al., 2014; Vehtari et al., 2017) from the k-th cluster (chain):

$$\log p_k(y_i|y_{-i}) = \log \int_{\theta \in \Theta} p(y_i|\theta) p_k(\theta|y_{1,\ldots,i-1,i+1,\ldots,n}) d\theta, \quad i = 1, \ldots, n,\ k = 1, \ldots, K. \tag{8.6}$$

Second, we solve

$$w^*_{1,\dots,K} = \arg\max_{w\in\mathbb{S}(K)} \sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k p_k(y_i|y_{-i}) + \log p_{\text{prior}}(w), \tag{8.7}$$

where $\mathbb{S}(K)$ is the space of K-dimensional simplex,

$$\mathbb{S}(K) = \{w : 0 \le w_k \le 1, \forall 1 \le k \le K;\ \sum_{k=1}^{K} w_k = 1\},$$

and $p_{\text{prior}}(w)$ is prior regularization.

In Section 8.5, we further approximate all $\log p_k(y_i|y_{-i})$ terms by importance sampling—it suffices to fit all the full data once in each chain. We will also discuss the choice of priors $p_{\text{prior}}(w)$.

We view this optimization as a finite-sample estimate in (8.2), where we construct the distribution family as a mixture from sampled posterior clusters, $\mathcal{F} = \left\{\sum_{k=1}^{K} w_k p_k(\theta|y) : w \in \mathbb{S}(K)\right\}$.

Finally, plugging the stacking wights $w^*_1, \dots, w^*_K$ into (8.5), we obtain the chain-weighted Monte Carlo estimates. The resulting approximation of the target distribution uses $\sum_{k=1}^{K} S_k$ draws, with each θ_{ks} having weight w^*_k/S_k.

Notably, we are by default using the logarithmic scoring rules in predictive evaluation, for it is strictly proper. It is straightforward to adopt a user-specified prediction utility by replacing the log predictive densities in the optimization step (8.7).

Step 4: Monitoring convergence. After K parallel runs, we cannot exclude the possibility that another local mode or separated posterior region has been overlooked. We could use capture-recapture methods to estimate the number of unseen modes. When there is a discrete combinatorial explosion, it is essentially impossible to capture the full support of the distribution. So we are implicitly assuming that we have a rough sense of the support of most of the posterior mass, or, conversely, that we were previously willing to approximate the target distribution using a single mode, in which case we would hope a multimodal average to be an improvement.

On the other hand, there is no need to capture all modes that are predictively identical. We

monitor the weighted log predictive density as a function of how many components are added in stacking. Ideally we should test it over an independent hold-out test data set, and stop when the log predictive density of the stacked posterior reaches the maximum. Alternatively we can use cross validation. For each $K' \leq K$, obtain stacking weights $w_k^{K'}$ from chain $1, \ldots, K'$, and monitor the their stacked lpd as a function of number of chains K', which typically monotonically increases:

$$\text{lpd}_{\text{loo}}(K') = \sum_{i=1}^{n} \log \sum_{k=1}^{K'} w_k^{K'} p_k(y_i|y_{-i}),\ 1 \leq K' \leq K. \tag{8.8}$$

We terminate if $\text{lpd}_{\text{loo}}(K')$ becomes relatively stable. Otherwise we sample extra chains and repeat steps 1–4 on all chains.

8.5 Practical implementation

Leave-one-out posterior distributions. Let $p_k(\theta|y)$ to be the stationary distribution from which the k-th cluster (chain) is drawn. Working with the exact leave-one-out distributions $p_k(\theta|y_{-i})$ in (8.6) is not only computationally intensive (requiring the model to be fit n times) but also conceptually ambiguous: Using full data and given initialization, the sampler obtains $\theta_{k1}, \ldots, \theta_{kS_k}$ from the k-th region, but what if after y_i is removed from the same initialization reaches another mode, or what if there is a phase transition and there are no longer K clusters?

The usual leave-one-out model can be written as, $p(\theta|y_{-i}) \propto p(\theta|y_{-i})p(\theta) = p(\theta|y)p(\theta)/p(y_i|\theta) = p(\theta|y)/p(y_i|\theta)$. We avoid the ambiguity by defining $p_k(\theta|y_{-i})$ to be

$$p_k(\theta|y_{-i}) \coloneqq \frac{p_k(\theta|y)/p(y_i|\theta)}{\int_{\theta\in\Theta} p_k(\theta|y)/p(y_i|\theta)}. \tag{8.9}$$

Efficient approximation of leave-one-out distributions. We use Pareto smoothed importance sampling (PSIS, Vehtari et al., 2017, 2019b) to compute (8.9). It suffices to only fit the full model once per chain. For each chain k, we obtain the raw leave-one-out importance ratios $1/p(y_i|\theta_{ks})$, $i = 1, \ldots, n$ and stabilize these by replacing the largest ratios by the expected order statistics in a fitted generalized Pareto distribution and followed by right truncation. Labeling the

Pareto-smoothed importance ratio as r_{iks}, we approximate $p_k(y_i|y_{-i})$ by

$$p_k(y_i|y_{-i}) \approx \frac{\sum_{s=1}^{S_k} p_k(y_i|\theta_{ks}) r_{iks}}{\sum_{s=1}^{S_k} r_{iks}},\ k = 1, \ldots, K,\ i = 1, \ldots, n. \tag{8.10}$$

This is asymptotically ($S_k \to \infty$) unbiased and consistent. The finite-sample reliability and convergence rate can be assessed using the estimated shape parameter $\hat{k}$ of the fitted generalized Pareto distribution. We refer to Vehtari et al. (2017, 2019a) and Appendix B of this chapter for detailed algorithms and software implementation.

In summary, after parallel sampling, the extra computation costs of stacking only involve summations in (8.10) and a length-K-vector optimization in (8.7), which are negligible compared with the cost of sampling.

Prior on stacking weights. Extra priors beyond a simplex constraint in model averaging have been considered (Le and Clarke, 2017; Yao et al., 2018a) but seldom applied in practice. Under a flat prior $p_{\text{prior}}(w) = 1$, the optimum in (8.7) is nonidentified and numerically unstable if two simplexes $w' \neq w''$ entail the identical prediction $\sum_k w'_k p_k(\cdot|y) = \sum_k w''_k p_k(\cdot|y)$. We need an informative prior for the *predictive power* versus *Monte Carlo error* tradeoff.

If all chains are distributed identically, and within chain sampling is independent, the variance of (8.5) will be $\text{Var}\left(\sum_{k=1}^{K}\sum_{s=1}^{S_k} w_k {S_k}^{-1} h(\theta_{ks})\right) = \sum_{k=1}^{K} {w_k}^2 {S_k}^{-1} \text{Var}\left(h(\theta)\right)$, whose minimum is attained at $w_k = S_k / \sum_{k'} S_{k'}$. This justifies the uniform weights $1/K$ in the usual multi-chain Monte Carlo scheme where, after complete mixing, any weighting yields unbiased estimates.

Further, when the k-th chain has an effective sample size $S_{\text{eff},k}$ (Vehtari et al., 2020a), we approximate the variance of the Monte Carlo estimate (8.5) to be $\text{Var}\left(\sum_{k=1}^{K}\sum_{s=1}^{S_k} w_k {S_k}^{-1} h(\theta_{ks})\right) = \sum_{k=1}^{K} {w_k}^2 {S_{\text{eff},k}}^{-1} \text{Var}\left(h(\theta)\right)$, whose minimum will be attained at $w_k = S_{\text{eff},k} / \sum_{k'} S_{\text{eff},k'}$. This also suggests we can estimate the the effective sample size of w-weighted samples by:

$$\hat{S}_{\text{eff}} := \left(\sum_{k=1}^{K} {w_k}^2 {S_{\text{eff},k}}^{-1}\right)^{-1}.$$

To reduce Monte Carlo error, we partially pool stacking weights using a Dirichlet prior with a tuning scale parameter $\lambda > 0$ that controls the amount of partial pooling,

$$p_{\text{prior}}(w_{1,\ldots,K}) = \text{Dirichlet}\left(\frac{\lambda S_{\text{eff},1}}{\sum_{k'=1}^{K} S_{\text{eff},k'}}, \ldots, \frac{\lambda S_{\text{eff},K}}{\sum_{k'=1}^{K} S_{\text{eff},k'}}\right). \tag{8.11}$$

We add this regularization term into (8.7). If $\lambda = 1$ and $S_{\text{eff},k}$ is equal for all k, it becomes the unregularized Bayesian stacking. If $\lambda \to \infty$ and $S_{\text{eff},k} \propto S_k$, it results in the usual Monte Carlo estimate $w_k/S_k = 1/S$. Ideally λ can be further tuned using hold-out data or extra cross validation. In later experiments of this chapter, we simply use $\lambda = 1.001$ as a rule-of-thumb value.

Stacking using a flat prior is always convex, and therefore adding a small λ breaks the tie and makes it strictly convex. If all chains are already mixed, stacking with an informative prior does not hurt, and we will recover the approximately uniform weighting.

Thinning and importance resampling. For settings where it is inconvenient to work with weighted simulation draws, we can perform thinning to obtain a set of S_{thin} simulation draws approximating the weighted mixture of K distributions. We further adopt quasi Monte Carlo strategy to reduce variance. Given weights $\{w_k\}_{k=1}^K$ for K clustered simulation draws $\{\theta_{ks}\}_{k=1,s=1}^{K,\ S_k}$, and an integer $S_{\text{thin}} \leq \inf_k(S_k/w_k)$, we first draw a fixed-sized $S_k^* = \lfloor S_{\text{thin}} w_k \rfloor$ sample randomly without replacement from the k-th cluster, and then sample the remaining $S_{\text{thin}} - \sum_{k=1}^K S_k^*$ without replacement with the probability proportional to $\left(w_k - S_k^*/S_{\text{thin}}\right)$ from cluster k.

Lastly, we have implemented all related functions together to facilitate PSIS-loo based chain-stacking in an `R` package `loo`. It works seamlessly with Stan. See Appendix B for an example.

8.6 Related work

Scalable MCMC. Bayesian inference can be more scalable in advent of parallel distributed computation. Various subsampling methods have been introduced that distribute data batches to parallel nodes and aggregate the resulting inference (Huang and Gelman, 2005; Welling and Teh, 2011; Angelino et al., 2016; Mesquita et al., 2020; Quiroz et al., 2019). These methods typically

rely on certain approximations to rescale the subsampled posteriors.

Our approach is a divide-and-conquer strategy. It allows embarrassingly parallelization and eliminates between-chain communication, which often dominates the budget of parallel computations (Scott et al., 2016). Arguably, stacking does not speed up at all if the posterior is unimodal and tail-log-concave, when usual HMC mixes fast. We are aiming here for complicated models and pathological posterior geometry. We believe that the bottleneck of modern Bayesian computation is sometimes not the input dimension, but the slow mixing rate arising from awkward geometry of metastable distributions.

Multiple starting points. Gelman and Rubin (1992) used multiple sequences and importance resampling to approximate the posterior distribution, where each individual chain was iteratively constructed from a locally Student-t approximation at posterior mode. However, a poor initial point can still slow convergence (Geyer, 1992) because of the use of importance sampling. In our approach, we are less concerned about starting points and only prefer it to be overdispersed.

Raftery and Lewis (1992a,b) suggested to abandon poor initial points coming with slow convergence rate and high autocorrelation by restarting. In the context of multimodality, it is hard to tell if this represents a poor initialization (that sits near the boundary of an attraction region) or a bad mode. A restart may lose the chance to explore some posterior regions.

Our convergence criteria in Section 8.5 are similar to the early approaches on stochastic optimization stopping rules following the capture-recapture model (Good, 1953; Robbins, 1968; Finch et al., 1989). Those analyses were focused on the convergence in parameter space, while ours are directly targeted at the outcome space and are thereby more applicable to models with a large number of disjoint but functionally identical modes.

Approximate inference using mixtures. Although our narrative has been focused on MCMC sampling, stacking can be applied to multiple runs of approximate inference; see examples in Section 8.8. Using mixture distributions to enrich the expressiveness of variational Bayes is not new. Earlier works have used mixture of mean-field approximations to match the posterior (Bishop et al., 1998;

Jaakkola and Jordan, 1998; Gershman et al., 2012; Ranganath et al., 2016; Gal and Ghahramani, 2016; Miller et al., 2017; Chang et al., 2019). However, a direct application of mixture variational methods can be prohibitively expensive in large models, and weights are often fixed to ease the cost. Stacking does not need to specify either the parametric form of the mixture or the number of mixture components, both of which adapt to data and prevent extra model misspecification.

Comparison to importance sampling and Bayesian model (chain) averaging. We compare stacking with other possible chain-combination methods. First we can use importance sampling to adjust for differences between non-mixed chains and the target distribution, with the hope of recovering the exact Bayesian posterior. The importance ratio for the k-th cluster/chain is

$$\text{(Importance sampling)}: \ \alpha_k \propto \frac{1}{S_k}\sum_{s=1}^{S_k} p(\theta_{ks}, y) \approx \int_\Theta p_k(\theta|y)d\theta, \ k = 1, \ldots, K. \tag{8.12}$$

Under the ideal assumption that each chain is well-separated, and all regions of the posterior distributions have been fully explored,

$$\Theta = \bigcup_{k=1}^{K} \Theta_k; \quad \forall k' \neq k, \ p_k\left(\Theta_{k'}\right) \approx 0; \quad s.t. \ \forall \theta \in \theta_K, \ p(\theta|y) \approx \alpha_k p_k(\theta|y). \tag{8.13}$$

Then the importance ratio $p(\theta|y)/p_k(\theta|y) \approx$ constant α_k under $p_k(\cdot|y)$. Hence, the importance resampled draws match the exact posterior. Under the same assumption, the importance ratio α_k is proportional to the marginal likelihood. To see this, rewrite (8.12) as

$$\text{(BMA)}: \ \alpha_k \propto \int_\Theta p(y|\theta)p_{\text{prior}}(\theta)\mathbb{1}(\Theta_k)d\theta = P(y|\Theta_k), \ k = 1, \ldots, K.$$

Under a flat prior $p_{\text{prior}}(\Theta_k) = 1/K$, this leads to $\Pr(\Theta_k|y) \propto \Pr(y|\Theta_k) \propto \alpha_k$. Thus, using the importance ratio (8.12) in the weighted Monte Carlo (8.5) is exactly Bayesian model averaging (BMA, Madigan et al., 1996; Hoeting et al., 1999) on a discrete space $\{\Theta_k : 1 \leq k \leq K\}$. When assumption (8.13) does not hold, importance sampling (8.12) can still be viewed as BMA on models

that are implicitly constructed from each chain. In all experiments later in the chapter, we refer BMA to (8.12).

Yao et al. (2018a) introduced a pseudo-BMA weighting for model averaging. In our context, the pseudo-BMA weight for cluster k is

$$(\text{pseudo}-\text{BMA}):\ \alpha_k \propto \exp\left(\sum_{i=1}^{n}\log p(y_i|y_{-i},\Theta_k)\right) \approx \exp\left(\sum_{i=1}^{n}\log\sum_{s=1}^{S_k}\frac{r_{iks}p(y_i|\theta_{ks})}{r_{iks}}\right),$$

where r_{iks} is the same leave-one-out importance ratio in (8.10).

In comparison, BMA is fully Bayesian under assumption (8.13) and correct model specification. In many approximate inferences, $p_k(\cdot|y)$ is underdispersed and BMA loses mass; Even when using multi-chain MCMC, Θ_k are often duplicate (without clustering) or overlapped. As a result, BMA is sensitive to initialization and priors. Furthermore, Yao et al. (2018a) noted that BMA and pseudo-BMA can perform disastrously when there are many similar weak models in the list of candidate models. Similiary, BMA, importance sampling, and pseudo-BMA overweight "bad" modes when they are oversampled. As discussed by Geyer (1992) a simple unweighted average over chains helps when the starting distribution is close to the target density and chains mix slowly—the scenario in which other naive methods will work, too. In contrast, stacking is invariant to chain duplication and is less sensitive to initial values.

Understanding "overfitting". Overfitting is a combination of model and inference. Given a model, it is possible some regions of posterior distribution more significantly overfits the data than others. Therefore, in (8.7) we use leave-one-out log predictive densities (loo lpd) to evaluate the expected log predictive density (elpd) for each chain even though they come from the same model.

For a fixed K and weights w, in the limit when $n \to \infty$, the mean loo lpd of aggregated chains: $n^{-1}\sum_{i=1}^{n}\log\sum_{k=1}^{K} w_k p_k(y_i|y_{-i})$ converges to elpd: $\mathbb{E}_{\tilde{y}\sim p_{\text{true}}}\log\sum_{k=1}^{K} w_k p_k(\tilde{y}|\theta)p_k(\theta|y)d\theta$ (see Theorem 8.1 for a rigorous statement) . However, stacking uses the data twice, once in the parallel sampling and once in the aggregation of chains. In the aggregation step, we optimize over weights w. If both n and K go to infinity, the aggregated mean loo lpd is no longer an asymptotically

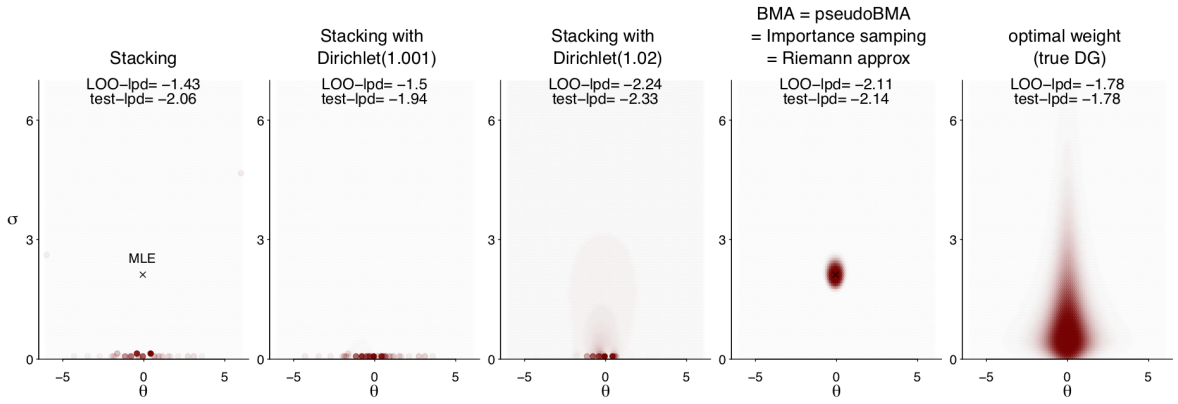


Figure 8.4: *We generate random θ and σ and further y from normal(θ, σ), and stack 9×10^4 points of (θ, σ) from a uniform grid. Stacking overfits due to the non-identification and finite sample size n. BMA, pseudo BMA, and importance sampling are identical and overconfidently concentrate.*

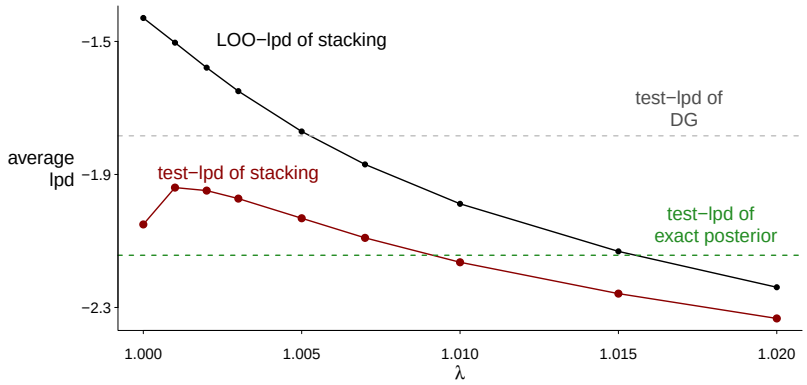


Figure 8.5: *Overfitting is revealed the gap between leave-one-out and test data lpd. A Dirichlet prior with a small λ reduces overfitting in stacking, even better than the exact Bayesian posterior.*

unbiased or consistent estimate of elpd.

Figure 8.4 constructs an extreme example. The data $\{y_i\}_{i=1}^{100}$ are generated from normal(θ_i, σ_i), where θ_i and $\sigma_i > 0$ are generated from a 2-dimensional (half) Student-t distribution centered at 0 and 2 respectively (see the last column).

Now to fit the model $y_i \sim$ iid normal$(\theta, \sigma), i = 1, 2, \ldots, 100$, we draw $K = 9 \times 10^4$ samples of (θ, σ) from uniform$(-7, 7) \times (0, 7)$. We view these as K chains, with one iteration per chain, and compute stacking weights. We evaluate the leave-one-out and test data lpd (using holdout data of size 1000) of stacking, Bayesian posterior, and the true data generating process.

In this setting BMA, pseudo-BMA, importance sampling using a uniform proposal, and Riemann approximation are the same method, all over-concentrated at the posterior mode. Overfitting

is revealed by a large gap between leave-one-out lpd and test data lpd. What's worse, the overconfidence of exact Bayesian inference and BMA will be amplified by a larger n. In contrast, stacking overfits because of nonidentification, and is asymptotically optimal for a fixed K and $n \to \infty$. A Dirichlet prior with small λ reduces overfitting (Figure 8.5) in stacking. It is in agreement with our recommendation in Section 8.5. This example also suggests we can use stacking to re-aggregate a *unimodal* posterior distribution and achieve better prediction than from exact Bayesian inference.

8.7 Asymptotic analysis in a theoretical example

In this section, we analyze the asymptotic behavior of stacking. We first prove that in general chain-stacking is no worse than chain-picking. Then we derive a closed-form solution in a theoretical example to show that with model-misspecification and multimodal posterior, chain-stacking can be predictively better than the exact posterior inference.

Optimality of the stacked predictive distribution. The stacking weights are *not* the same as posterior masses of each mode. Even asymptotically, minimizing cross validation errors is different from integrating the target distribution. Theorem 8.1 affirms that the stacked inference is optimal from a predictive paradigm—it asymptotically maximizes the expected log predictive densities (elpd) among all linearly weighted combination of parallel chains of form (8.5).

Theorem 8.1 (asymptotics of chain-stacking)**.** *Assuming we draw S posterior samples in each chain from their stationary distribution p_k, and we approximate the leave-one-out distribution by PSIS as in* (8.10)*, $p^S_{k,-i}(y_i) = \sum_{s=1}^{S_k} p_k(y_i|\theta_{ks}) r_{iks} / \sum_{s=1}^{S_k} r_{iks}$, then for a fixed number of chains K and a fixed weight vector w, when in the limit of both the size of observations n and number of posterior draws S, under regularities conditions (see Appendix), the objective function in stacking converges to stacked elpd:*

$$\frac{1}{n}\sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k p^S_{k,-i}(y_i) - \mathrm{E}_{\tilde{y}|y_{1:n}} \log \sum_{k=1}^{K} w_k p_k(\tilde{y}|y_{1:n}) \xrightarrow{L_2} 0, \quad n \to \infty,\ S \to \infty.$$

Cauchy example revisit: When can stacking outperform exact Bayes in the limit? Let's revisit the Cauchy mixture example in Section 8.3 and Figure 8.2. We consider univariate observations $y_{1,\ldots n}$ iid from the data generating process,

$$\text{DG}: \; y_i \sim \text{Cauchy}\left((2z_i - 1)a, 1\right), \quad z_i \sim \text{Bernoulli}(p_0), \quad i = 1, 2, \ldots, n.$$

In other words, y is either Cauchy$(a, 1)$ or Cauchy$(-a, 1)$ with probabilities p_0 and $1 - p_0$, where the location $a > 0$ and probability $p_0 \in [0.5, 1]$ are unknown constants (the $0 \leq p_0 < 0.5$ counterpart is symmetric and hence omitted). We denoted the density of this data generating process by $p_{\text{true}}(y)$.

We now fit y with the iid Cauchy likelihood with unknown parameter μ and a prior $p_0(\mu)$ that has full support on $\mathbb{R}$,

$$\text{Model}: \; y_i \sim \text{Cauchy}(\mu, 1), \quad \mu \sim p_0(\mu), \quad \mu \in \mathbb{R}\,.$$

In particular, the data generating process can be expressed from this model if an inference θ is given by a mixture of two points,

$$\text{express DG in Model}: \; \mu \sim p_0\delta(a) + (1 - p_0)\delta(-a).$$

The following theorems characterize the behavior of modes and the concentration of exact full Bayesian inference in the limit of large n. Proofs and related lemmas appear in Appendix A.

Theorem 8.2 (characterize posterior mode in the Cauchy example)**.** *We construct a deterministic function $\xi(a)$ as visualized in Figure 8.6. It is an increasing function of a, with $\xi(2) = 0.5$ and $\xi(\infty) = 1$. The modality of posterior density $p(\mu|y_1, \ldots, y_n)$ has a closed form determination.*

(a) For any $a > 2$, and $p_0 \geq \xi(a)$, there exists a large N, such that for all $n > N$, the posterior is unimodal. The peak is near $\mu = a$ for a large a.

(b) For any $a > 2$, and $0.5 \leq p_0 < \xi(a)$, there exists a large N, such that for all $n > N$, the posterior is bimodal. The two local maximums are near $(-a, a)$ for a large a.

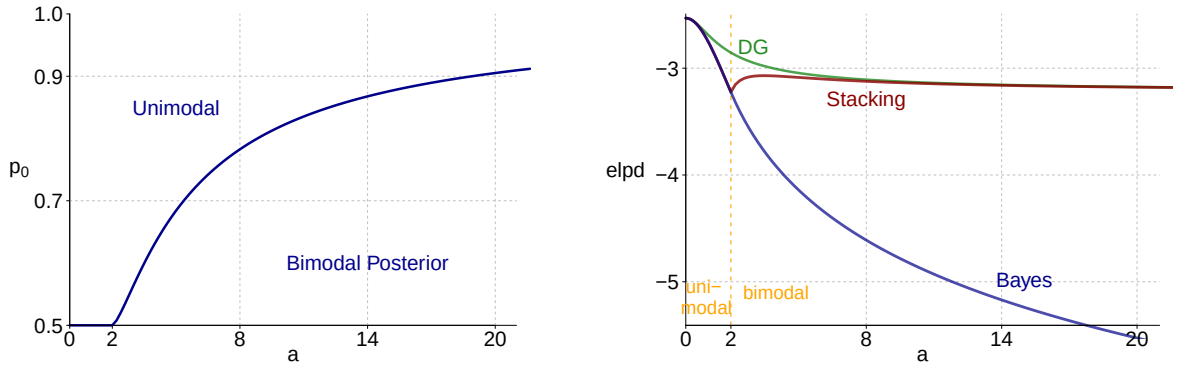


Figure 8.6: *Left: the deterministic function $\xi(a)$. For any $a > 2$ the posterior is bimodal with a large n if and only if $p_0 < \xi(a)$. Right: the elpd of the true data generating process and the asymptotic ($n \to \infty$) elpd of full Bayes and multi-chain stacking at $p_0 = 0.5$. When $p_0 = 0.5$, $a < 2$ the posterior is unimodally spiked at 0, and stacking is identical to Bayes.*

(c) For any $0 < a < 2$, there exists a large N, such that for all $n > N$, the posterior is unimodal with global maximum between 0 and a. If further $p_0 = 0.5$, the maximum is at 0.

(e) When $a > 2, p_0 = 0.5$ and equipped a symmetric prior $p(\mu) = p(-\mu)$, there exists a large N such that, for all $n > N$, the posterior is always bimodal with two maximums, which asymptotically ($n \to \infty$) converge to $\mu = \pm\sqrt{a^2 - 4}$.

Theorem 8.3 (posterior convergence in the Cauchy example)**.** *(a) For any $a > 2$, and $p_0 > 0.5$, the posterior distribution $p(\theta|y_1, \ldots, y_n)$ converges in distribution to a point mass $\delta(\gamma)$ as $n \to \infty$, where $\gamma = \gamma(p_0, a)$ depends on p_0 and a.*

(b) For any $a > 2$, $p_0 = 0.5$, a prior that is symmetric $p(\mu) = p(-\mu)$, the posterior distribution $p(\theta|y_1, \ldots, y_n)$ is asymptotically only charged at two points $\pm\gamma$, with a closed form expression $\gamma = \sqrt{a^2 - 4}$. More precisely, the posterior distribution $p(\theta|y_1, \ldots, y_n)$ is almost surely concentrated at $\pm\sqrt{a^2 - 4}$ with equal probabilities $1/2$.

(c) Under the same condition in (b), for any $\eta > 0$, almost surely the following limits hold,

$$\limsup_{n\to\infty} \Pr\left(\left|\mu - \sqrt{a^2-4}\right| < \eta \mid y_1, \ldots, y_n\right) = \limsup_{n\to\infty} \Pr\left(\left|\mu + \sqrt{a^2-4}\right| < \eta \mid y_1, \ldots, y_n\right) = 1$$

When $a > 2$, if $0.5 < p_0 \leq \xi(a)$, two modes (γ^+, γ^-) exist, but the exact inference will asymptotically concentrate at the right mode $\gamma = \gamma^+$. Even when $p_0 = 0.5$ so that the two centers

$\pm a$ are equally important in the data generating process, the exact inference would still pick one mode asymptotically, with the left and right mode having equal chances of being selected.

Corollary 8.4 (asymptotic elpd of chain-stacking)**.** *In all cases in Theorem 8.3, the expected log predictive density (elpd) from the exact Bayesian posterior* $p(\mu|y_1,\ldots,y_n)$ *is*

$$\begin{aligned}\text{elpd}_{\text{bayes}} &= \int_{\mathbb{R}} p_{\text{true}}(\tilde{y}|p_0)\log\int_{\mathbb{R}} p(\tilde{y}|\mu)p(\mu|y_1,\ldots,y_n)d\mu d\tilde{y}\\ &\xrightarrow{n\to\infty} -\left(p_0\log\left(\pi(4+(\gamma-a)^2)\right)+(1-p_0)\log\left(\pi(4+(\gamma+a)^2)\right)\right)\\ &\overset{a \text{ is large}}{\approx} -(1-p_0)\log\left(1+a^2\right)-\log 4\pi.\end{aligned}$$

When $a > 2$, and $0.5 \le p_0 \le \xi(a)$, the two modes (γ^+,γ^-) are detectable from multi-chain MCMC. In this case, stacking behaves better than exact Bayesian inference. Indeed, the next corollary shows that stacking approximates the data generating process in KL divergence.

Corollary 8.5 (asymptotic optimality of chain-stacking)**.** *(a) When n is large, for any* $a > 2$ *and* $0.5 < p_0 < \xi(a)$*, both modes* γ^- γ^+ *receive asymptotically nonzero weights, and the elpd of the stacking average,*

$$\text{elpd}_{\text{stacking}} = \int_{\mathbb{R}} p_{\text{true}}(\tilde{y}|p_0)\log\int_{\mathbb{R}} p(\tilde{y}|\mu)p_{\text{stacking}}(\mu|y_1,\ldots,y_n)d\mu d\tilde{y},$$

is strictly larger than $\text{elpd}_{\text{bayes}}$.

(b) When a is large, stacking weights for (γ^-,γ^+) *are asymptotically close to* $1-p_0$ *and* p_0*. Consequently, the stacked posterior predictive distribution approximates the data generating process,*

$$\text{KL}\left(p_{\text{true}}(\cdot),\ \int_{\mathbb{R}} p(\cdot|\mu)p_{\text{stacking}}(\mu|y_1,\ldots,y_n)d\mu\right)\gtrapprox 0,\ \text{ when } n\to\infty, a \text{ is fixed and large.}$$

When n is large, for $a > 2, p_0 = 0.5$, the stacking weights for two modes $\pm\sqrt{a^2-4}$ are asymptotically equally 0.5. We analytically evaluate the elpd under the true data generating

process, the asymptotic ($n \to \infty$) elpd of full Bayes, and multiple chain stacking in the right panel of Figure 8.6. Stacking is predictively superior to the full Bayes. The elpd difference between the data generating process and stacking vanishes for a large a, implying the KL divergence between them approaches 0.

Lastly, our Cauchy example at $p_0 = 0.5$ might remind readers of the one constructed by Diaconis and Freedman (1986a,b). They used a Dirichlet prior with the parameter measure Cauchy$(\mu, 1)$ to fit observations essentially coming from $y \sim 0.5\delta(a) + 0.5\delta(-a)$ with $a > 1$. The resulting Bayesian posterior of μ is concentrated at $\pm\sqrt{a^2 - 1}$. However, instead emphasizing the inconsistency of this Bayesian procedure, we use our example to praise stacking: it approximates the *true* data generating process given an *misspecified* model, *inconsistent* Bayesian inference, and *non-mixing* samplers.

8.8 Examples

We demonstrate effectiveness of stacking by applying it to a series of challenging problems.

Latent Dirichlet allocation. Latent Dirichlet allocation (LDA, Blei et al., 2003) is a mixed-membership clustering model widely used in natural language processing, computer vision, and population genetics. In the model, the j-th document ($1 \le j \le J$) is drawn from the l-th topic ($1 \le l \le L$) with probability θ_{jl}, where the topic is defined by a vector of probability distribution ϕ_l over the vocabulary, such that each word in the document from topic l is independently drawn from a multinomial distribution with probability ϕ_l.

Despite its popularity for data exploration, LDA suffers from computational instability as the inference may not replicate itself from either multiple runs (Mäntylä et al., 2018) or data shuffle (Agrawal et al., 2018). This confuses users as a different result is produced from each new run, and reduces the predictive power of text mining classifiers. Past literature suggests to examine and select one best fit from multiple unstable inference results subjectively or through cross validation, or to conduct extra manual tuning for hyperparameters to get rid of posterior multimodality, which however changes the original model and may further undermine classification efficiency (Tian et al.,

Chain weight	Top words in the topic
0.20	mr, man, wickham, good, give, young, lydia
0.18	mr, man, young, bingley, collins, darcy
0.13	mr, lady, catherine, dear, great, young
0.12	wickham, elizabeth, mr, darcy, replied, hope
0.09	elizabeth, darcy, mr, sister, wickham, make

Figure 8.7: *Weights of the top 5 chains in the LDA model with $L = 5$, and top words in the topic that the first paragraph belongs to computed from these 5 chains.*

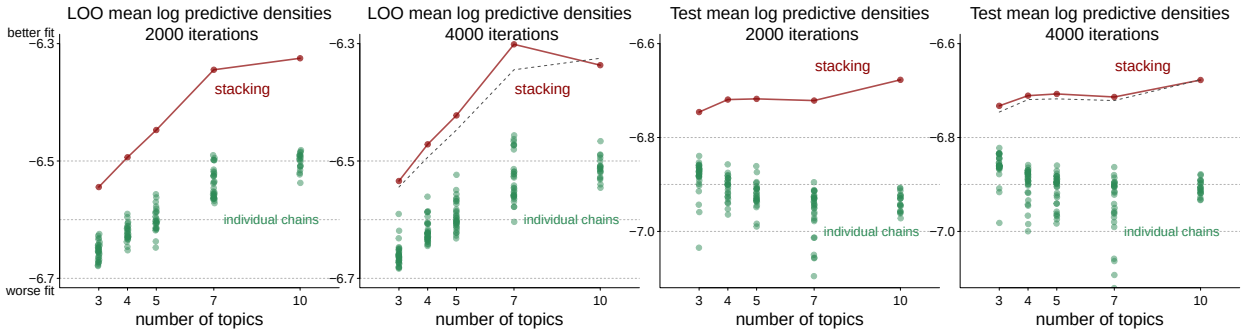


Figure 8.8: *The mean log predictive densities from 30 randomly initialized chains, and the stacked average of them, evaluated using both leave-one-word-out and independent test data. The number of topics L in the LDA model varies from 3 to 10, and each chain contains 2000 or 4000 iterations. Individual chains do not mix, and the best of them is invariably worse than stacking.*

2009; Carreño and Winbladh, 2013).

We apply an LDA topic model to texts in the novel *Pride and Prejudice*. After removing frequent and rare words, the book contains 2025 paragraphs and 32877 words, with a total unique vocabulary size of 1495. We randomly split the words in the data into 70% training and 30% test. The dimension of the parameters θ and ϕ grows as a function of the number of topics L by $2025 \times L$ and $L \times 1495$ respectively. We place independent Dirichlet(0.1) priors on θ and ϕ. We vary L from 3 to 15, and for each fixed model we sample with Stan using 30 parallel chains initialized at random starting points with 2000 or 4000 iterations per chain.

Due to the multimodal posterior $p(\phi, \theta|y)$, individual chains do not mix after 4000 iterations. As represented by green dots in Figure 8.8, different chains yield different log predictive densities on test data, suggesting the multimodality is more than label-switching. Figure 8.7 lists, for five runs, the top words in the topic to which the first paragraph belongs.

Following our stacking approach, the 30-chain-stacked average (red line in Figure 8.8) improves

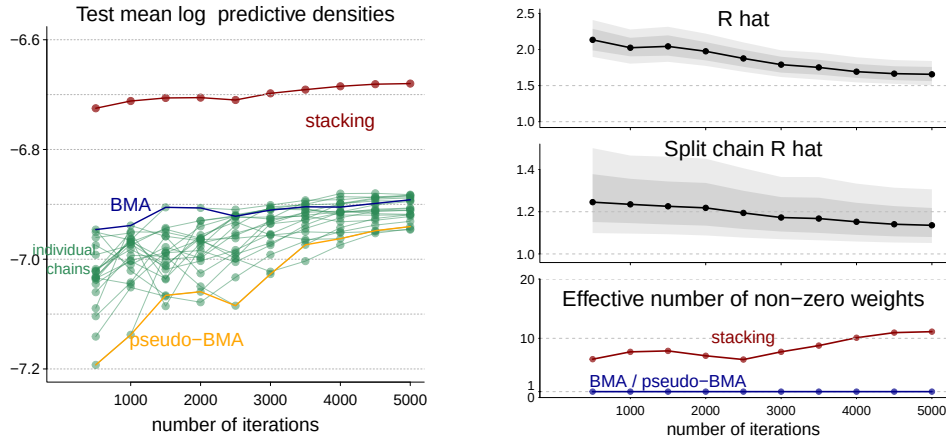


Figure 8.9: *Stacking benefits from early-stopped MCMC. We run LDA with* $L = 10$ *topics on 30 chains. As the number of iterations increase from 500 to 5000, the test lpd of individual chains increases, while the stacked average has a flatter slope, indicating we can stop early without losing much predictive power. Monitoring* $\widehat{R}$ *and split-chain* $\widehat{R}$ *of all pointwise likelihoods, we find that* $\widehat{R}$ *is much bigger than split-*$\widehat{R}$*. The bottom right shows the effective number of nonzero weights. BMA and pseudo-BMA put nearly all weight on one chain.*

the model fit compared with even the best of individual chains by orders of magnitude, measured in test data mean log predictive densities. Indeed, the improvement of stacking in mean lpd (≈ 0.2) is standardized by sample size and equivalent to roughly an $\exp(10^5)$ outperforming margin in the scale of Bayes factors. There is a mismatch between the trend of loo and test lpd, indicating the inconsistency of single chain loo-selection. This may come from (a) the non-iid nature of textual data, and (b) the parameter size is nearly the same as sample size such that loo has not reached its consistency territory. But even so, stacking still performs well in test data and can be combined with other predictive metric such as leave-one-document-out.

The left panel of Figure 8.9 shows the test data predictive performance using varying number of itearations from 500 to 5000 (with fixed number of topics $L = 10$). As the number of iterations increase, test lpd from inferences using individual chains elevates, while the stacked average has a flatter slope, indicating that we can stop earlier and stack chains without losing much predictive power, even though these chains are not completely mixed. The upper and middle right panel show median, 30% and 50% central interval of $\widehat{R}$ and split-chain $\widehat{R}$ for all pointwise likelihoods. $\widehat{R}$ is much bigger than split chain $\widehat{R}$, suggesting that the non-mixing is mostly due to lack of between-

mode transitions. Given that in this problem sampling takes up to 12 hours CPU time per chain per 1000 iterations, such *early stopping of iterations* provides a remarkable opportunity to reduce computation costs. This is also manifested in Figure 8.8: for all $L \in [3, 10]$, individual chains perform better when per-chain iterations increase from 2000 to 4000, whereas the stacked average remains nearly unchanged (compare the red and dashed grey lines in the second and forth panel).

The bottom right panel of 8.9 shows the effective number of nonzero weights. In agreement with our theoretical discussion, BMA and pseudo-BMA put nearly all mass onto one chain, and in fact they often do not even select the optimal chain for the test data (left column). Accordingly, it is no surprise that stacking outperforms BMA and pseudo-BMA.

In addition to the benefit of early stopping of iterations, stacking provides an extra bonus of *early stopping of topics*. Usually, the number of topics L involves manual tuning. *Stacking effectively expands the model space*. Therefore, we observe in the right two panels of Figure 8.8 that the stacked average is less sensitive to L in test data lpd. Stacking compensates the lack of mixture components in the model through additional mixtures of posteriors during chain aggregation.

Gaussian process regression. Consider a regression problem with scalar observations $y_i = f(x_i) + \epsilon_i, i = 1, ..., n$, at input locations $X = \{x_i\}_{i=1}^n$, and ϵ_i are independent noises. We place a Gaussian process prior on latent functions f with zero mean and squared exponential covariance. In the next two experiments, we apply stacking to remedy bimodality in hyperparameter *optimization*, and slow mixing in *sampling*, respectively.

Combining modes in hyperparameter optimization. In Gaussian process regression, posterior bimodality can occur even with a normal likelihood:

$$y_i = f(x_i) + \epsilon_i,\ \epsilon_i \sim \text{normal}(0, \sigma),\ f(x) \sim \mathcal{GP}\left(0, \alpha^2 \exp\left(-\frac{(x - x')^2}{\rho^2}\right)\right). \tag{8.14}$$

We use data from Neal (1998). The univariate input x is distributed normal$(0, 1)$, and the corresponding outcome y is also Gaussian with standard deviation 0.1. With probability 0.05, the point

is considered an outlier and the standard deviation is inflated to 1. In all cases, the true mean of $y|x$ is

$$f_{\text{true}}(x) = 0.3 + 0.4x + 0.5\sin(2.7x) + 1.1/(1+x^2). \tag{8.15}$$

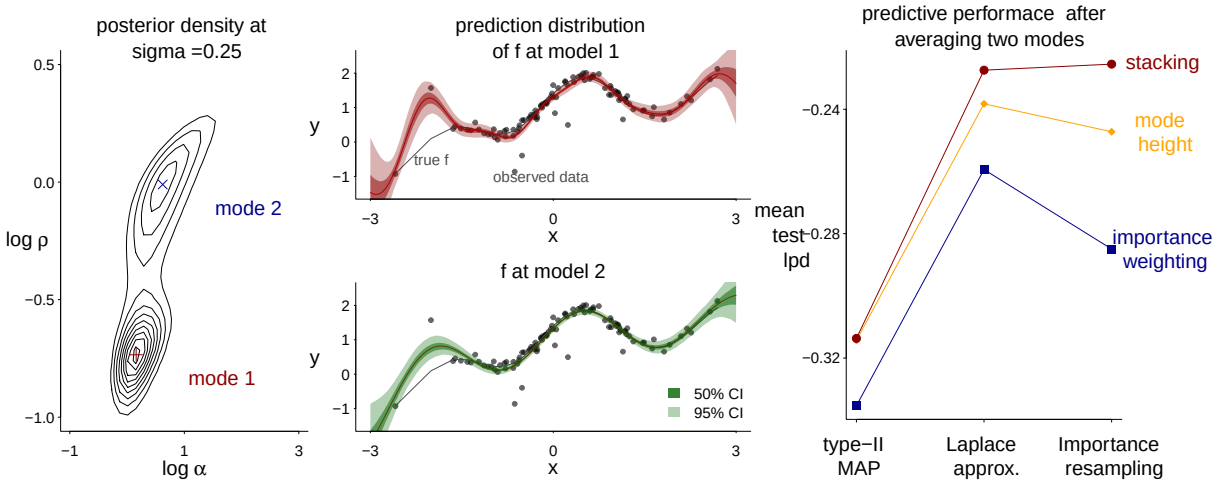


Figure 8.10: *The posterior distribution of hyperparameters $p(\rho, \alpha, \sigma|y)$ has at least two local modes. The left panel shows contours of the marginal posterior of ρ and α at fixed $\sigma = 0.25$. The middle panel shows draws from the posterior predictive distribution $f|y$ at the two hyperparameter modes. We can either pick these two modes as type-II MAP or locally approximate the posterior of hyperparameters at the modes by Laplace approximation or uniform-grid importance resampling. Then the resulting modes or local approximation can be combined according to stacking, mode height, or importance weighting. The right panel shows that stacking performs the best on test data log predictive densities for all schemes.*

Model (8.14) requires inference on $f(x_i)$ and all hyperparameters $\theta = (\alpha, \rho, \sigma)$. We integrate out all $f(x_i)$ and obtain the marginal posterior distribution

$$\log p(\theta|y) = -\frac{1}{2} y^T \left(K(X,X) + \sigma^2 I\right)^{-1} y - \frac{1}{2}\log|K(X,X) + \sigma^2 I| + \log p(\theta) + \text{constant}, \tag{8.16}$$

where $p(\theta)$ is the prior for which we choose an elementwise Cauchy$^+(0,3)$.

In Neal's dataset with training sample size $n = 100$, at least two local maxima of (8.16) can be found. We visualize the marginal distribution of $p(\rho, \sigma|y)$ at $\sigma = 0.25$ on the leftmost of Figure 8.10.

Now we consider three standard mode-based approximate inference of $\theta|y$:

a. Type-II MAP. The value $\hat{\theta}$ that maximizes the marginal distribution (8.16) is called the type-II MAP estimate. Using this point estimate of hyperparameters $\theta = \hat{\theta}$, we further draw $f|\hat{\theta}, y$.

b. Laplace approximation. We compute Σ: the inverse of the negative Hessian matrix of (8.16) at the local mode $\hat{\theta}$, draw z from multi-variate-normal$(0, I_3)$, and use $\theta(z) = \hat{\theta} + \mathrm{V}\Lambda^{1/2}z$ as the approximate posterior samples around the mode $\hat{\theta}$, where the matrices V, Λ are from the eigendecomposition $\Sigma = \mathrm{V}\Lambda^{1/2}\mathrm{V}^T$.

c. Importance resampling. Instead of standard Gaussians in the Laplace approximation, we now draw z from uniform$(-4, 4)$, and then resample z without replacement with probability proportional to $p\left(\theta(z)|y\right)$ and use the kept samples of $\theta(z)$ as an approximation of $p(\theta|y)$.

In the existence of two local modes $\hat{\theta}_1, \hat{\theta}_2$, we either obtain two MAPs, or two nearly nonoverlapped draws, $(\theta_{1s})_{s=1}^S, (\theta_{2s})_{s=1}^S$. We then evaluate the predictive distribution of f, $p_k(f|y,\theta) = \int p(f|y,\theta)q(\theta|\hat{\theta}_k)d\theta,\ k = 1, 2$, where $q(\theta|\hat{\theta}_k)$ is a delta function at the mode $\hat{\theta}_k$, or the draws from the Laplace approximation and importance resampling that is expanded at $\hat{\theta}_k$. We visualize the predictive distribution of f using two local MAP estimates in the middle panel of Figure 8.10. The one with the smaller length scale is more wiggling and passes the training data more closely.

For each of these three mode-based inferences, we consider three strategies to combine two modes:

a. Mode height. We reweigh the predictive distribution of f according to the height of the marginal posterior density at the the mode: $w_k \propto p(\hat{\theta}_k|y), k = 1, 2$.

b. Importance weighting. For approximate posterior draws $(\theta_{1s})_{s=1}^S, (\theta_{2s})_{s=1}^S$, we reweigh them proportional to the mean marginal posterior density $w_k \propto 1/S \sum_{s=1}^S p(\theta_{ks}|y)$. We choose the importance weights of two MAPs using the ones from importance resampling as it approximates the total posterior mass in the surrounding region near the mode.

c. Stacking. Our fast approximate loo does not apply to MAP estimation directly. Therefore, we split the data into training y_{train} and validation data y_{val}. We first obtain either MAPs or approximate hyperparameter draws using training data and optimize their predic-

tions on validation data. Stacking maximizes $\sum_{i=1}^{n_{\text{val}}} \log \left(\sum_{k=1}^{K} w_k p(y_{\text{val},i} | y_{\text{train}}, \hat{\theta}_k) \right)$ for MAPs or $\sum_{i=1}^{n_{\text{val}}} \log \left(\frac{1}{S} \sum_{k=1}^{K} w_k \sum_{s=1}^{S} p(y_{\text{val},i} | y_{\text{train}}, \theta_{ks}) \right)$ for Laplace and importance resampling draws.

In the right panel of Figure 8.10, we evaluate these three weighting strategies by computing the mean expected log predictive density of the combined posterior distribution on hold-out test data ($n_{\text{test}} = 300$). No matter whether we are combining two point estimates or two distinct Laplace/importance resampling draws near the two modes, the stacking weights provide better predictive performance on test data.

Combining non-mixed chains from Gaussian process regression with a Student-t likelihood. Neal (1998) originally constructed this example in which noise ϵ_i in (8.14) is modeled by a t distribution with mean 0, scale σ and degrees of freedom ν:

$$p(y_i | f_i, \sigma, \nu) = \frac{\Gamma((\nu + 1/)2)}{\Gamma(\nu/2)\sqrt{\nu\pi}\sigma} \left(1 + \frac{(y_i - f_i)^2}{\nu\sigma^2} \right)^{-(\nu+1)/2}, \; f \sim \mathcal{GP} \left(0, \alpha^2 \exp \left(-\frac{(x - x')^2}{\rho^2} \right) \right).$$

The Student-t model is robust to outlying observations but is computationally challenging, because of (a) lack of closed-form expression for $p(f|y)$, and (b) heavy-tailed posterior densities. Approximate methods exist, such as factorizing variational approximation (Tipping and Lawrence, 2005), Laplace approximation (Vanhatalo et al., 2009), and expectation propagation (Jylänki et al., 2011), but posterior sampling remains difficult.

We generate training data $x_{1:n}$ from uniform$(-3, 3)$, and the outcome y_i has the same mean in (8.15). y_i either has standard deviation $\sigma_1 = 0.1$, or inflated to $\sigma_2 > 0.1$ with probability proportional to $\exp \left(- \left(C \frac{i - 0.4n}{n} \right)^2 \right)$, where $C > 0$ is a concentration factor that decides how the outliers are concentrated with each other in x-space. In the experiment, we vary σ_2 from 0.1 to 1 and C from 1 to 8. $n_{\text{test}} = 300$ hold-out test data points $(\tilde{X}_i, \tilde{y}_i)_{i=1}^{n_{\text{test}}}$ are generated from the same mechanism.

We fix the degrees of freedom $\nu = 2$ and sample from the full posterior distribution $p(f_1, \ldots, f_n, \sigma, \alpha, \rho)$ from $K = 8$ parallel chains and 8000 iterations per chain in Stan. We draw initialization from uniform$(-10, 10)$ for unconstrained parameters and set the maximum tree depth

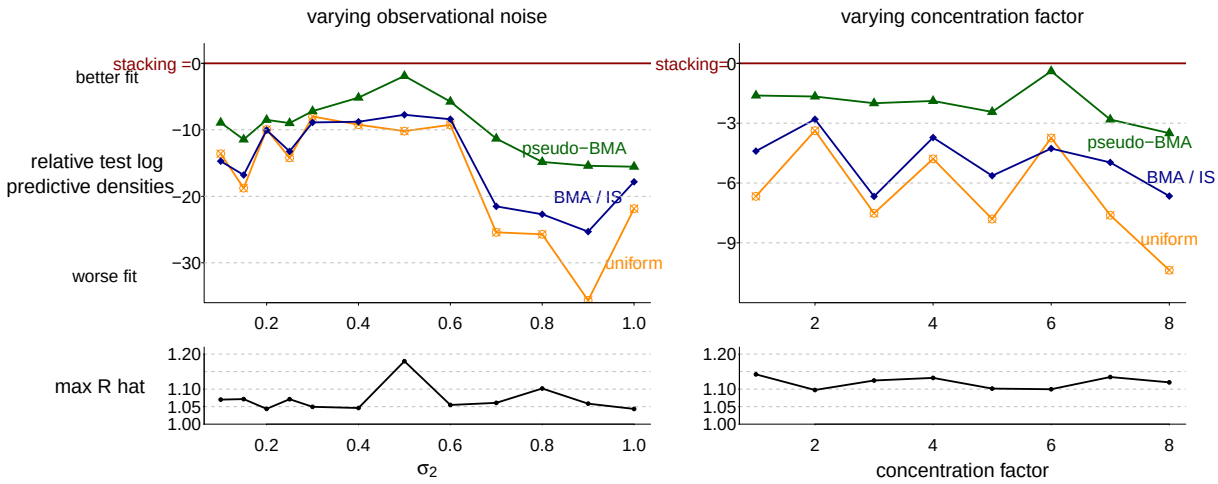


Figure 8.11: *Left: We fix the concentration factor $C = 5$ and vary the outlier standard deviation σ_2 from 0.1 to 1 in the data generating mechanism. Right: We fix $\sigma = 0.3$ and vary the concentration factor C from 1 to 8. In each setting, we sample from the posterior distribution using 8 chains with 8000 iterations each, and combine chains using four weighting methods. We report the test log predictive densities (using $n_{\text{test}} = 300$ independent test data) of three other methods subtracting stacking, which are always negative. The lower row reports the maximum $\widehat{R}$ among all parameters.*

to 5 for the NUTS algorihm. In the lower row of Figure 8.11, we report the maximum $\widehat{R}$ of all sampling parameters among 8 chains: clearly these do not mix in all settings.

We compare four chain-combination strategies: BMA, pseudo-BMA, uniform averaging, and stacking. After each iteration of $(\sigma, \rho, \alpha, f)$, we draw posterior predictive sample of $\tilde{f} = f(\tilde{X})$, and compute the mean test data log predictive densities. Since test performance changes in orders of magnitude under different data-generating settings, in Figure 8.11 we use stacking as a baseline and compare the test log predictive densities of other methods by subtracting stacking ones. In all cases, stacking outperforms other three approaches.

There are three contributors to the poor mixing in this example. First, chainwise predictions may diverge even when parameters are nearly mixed. Figure 8.12 display sampling results for a dataset with $n = 20, \sigma_2 = 0.6, C = 5$. In the leftmost column, all $(\sigma, \rho, \alpha, f)$ and transformed parameters have $\widehat{R} < 1.05$. But the log predictive densities are different across chains, shown in the second column (chains have have been re-ordered by test lpd). Stacking detects this difference via leave-one-out cross validation.

Second, the posterior distribution $f|y$ can be multimodal. The rightmost column of Figure 8.12

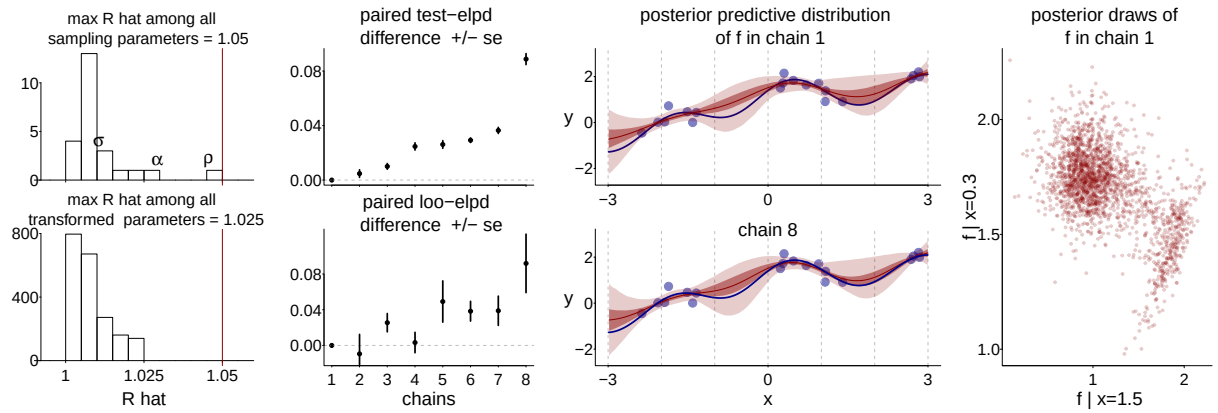


Figure 8.12: *In an experiment ($n = 20, \sigma_2 = 0.6, C = 5$) even when $\widehat{R}$ of all parameters are smaller than 1.05, 8 chains exhibit different predictive capabilities. The second column shows the estimated log predictive densities subtracting chain 1 and the standard error in test data or loo. Chains have been reordered by test scores. The third column shows the prediction of f in chain 1 and 8. The rightmost column is the joint posterior predictive draw f at $x = 1.5$ and 0.3 in chain 1.*

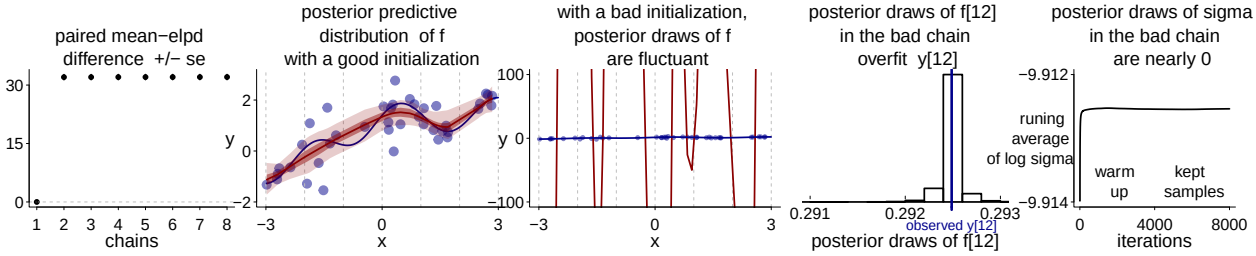


Figure 8.13: *In an experiment ($n = 40, \sigma_2 = 1, C = 5$), chain 1 is trapped in a bad local mode, where the posterior $f|y$ is narrow and fluctuate. It overfits observed value, and σ is trapped near 0 among 8000 iterations. It has a low elpd on both test data and loo, hence abandoned in stacking.*

displays the joint posterior distribution of f conditioning on $x = 0.3$ and 1.5, clearly bimodal. In this example, this is not a sampling concern owing to the small between-mode energy barrier, and HMC/NUTS sampler in Stan is able to move between these two modes rapidly.

Third, some chains may be trapped in bad local modes. In Figure 8.13, we outline the sampling result from another dataset ($n = 40, \sigma_2 = 1, C = 5$). Chain 1 is trapped in a local mode with $\sigma \approx 0$ and is unable to escape the local trap after 8000 iterations. The posterior prediction f fluctuates and overfits the observations: $f_{12}|y$ is nearly a delta function at y_{12}. The strong overfitting of this chain leads to a low elpd on both test data and leave-one-out cross validation, hence it is abandoned by stacking.

Hierarchical models. Consider observations from J exchangeable groups. For simplicity we assume a balanced one-way design, with data $y_{ij}, i = 1, \ldots, N$ from groups j. We apply a hierarchical model with parameters $(\theta, \sigma, \mu, \tau)$,

$$\text{centered}: \quad y_{ij}|\theta, \sigma \sim \text{normal}(\theta_j, \sigma),\ \theta_j|\mu, \tau \sim \text{normal}(\mu, \tau),\ 1 \le i \le N,\ 1 \le j \le J. \tag{8.17}$$

Sampling in the space of $(\theta, \sigma, \mu, \tau)$ is called *centered parameterization*. When the likelihood is not strongly informative, the prior dependence between τ and θ in (8.18) can produce a funnel-shaped posterior that is non-log-concave, and slow-to-mix near $\tau = 0$, due to a large entropic barrier.

Alternatively, with *non-centered parameterization*, sampling occurs in the space of (ξ, σ, μ, τ) through a bijective mapping $\theta_j = \mu + \tau\xi_j$, and the model is equivalently reparameterized by

$$\text{non–centered}: \quad y_{ij} \sim \text{normal}(\mu + \tau\xi_j, \sigma),\ \xi_j \sim \text{normal}(0, 1),\ 1 \le i \le N,\ 1 \le j \le J. \tag{8.18}$$

When the likelihood is not strongly informative, the non-centered parameterization is preferred (Betancourt and Girolami, 2015; Gorinova et al., 2020), but when the likelihood is strongly informative, then the non-centered parameterized posterior has a funnel shape. The data informativeness can be crudely measured by the inverse of F-statistics (between group variance divided by within group variance). But beyond such heuristics and limited classes of models where analytic results can be applied, there is no general guidance on which parameterization to adopt.

Parallel to the slow mixing rate due to the funnel shaped posterior, the posterior in (8.17) can contain two modes, usually arising when the data indicate a larger between-group variance than does the prior. Liu and Hodges (2003) characterized bimodality of this model under conjugate priors in closed form.

To understand how the posterior bimodality affects sampling efficiency, in the first simulation we generate data from $J = 8$ groups and $N = 10$ observations per group. The true τ and σ vary from 0.1 to 20, with a varying amount of t-distributed noise added to θ. We place a conjugate inverse-gamma$(0.1, 0.1)$ prior on both τ^2 and σ^2. For every realization of data, we sample from

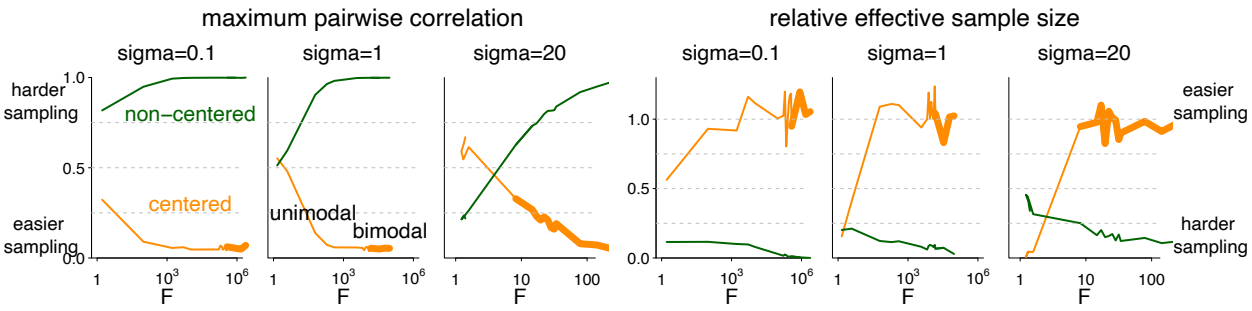


Figure 8.14: *We fit the hierarchical model on data simulated from by various generating process. When the between group variation is large or the within group variation is small, whose ratio is the sample F statistics, the centered parameterization is more efficient, amid less correlated posterior and large effective sample size. Counterintuitively, this is also when the posterior bimodality occurs.*

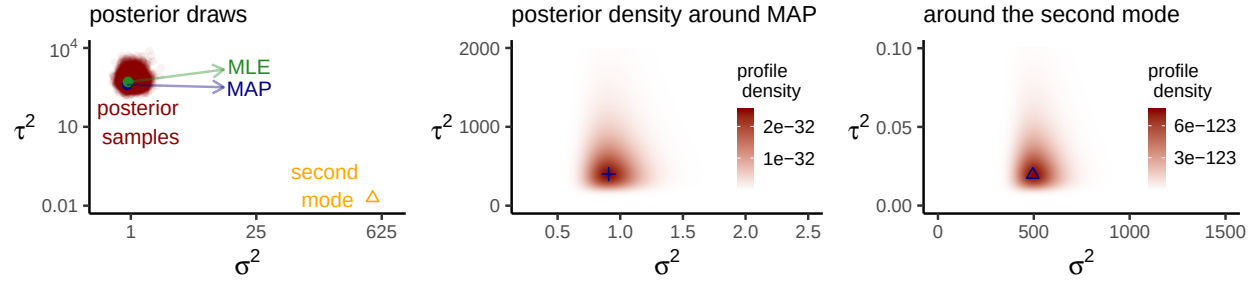


Figure 8.15: *A grid search finds two posterior modes when data are generated by* $\sigma = 1$ *and* $\tau = 25$*. The second mode in density and prediction ability, ignored by posterior sampling.*

the posterior distribution in both centered and non-centered parameterization using 4000 iterations, and analytically determine whether the centered parameterization has two posterior modes.

In Figure 8.14, we assess the maximum absolute parameterwise correlations (left three columns), and the relative effective sample size (ESS divided by total iterations, right three columns) in posterior samples. Conforming our heuristics, when between-group variation is large and within-group variation is small, the centered parameterization is more efficient, and vice versa.

Surprisingly, in this example metastability and multimodality evolve in opposite directions. In Figure 8.14 we visualize the occurrence of posterior bimodality in centered parameterization by a thicker line width. When the between-group variation increases, the centered posterior eventually becomes bimodal, but sampling becomes more efficient.

How is this possible? Figure 8.15 presents an example where the data are generated by $\sigma = 1$

and $\tau = 25$. Both the MAP and MLE are close to the true value. A second local mode explains all variation by a large σ (opposite to Figure 8.13), but it is orders of magnitude lower than the first one in posterior densities, hence ignored by sampling. That's why the centered parameterization runs smoothly in the existence of posterior bimodality. The bad mode also has a low loo elpd, so stacking assigns it zero weight when we combine modes.

This simulation leaves a few open problems: which parameterization to choose in practice, whether the sample has included all local modes, whether the ignored modes are predictively important, and if we should search for them in the first place. The bimodality analysis of Liu and Hodges (2003) applies to conjugate priors. But multimodality readily exists in hierarchical models. To be specific, when the group-level standard deviation τ has a flat prior, $\tau = 0$ is *always* a mode of the joint posterior distribution. From the modeling perspective, this mode represents complete pooling.

Given that the centered parameterization behaves like an implicit truncation and has sampling difficulty in the small τ region, we propose a stacking-based solution for reparameterizations. We run $K + 1$ chains. The first chain is complete pooling: restricting $\tau = 0$ and $\theta_j = \theta_1$. The next K parallel chains are centered parameterization with a zero-avoiding prior (Chung et al., 2013) on τ. Finally, we use stacking to average these $K+1$ chains. Intuitively, if $\tau \approx 0$ is predicatively important but missed by the implicitly left truncated centered parameterization, the first chain fills the hole; when $\tau \approx 0$ is incompetent, the centered sampling is boosted by circumventing the computationally intensive region $\tau \approx 0$.

To validate our proposal, we simulate data with dimensions $J = 100$ (number of groups) and $N = 20$ (observations per group). We vary the true within-group standard deviation σ from 0.1 to 100 and add between-group noises Bv_j to θ_j, where B is a constant scalar varying from 0 to 50, and each v_j is an independent Student-$t(1)$ noise. We place a zero-avoiding prior $\tau^2 \sim$ inv-gamma$(0.1, 0.1)$. We sample one chain (3000 iterations) from the complete pooling model, eight chains each from centered and non-centered parameterization, stack the complete pooling and centered ones, and evaluate the prediction ability of the posterior inference using mean log

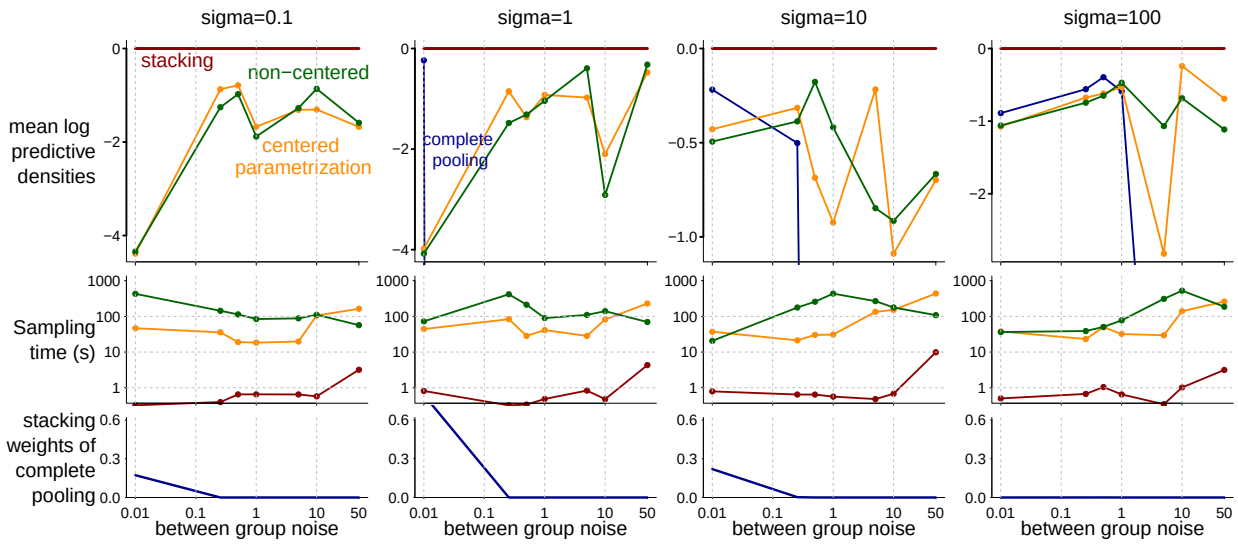


Figure 8.16: *We stack 8 parallel centered-parameterized chains and 1 complete pooling chain. The stacked average always has better test data performance than both centered and non-centered ones in all data configurations. The additional computation cost of stacking is minimal. Even when the complete pooling chain receives zero weight, stacking still helps remedy slow mixing of remaining chains and achieves better elpd than uniform mixing.*

predictive densities on $N_{\text{test}} = 300$ independent test data in each group. In the upper row of Figure 8.16, we place the stacking average as the baseline and extract its elpd from other parameterizations. The complete pooling model almost always has lpd so low that it does not even appear on the graph, and should never be used by itself. Instead of picking between the centered or and non-centered parameterization, the stacking estimate (red line) always has a larger log predictive density than the best of them. Such advantage is achieved at a negligible computation cost compared with sampling time (middle row). These patterns are robust under different prior and data configurations, and we have omitted similar outcomes when we tune J from 10 to 500 and for other zero-avoiding priors.

Lastly, in this example, stacking remedies both the incapability to sample in small τ regions, and between-chain-non-mixing in the centered parameterization. The last row of Figure 8.16 monitors stacking weights for the complete pooling chain. Even when it receives zero weight, the stack-weighted draws from centered parameterization are better than the uniform mixing of eight chains.

Stacking multi-run variational inference in a horseshoe regression. The regularized horseshoe prior (Piironen and Vehtari, 2017b,c) is an effective tool for Bayesian sparse regression. Denoting $y_{1:n}$ as a binary outcome and $x_{n\times D}$ as predictors, the logistic regression with a regularized horseshoe prior is,

$$\Pr(y_i = 1) = \text{logit}^{-1}(\beta_0 + \sum_{d=1}^{D} \beta_d x_{id}),\ i = 1,\dots,n, \quad \beta_d|\tau,\lambda,c \sim \text{normal}\left(0, \frac{\tau c \lambda_d}{(c^2+\tau^2\lambda_d^2)^{1/2}}\right),$$
$$c^2 \sim \text{Inv–Gamma}(\alpha,\beta), \quad \tau \sim \text{Cauchy}^+(0,1), \quad \lambda_d \sim \text{Cauchy}^+(0,1),\ d = 1,\dots,D.$$

Sampling from the exact posterior $p(\beta,\tau,c,\lambda|y)$ is computationally intensive and not scalable to big data. Unfortunately, mean-field variational inference (VI, Blei et al., 2017) which optimizes over the best mean-field Gaussian approximation to the joint posterior measured in KL divergence, behaves poorly on horseshoe regression. In particular, VI cannot capture the posterior multimodality (see examples in Yao et al., 2018b), which is a key aspect of the regularized horseshoe, a continuous counterpart of the spike-and-slab prior.

In general, the optimization problem in variational inference is not convex. Equipped with stochastic gradient descent, multiple runs of variational inference can return entirely different parameters. The common practice is to either select the best run based on the evidence lower bound (elbo) or test data performance. In the presence of posterior multimodality, the best that a normal approximation can do is to pick one mode, which in particular undermines the advantage of altering between no pooling and complete pooling of horseshoe regressions.

In next two experiments, we apply stacking to multiple runs of automatic variational inference (ADVI, Kucukelbir et al., 2017). In the k-th run, $k = 1,\dots,K$, we obtain S posterior approximation draws $\theta_{k1},\dots,\theta_{kS}$. We treat these as posterior samples, obtain the leave-one-out predictive densities, and use stacking to derive the optimal combination weights of all K runs.

Synthetic data. We first generate data from the model, $\Pr(y_i = 1) = \text{logit}^{-1}\left(\sum_{d=1}^{400} \beta_d x_{id}\right),\ i = 1,\dots,n = 40$. The design matrix X is normally distributed with shared featurewise components to

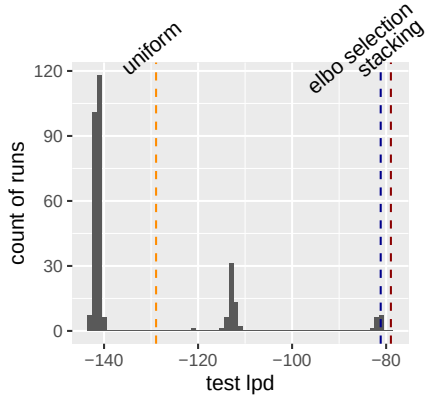


Figure 8.17: *Test data elpd among 300 runs of variational inference using synthetic data. Stacking over 300 runs achieves better prediction than any single run and also outperforms uniform mixing.*

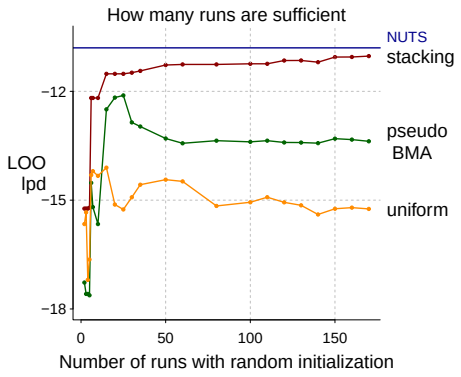


Figure 8.18: *Monitoring convergence for the leukemia example. Pseudo-BMA and uniform weighting have lower loo-lpd with more runs. Stacking is stable after 10 runs and gives a fit close to NUTS while requiring much less computation time.*

increase linear dependence. Of the 400 predictors, only the first three have nonzero coefficients $\beta_{1,2,3} = (3, 2, 1)$; this is the example discussed in Van Der Pas et al. (2014) and Piironen and Vehtari (2017c). We assess the model prediction on hold-out test data with size $n_{\text{test}} = 200$.

Figure 8.17 presents the test data log predictive densities among 300 ADVI runs with 10^5 stochastic gradient descent iterations each run. Stacking achieves better prediction than any single run and uniform mixing. Most of the runs have a low lpd, making the uniform reweighing undesired. The elbo selection selects the second best run (in test data lpd).

Leukemia classification. We consider regularized horseshoe logistic regression on the leukemia classification dataset. It contains 72 patients $y_i = 0$ or $1, 1 \leq i \leq 72$, and a large set of predictors consisting of 7128 gene features $x_{id}, 1 \leq d \leq 7128$.

In this section, we view HMC/NUTS sampling in Stan as the gold standard, which is slow (several hours per 1000 iterations) but mixes well in this dataset (Piironen and Vehtari, 2017c). We push the limit of variational inference by averaging 200 parallel ADVI runs with 10^5 stochastic gradient descent iterations, where each run takes less than one minute, but the approximation from any VI run is inaccurate.

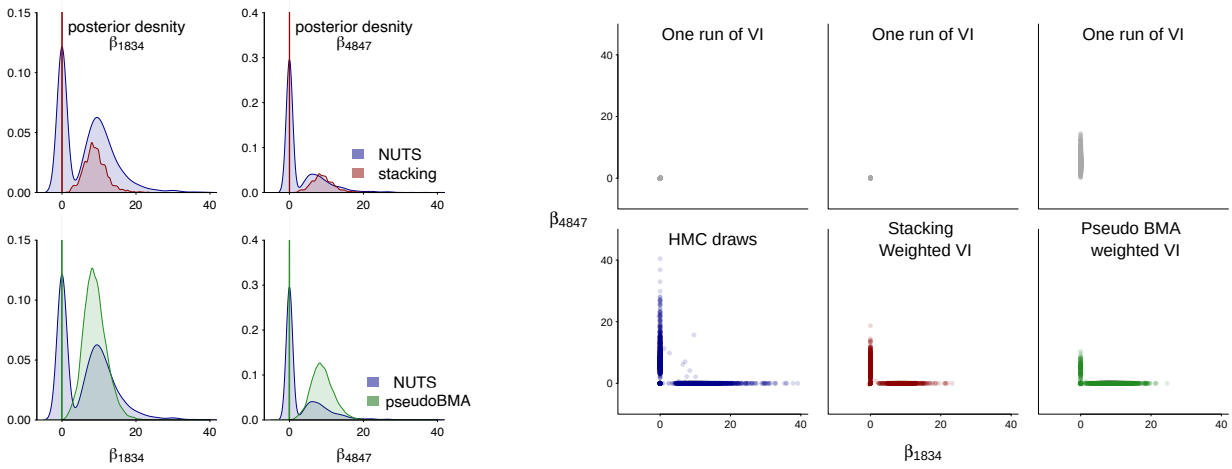


Figure 8.19: *The stacked VI posterior distribution matches HMC/NUTS draws reasonably well both marginally (left panel) and jointly (right panel) for the leukemia example, although individual runs are inaccurate. The graph displays two parameters β_{1834} and β_{4847} that have the largest absolute posterior means.*

Figure 8.18 displays the leave-one-out log predictive density of the combined distribution as a function of the number of runs to average, as previously described in (8.8). For stacking, there is a first jump at 5 runs, a second jump at roughly 10 runs, and then almost stable afterwards. For pseudo-BMA and uniform weighting, the loo elpd is worse with more runs, because VI is sensitive to initialization, and pseudo-BMA, BMA, and uniform weighting are sensitive to weak but duplicated runs (Yao et al., 2018a). Stacking achieves a much better leave-one-out lpd than all individual chains and other weighting methods, nearly comparable to HMC/NUTS. There is one caveat: because of the optimization procedure, the loo lpd of stacking likely overestimates its expected lpd.

To better evaluate how close the final inference is to the exact sampling, we visualize the stacked posterior VI draws of β_{1834} and β_{4847} (we pick these two variables which in our computation had the largest absolute posterior means as estimated with HMC/NUTS) in Figure 8.19. Stacked VI approximates the posterior well both marginally (left two columns) and jointly (right three columns). It captures the main shape: a spike concentrated at 0 and a slab part—a true spike in the stacked distribution might be even more appealing for interpretation. We also plot the joint distributions from three individual runs, all distant from the truth. Stacking recombines these

individual mean-field normal approximations, the mixture of enough of which can approximate any continuous distribution.

Finally as a caveat, the PSIS-loo approximation is applicable to VI under assumption that each VI optimum q_k locally matches the exact posterior p (up to a normalization constant c_k):

$$\exists \Theta_k \subset \Theta,\ q_k(\Theta_k) \approx 1, \quad s.t.\ \forall \theta \in \Theta_k,\ q_k(\theta) \approx c_k p(\theta|y), \tag{8.19}$$

which can be assessed by diagnostics in Yao et al. (2018b). In this example, it is implausible that (8.19) would exactly hold, but PSIS-loo still yields useful results. Alternatively, we can circumvent assumption (8.19), replace loo by a training-validation split, and perform stacking on the validation set, as shown in Section 8.8.

Bayesian neural networks. The posterior distribution of neural network parameters is well known to be often multimodal. We demonstrate stacking for such an example using the MNIST dataset, a collection of images of handwritten digits that are to be classified into their true labels, 0–9. We consider a two-layer neural network with tanh activation function:

$$\Pr(y_i = k) \propto \exp(\sum_{j=1}^{J} h_{ij}\beta_{jk} + \phi_k), \quad h_{ij} = \tanh(\sum_{m=1}^{M} x_{im}\alpha_{mj}), \quad i = 1, \ldots, n,\ k = 0, \ldots, 9.$$

where n is the sample size, J is the number of hidden nodes, and $M = 784$ is the input dimension. Making scalable Bayesian inference remains an open computation problem and beyond the scope of this chapter. To simplify the problem while keeping the pathological multimodality in the posterior distribution, we subsample $n = 1000$ training data from the labels $y = 1$ and 2 and set the number of hidden nodes $J = 40$. We use hierarchical priors, $\alpha \sim \text{normal}(0,\sigma_\alpha)$, $\beta \sim \text{normal}(0,\sigma_\beta)$, $\sigma_\alpha, \sigma_\beta \sim \text{normal}^+(0,3)$. Switching the order of hidden nodes does not change the predictive density. We eliminate the combinatoric non-identification in all other experiments in this section by constraining the order of β: $\beta_1 \geq \beta_2 \ldots, \geq \beta_J$.

We sample from the posterior distribution $p(\phi, \beta, \alpha|y, x)$ using 50 parallel HMC/NUTS chains

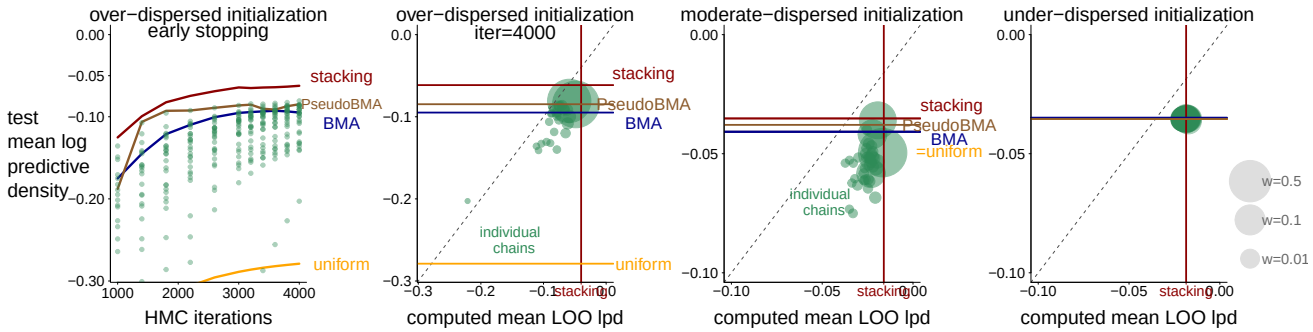


Figure 8.20: *(1): The test mean log predictive densities of early stopped chains. Stacking performs consistently better than single chains or other weighting methods. (2–4): The mean leave-one-out and test data log predictive densities of 50 individual chains (green dots), their stacking weights (size of the dot), and the test mean lpd of from four weighting strategies when fitting 40-hidden node neural network on MNIST. There were 4000 iterations per chain, and network parameters are initialized from uniform*$(-50, 50)$*,* $(-5, 5)$*, and* $(-0.001, 0.001)$*, respectively. Some individual changes in the overdispersed setting are out of lower-range and not shown.*

in Stan. In the right three panels in Figure 8.20, we present the posterior predictive performance of individual chains and combinations, evaluated by the mean log predictive densities on both leave-one-out data and an test data with $n_{\text{test}} = 2167$. The test score standard deviation is negligible. The initial values of unconstrained parameters in panels 2–4 are drawn from uniform$(-50, 50)$, $(-5, 5)$, and $(-0.001, 0.001)$, respectively. Each green dot stands for one chain, and the size of the dot reflects the chain weight in stacking (we rescale the size proportional to $w_{\text{stacking}}^{1/5}$ to manifest extremely small weights, see the legend on the right). Under an overdispersed initialization, the posterior inferences considerably diverge, and uniform weighting is jeopardized by "unlucky" chains, while stacking is not affected by a large number of bad chains. The PSIS-loo approximation does not accurately estimate the test performance (detected by large $\hat{k}$ diagnostics), but stacking still outperforms all other weighting strategies. Under the $(-0.001, 0.001)$ initialization, all 50 chains are essentially identical, and there is no gain from reweighing. In this experiment, a carefully tuned underdispersed initialization is the most efficient. However, choosing optimal starting values in general models remains difficult, whereas stacking is less sensitive to the initialization.

Early stopping is a commonly used ad hoc regularization method in neural networks (Vehtari et al., 2000). The leftmost column in Figure 8.20 demonstrates that we can stack early stopped

chains to achieve a prediction-power and computation-cost tradeoff. In the setting of 40 hidden nodes and overdispersed initialization, stacking is strictly better than the best single chain however early we stop. Stacking with 1500 HMC iterations is better than the best chain at iteration 4000. BMA and pseudo-BMA effectively choose just a single chain, and they and select the wrong chains at times. Uniform weighting is again the worst due to its sensitivity to bad initializations.

The existing literature on neural net ensembles advocate to *uniformly* average over all ensembles constructed by local MAPs found through stochastic gradient descent (Lakshminarayanan et al., 2017), bootstrap resampling (Osband et al., 2019), or varying priors (Pearce et al., 2020). Our experimental results show that inference from uniform weights is highly sensitive to starting points and can be especially disappointing under an overdispersed initialization. The approximate loo-based stacking sheds light on the benefit of post-inference multi-chain-reweighing in modern deeper neural networks. The additional optimization cost is tiny compared to the cost of model training. We leave question of scalability to modern Bayesian deep learning models to future investigation.

8.9 Discussion

Learn better epistemic uncertainty to expiate aleatoric misspecification. Uncertainty comes into inference and prediction through two sources: (a) due to finite amount of data, we learn the *epistemic* uncertainty of unknown parameter θ through the posterior distribution $p(\theta|y)$, and (b) due to either the stochastic nature of real world, even when θ is known, we represent the *aleatoric* uncertainty through the probabilistic forecast of next unseen outcome as $p(\tilde{y}|\theta, y)$. The final probabilistic prediction contains both of them via $p(\tilde{y}|y) = \int p(\tilde{y}|\theta, y)p(\theta|y)d\theta$.

Given a model, the epistemic uncertainty is mathematically well-defined though Bayesian inference, but will only be optimal under the true model and when averaging over the prior distribution. By being open-minded to model misspecification, the optimization (8.2) searches for the "best" probabilistic inference and uncertainty quantification with respect to a given utility function.

This chapter calls attention to post processing and re-calibrating Bayesian epistemic uncertainty.

Stacking reweighs separated component in the posterior density, while in general we can consider other transformations of the posterior draws such as location–scale shift, mixtures, and convolutions.

Bayesian inference is known to be poorly-calibrated under model misspecification (Gelman and Shalizi, 2013). In the context of model-selection and averaging, the marginal-likelihood-based "full-Bayes" approach produces over-confident prediction when none of the model is true (Clarke, 2003; Wong and Clarke, 2004; Clyde and Iversen, 2013; Yao et al., 2018a; Yang and Zhu, 2018; Yao, 2019; Oelrich et al., 2020), and therefore is not Bayes optimal (Le and Clarke, 2017).

The suboptimality of Bayesian posteriors does not mean we think Bayesian inference is wrong, but it does imply that there are tensions between a reckless application of Bayes rule under the wrong model and the Bayesian decision theory, and more generally, between Bayesian inference and Bayesian workflow. In the words of Gelman and Yao (2021), such tensions can only be resolved by considering Bayesian logic as a tool, a way of revealing inevitable misfits and incoherences in our model assumptions, rather than as an end in itself.

Chain-stacking as nonparametric inference. A parametric model $y|\theta \sim p(y|\theta)$, $\theta \sim p(\theta), \theta \in \Theta$ restricts the data generating mechanisms, which a priori are only supported at $\{p(y|\theta) \mid \theta \in \Theta\}$. The nonparametric Bayesian approach allows more flexible modeling that assigns a prior on a larger space but is subject to other challenges of prior constriction and computation.

Multi-chain stacking enriches Bayesian inference in the same way that nonparametric priors make models flexible. By allowing inference to depart from Bayes' rule, we identify and correct for model misspecification through stacked inference. The nonparametric aspect of stacking is also reflected by the unspecified number of mixture components, as conceptually an infinite mixture of simple distributions can approximate any continuous distribution. Of course, stacking cannot resolve all model misspecification since it uses parametric inferences as building blocks.

Chain-stacking as diagnostics. Besides improving on model predictions, multi-chain stacking can be used as a diagnostic tool.

First, uniformly identical stacking weight implies that parallel chains have mixed in overall

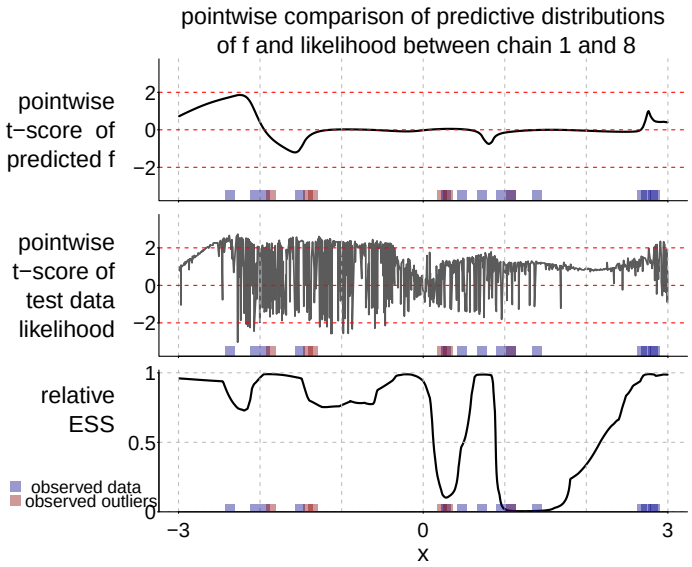


Figure 8.21: *Pointwise comparison of* $\mathbb{E}[\tilde{f}|y,\tilde{x}]$ *and log predictive densities* $\log p(\tilde{y}|\tilde{x},y)$ *at each* $\tilde{x}$ *from chain 1 and 8 in Figure 8.12. We compare two chains by a pointwise t test, where we plug in the effective sample size.*

predictive performance. In comparison, $\widehat{R}$ is only marginally diagnostic on parameters. One chain can be slightly but constantly better than the other, and such difference will be accumulated across all points. For example, Figure 8.21 compares the pointwise predictive distributions of chains 1 and 8 in Figure 8.12. For each point $\tilde{x}_i$, we compute the parameter estimation $\mathbb{E}(f|y,X,\tilde{x}_i) = \int f p(f|y,X,\tilde{x}_i)df\,d\sigma$ and the log predictive density $\log p(\tilde{y}_i|\tilde{x}_i) = \log \int p(\tilde{y}_i|f,\sigma)p(f|y,X,\tilde{x}_i)df\,d\sigma$, both using Monte Carlo draws from chain 1 and 8. We compare two Monte Carlo integrals by a t test, and adjust the sample autocorrelation by plugging in the estimated effective sample size (ESS) of the draws. In 84% of the test points $\tilde{x}_i$ uniformly distributed on $(-3,3)$, chain 8 has a higher pointwise predictive density than chain 1.

The distribution of log predictive densities often has a thicker tail than the distribution of individual parameters, thereby having a larger Monte Carlo variation and slower mixing rate. Even when all parameters are normally distributed in the posterior, the posterior log likelihood can be χ^2 distributed (see examples in Yao, 2019; Paananen et al., 2021). Compared with $\widehat{R}$, stacking can reveal subtle aspects of poor mixing among chains, offering a diagnostic that is targeted to prediction.

Second, we can use stacking to diagnose where the posterior geometry cause sampling issues. In the hierarchical model example, a nonzero complete-pooling chain ($\tau = 0$) indicates that the simulation using centered parameterization has not fully explored the basin around $\tau \approx 0$.

Lastly, stacking can diagnose interactions that have not been included in the model. When stacking reveals strong and persistent differences among chains, it signifies potential model misspecification. In particular, the data can be a mixture of several generating processes corresponding to different parameters (Kamary et al., 2019). We can expand to a hierarchical model as described in Section 8.3.

Stacking as part of Bayesian workflow. We view stacking of parallel chains as sitting on the boundary between black-box inference and a larger *Bayesian workflow* (Gabry et al., 2019; Gelman et al., 2020).

For an automatic inference algorithm, stacking enables accessible inference from non-mixing chains and a free enrichment of predictive distributions, which is especially relevant for repeated tasks where computation time is constrained.

For Bayesian workflow more generally, we recommend stacking in the model exploration phase, where we need to obtain *some* inference. Parallel computation can be running asynchronously—it may be that only some chains are running slowly—and stopping in the middle frees up computation and human time that can be reallocated to explorations of more models. In addition, non-uniform stacking weights when used in concert with trace plots and other diagnostic tools can help us understand where to focus that effort in an iterative way.

8.10 Software implementation

We demonstrate the implementation of multiple-chain stacking in the general-purpose Bayesian inference engine Stan (Stan Development Team, 2020). We use the Cauchy mixture model as an example. First save the following Stan file to `cauchy.stan`.

```
data {
int n;
vector[n] y;
}
```

```
parameters {
real mu;
}
model {
y ~ cauchy(mu, 1);
}
generated quantities {
vector[n] log_lik;
for (i in 1:n)
log_lik[i] = cauchy_lpdf(y[i]| mu, 1);
}
```

In the `generated quantities` block, we save `log_lik`: the log likelihood of each data point at each posterior draw. We generate data from a Cauchy mixture according to example (iii) in Figure 8.2, and sample from its posterior densities. Here is the R code:

```
library(rstan)
library(loo)
set.seed(100)
mu = c(-10,10)
n = 100
y = rep(NA, n)
p = 0.5
y[1:(n*p)] = rcauchy(n*(p),mu[1], 1)
y[(n*(p)+1):n] = rcauchy(n*(1-p),mu[2], 1)
K = 8
# Fit the model in stan
set.seed(100)
stan_fit = stan("cauchy.stan", data=list(n=n, y=y), chains=K, seed=100)
mu_sample = extract(stan_fit, permuted=F, pars="mu")[,,"mu"]
print(Rhat(mu_sample))
```

We are using eight parallel chains, and the resulted $\widehat{R} = 1.6$, clearly not mixing.

`chain_stack()` is a function to combine multiple chains in a Stan fit object, returned by `stan()`. It only require the whole model fit once, and save the point wise log likelihood in each iteration, called via `log_lik` here. The `chain_stack()` function uses the Stan optimizer (the default is L-BFGS), and its first time compiling takes up to a few minutes. `lambda` is the tuning parameter that controls the Dirichlet prior on stacking weights.

```
> library(devtools)
> source_url("https://github.com/yao-yl/Multimodal-stacking-code
/blob/master/chain_stacking.R?raw=TRUE")
> stan_model_object = stan_model("stacking_opt.stan")
> stack_obj=chain_stack(fits=stan_fit,lambda=1.0001,log_lik_char="log_lik")

Output:  Stacking 8 chains, with 100 data points and 1000 posterior draws;
using stan optimizer, max iterations = 1e+05
...done.
Total elapsed time for approximate LOO and stacking = 0.87 s
```

We can assess the reliability of the approximate leave-one-out using the $\hat{k}$ diagnostics. In this example, all pointwise $\hat{k}$ estimates (100 observations $\times$ 8 chains = 800 in total) are smaller than 0.5, indicating that the loo approximation is accurate in this example.

```
>  print_k(stack_obj)

Output:                Count Proportion
(-Inf, 0.5] (good)     800   1
(0.5, 0.7]  (ok)       0     0
(0.7, 1]    (bad)      0     0
(1, Inf)    (very bad) 0     0
```

We access the chain wights using

```
> chain_weights = stack_obj$chain_weights
```

Finally, we can use the weighted samples to calculate any posterior integral $\mathbb{E}_{\text{stacking}}(h(\mu)|y)$ as in (8.5). Here we compute $\Pr(\mu > 0|y)$: the total mass of positive values in the stacked inference.

```
> h = function(mu){mu>0}
> round(chain_weights %*% apply(h(mu_sample), 2, mean), digits=3)
[1] 0.523
```

Alternatively, we provide a quasi Monte Carlo based importance resampling function `mixture_draws()` that draws posterior samples form the stacked inference. This enables us to compute the same integral $\mathbb{E}_{\text{stacking}}[h(\mu) \mid y]$ using usual Monte Carlo methods:

```
> resampling=mixture_draws(individual_draws=mu_sample,weight=chain_weights)
> mean(h(resampling))
[1] 0.523
```

Chapter 9. Bayesian hierarchical stacking:

Some models are (somewhere) useful[1]

"The fountains mingle with the river, And the rivers with the ocean,
The winds of heaven mix for ever, With a sweet emotion;
Nothing in the world is single; All things meet and mingle.

—Percy Bysshe Shelley

9.1 Introduction

Statistical inference is conditional on the model, and a general challenge is how to make full use of multiple candidate models. Consider data $\mathcal{D} = (y_i \in \mathcal{Y}, x_i \in \mathcal{X})_{i=1}^n$, and K models $M_1, \ldots, M_k$, each having its own parameter vector $\theta_k \in \Theta_k$, likelihood, and prior. We fit each model and obtain posterior predictive distributions,

$$p(\tilde{y}|\tilde{x}, M_k) = \int_{\Theta_k} p(\tilde{y}|\tilde{x}, \theta_k, M_k) p(\theta_k|\{y_i, x_i\}_{i=1}^n, M_k)\, d\theta_k. \tag{9.1}$$

The model fit is judged by its expected predictive utility of future (out-of-sample) data $(\tilde{y}, \tilde{x}) \in \mathcal{Y} \times \mathcal{X}$, which generally have an unknown *true* joint density $p_t(\tilde{y}, \tilde{x})$. Model selection seeks the best model with the highest utility when averaged over $p_t(\tilde{y}, \tilde{x})$. Model averaging assigns models with weight $w_1, \ldots, w_K$ subject to a simplex constraint $\mathbf{w} \in \mathcal{S}_K = \{\mathbf{w} : \sum_{k=1}^K w_k = 1; w_k \in [0,1], \forall k\}$, and the future prediction is a linear mixture from individual models:

$$p(\tilde{y}|\tilde{x}, \mathbf{w}, \text{model averaging}) = \sum_{k=1}^K w_k p(\tilde{y}|\tilde{x}, M_k), \;\; \mathbf{w} \in \mathcal{S}_K. \tag{9.2}$$

[1]This chapter is a slight modified version of Yao et al. (2021b).

Stacking (Wolpert, 1992), among other ensemble-learners, has been successful for various prediction tasks. Yao et al. (2018a) applies the stacking idea to combine predictions from separate Bayesian inferences. The first step is to fit each individual model and evaluate the pointwise leave-one-out predictive density of each data point i under each model k:

$$p_{k,-i} = \int_{\Theta_k} p(y_i|\theta_k, x_i, M_k) p\left(\theta_k | M_k, \{(x_{i'}, y_{i'}) : i' \neq i\}\right) d\theta_k,$$

which in a Bayesian context we can approximate using posterior simulations and Pareto-smoothed importance sampling (Vehtari et al., 2017). Reusing data eliminates the need to model the unknown joint density $p_t(\tilde{y}, \tilde{x})$. The next step is to determine the vector[2] of weights $\mathbf{w} = (w_1, \ldots, w_K)$ that optimize the average log score of the stacked prediction,

$$\hat{\mathbf{w}}^{\text{stacking}} = \arg\max_{\mathbf{w}} \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} w_k p_{k,-i} \right), \text{ such that } \mathbf{w} \in \mathcal{S}_K. \tag{9.3}$$

However, the linear mixture (9.2) restricts an identical set of weight for all input x. We will later label this solution (9.3) as *complete-pooling stacking*. The present paper proposes *hierarchical stacking*, an approach that goes further in three ways:

1. Framing the estimation of the stacking weights as a Bayesian inference problem rather than a pure optimization problem. This in itself does not make much difference in the complete-pooling estimate (9.3) but is helpful in the later development.
2. Expanding to a hierarchical model in which the stacking weights can vary over the population. If the model predictors x take on J different values in the data, we can use Bayesian inference to estimate a $J \times K$ matrix of weights that partially pools the data both in row and column.
3. Further expanding to allow weights to vary as a function of continuous predictors. This idea generalizes the feature-weighted linear stacking (Sill et al., 2009) with a more flexible form and Bayesian hierarchical shrinkage.

There are two reasons we would like to consider input-dependent model weights. First, the

[2]We use the bold letter $\mathbf{w}$, or $\mathbf{w}(\cdot)$ to reflect that the weight is vector, or a vector of functions.

scoring rule measures the expected predictive performance averaged over $\tilde{x}$ and $\tilde{y}$, as the objective function in (9.3) divided by n is a consistent estimate of $\mathbb{E}_{\tilde{x},\tilde{y}} \log \left(\sum_{k=1}^{K} w_k p(\tilde{y}|\tilde{x}, \mathcal{D}, M_k) \right)$. But an overall good model fit does not ensure a good conditional prediction at a given location $\tilde{x} = \tilde{x}_0$, or under covariate shift when the distribution of input x in the observations differs from the population of interest. More importantly, different models can be good at explaining different regions in the input-response space, which is why model averaging can be a better solution to model selection. Even if we are only interested in the average performance, we can further improve model averaging by learning *where* a model is good so as to *locally* inflate its weight.

In Section 9.2, we develop detailed implementation of hierarchical stacking. We explain why it is legitimate to convert an optimization problem into a formal Bayesian model. With hierarchical shrinkage, we partially pool the stacking weights across data. By varying priors, hierarchical stacking includes classic stacking and selection as special cases. In Section 9.4, we generalize this approach to continuous input variables, other structured priors, and time-series and longitudinal data. We outline related work in Section 9.5. In Section 9.6, we evaluate the proposed method in several simulated and real-data examples, including a U.S. presidential election forecast. The proposed hierarchical stacking provides a Bayesian recipe for model averaging with input-dependent-weights and hierarchical regularization. It is beneficial for both improving the overall model fit, and the conditional local fit in small areas.

In the next chapter, we will turn heuristics from the previous paragraph into a rigorous learning bound, indicating the benefit from model selection to model averaging, and from complete-pooling model averaging to a local averaging that allows the model weights to vary in the population. The theoretical results characterize how the model list should be locally separated to be useful in model averaging and local model averaging.

9.2 Let the weight vary by input

The present paper generalizes the linear model averaging (9.2) to pointwise model averaging. The goal is to construct an input-dependent model weight function $\mathbf{w}(x) = (w_1(x), \ldots, w_K(x))$:

$\mathcal{X} \to \mathcal{S}_K$, and combine the predictive densities pointwisely by

$$p(\tilde{y}|\tilde{x}, \mathbf{w}(\cdot), \text{ pointwise averaging}) = \sum_{k=1}^{K} w_k(\tilde{x}) p(\tilde{y}|\tilde{x}, M_k), \text{ such that } \mathbf{w}(\cdot) \in \mathcal{S}_K^{\mathcal{X}}. \tag{9.4}$$

If the input is discrete and has finite categories, one naïve estimation of the pointwise optimal weight is to run complete-pooling stacking (9.3) separately on each category, which we will label *no-pooling stacking*. The no-pooling procedure generally has a larger variance and overfits the data.

From a Bayesian perspective, it is natural to compromise between unpooled and completely pooled procedures by a hierarchical model. Given some hierarchical prior $p^{\text{prior}}(\cdot)$, we define the posterior distribution of the stacking weights $w \in \mathcal{S}_K^{\mathcal{X}}$ through the usual likelihood-prior protocol:

$$\log p(\mathbf{w}(\cdot)|\mathcal{D}) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} w_k(x_i) p_{k,-i} \right) + \log p^{\text{prior}}(\mathbf{w}) + \text{constant}, \quad \mathbf{w}(\cdot) \in \mathcal{S}_K^{\mathcal{X}}. \tag{9.5}$$

The final estimate of the pointwise stacking weight used in (9.4) is then the posterior mean from this joint density $\mathbb{E}(\mathbf{w}(\cdot)|\mathcal{D})$. We call this approach *hierarchical stacking*.

Complete-pooling and no-pooling stacking. For notational consistency, we rewrite the input variables into two groups (x, z), where x are variables on which the model weight $w(x)$ depends during model averaging (9.4), and z are all remaining input variables.

To start, we consider x to be discrete and has $J < \infty$ categories, $x = 1, \dots J$. We will extend to continuous and hybrid x later. The input varying stacking weight function is parameterized by a $J \times K$ matrix $\{w_{jk}\} \in \mathcal{S}_K^J$: Each row of the matrix is an element of the length-K simplex. The k-th model in cell j has the weight $w_k(x_i) = w_{jk}, \forall x_i = j$. We fit each individual model M_k to all observed data $\mathcal{D} = (x_i, z_i, y_i)_{i=1}^n$ and obtain pointwise leave-one-out cross-validated log predictive densities:

$$p_{k,-i} \coloneqq \int_{\Theta_k} p(y_i|\theta_k, x_i, z_i, M_k) p(\theta_k|\{(x_l, y_l, z_l) : l \neq i\}, M_k) d\theta_k. \tag{9.6}$$

Same as in complete-pooling stacking, here we avoid refitting each model n times, and instead

use the Pareto smoothed importance sampling (PSIS, Vehtari et al., 2017, 2019b) to approximate $\{p_{k,-i}\}_{i=1}^n$ from one-time-fit posterior draws $p(\theta_k|M_k, \mathcal{D})$. The cost of such approximate leave-one-out cross validation is often negligible compared with individual model fitting.

To optimize the expected predictive performance of the pointwisely combined model averaging, we can maximize the leave-one-out predictive density

$$\max_{\mathbf{w}(\cdot)} \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} w_k(x_i) p_{k,-i} \right). \tag{9.7}$$

On one extreme, the complete-pooling stacking (9.3) solves optimization (9.7) subject to a constant constraint $w_k(x) = w_k(x'), \forall k, x, x'$. On the other extreme, no-pooling stacking maximizes this objective function (9.7) without extra constraint other than the row-simplex-condition, which amounts to separately solving complete-pooling stacking (9.3) on each input cell $\mathcal{D}_j = \{(x_i, z_i, y_i) : x_i = j\}$.

If there are a large number of repeated measurements in each cell, $n_j := ||\{i : x_i = j\}|| \to \infty$, then $1/n_j \sum_{i:x_i=j} \log \sum_{k=1}^K p_{k,-i}$ becomes a reasonable estimate of the conditional log predictive density $\int_{\mathcal{Y}} p_t(\tilde{y}|\tilde{x} = j) \log p(\tilde{y}|\tilde{x}, M_k) d\tilde{y}$, with convergence rate $\sqrt{n_j}$, and therefore, no-pooling stacking becomes asymptotically optimal among all cell-wise combination weights. For finite sample size, because the cell size is smaller than total sample size, we would expect a larger variance in no-pooling stacking than in complete-pooling stacking. Moreover, the cell sizes are often not balanced, which entails a large noise of no-pooling stacking weight in small cells.

9.3 Bayesian inference for stacking weights

Vanilla (optimization-based) stacking (9.3) is justified by *Bayesian decision theory*: the expected log predictive density of the combined model $\mathbb{E}_{\tilde{y}} \log \left(\sum_{k=1}^K w_k p(\tilde{y}|M_k) \right)$ is estimated by leave-one-out $1/n \sum_{i=1}^n \log \left(\sum_{k=1}^K w_k p_{k,-i} \right)$. The point optimum $\hat{\mathbf{w}}$ asymptotically maximizes the expected utility (Le and Clarke, 2017), hence is an M_*-optimal decision in terms of Vehtari and Ojanen (2012).

To fold stacking into a *Bayesian inference problem*, we want to treat the objective function in (9.7) as a log likelihood with parameter $\mathbf{w}$. After integrating out individual-model-specific parameters θ_k such that $p(y|x, M_k)$ is given, the outcomes y_i at input location x_i in the combined model have densities $p(y_i|x_i, w_k(x_i)) = \sum_{k=1}^{K} w_k(x_i)p(y_i|x_i, M_k)$, which implies a joint log likelihood: $\sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} w_k(x_i)p(y_i|x_i, M_k)\right)$. But this procedure has used data twice—in other practices, data are often used twice to pick prior, whereas here data are used twice to pick likelihood.

We use a two-stage estimation procedure to avoid reusing data. Assuming a hypothetically provided holdout dataset $\mathcal{D}'$ of the same size and identical distribution as observations $\mathcal{D} = \{y_i, x_i\}_{i=1}^{n}$, we can use $\mathcal{D}'$ to fit the individual model first and compute $\tilde{p}(y_i|x_i, M_k, \mathcal{D}') = \int p(y_i|x_i, M_k, \theta_k)p(\theta_k|M_k, \mathcal{D}')d\theta_k$. In the second stage we plug in the observed y_i, x_i, and obtain the pointwise full likelihood $p(y_i|\,\mathbf{w}, \mathcal{D}', x_i) = \sum_{k=1}^{K} w_k(x_i)\tilde{p}(y_i|x_i, M_k, \mathcal{D}')$.

Now in lack of holdout data $\mathcal{D}'$, the leave-i-th-observation-out predictive density $p_{k,-i}$ is a consistent estimate of the pointwise out-of-sample predictive density $\mathbb{E}_{\mathcal{D}'}\left(\tilde{p}(y_i|x_i, M_k, \mathcal{D}')\right)$. By plugging it into the two-stage log likelihood and integrating out the unobserved holdout data $\mathcal{D}'$, we get a profile likelihood

$$p(y_i|\,\mathbf{w}, x_i) \coloneqq \mathbb{E}_{\mathcal{D}'}\left(p(y_i, |\,\mathbf{w}, x_i, \mathcal{D}')\right) = \sum_{k=1}^{K} w_k(x_i)\,\mathbb{E}_{\mathcal{D}'}\left(p(y_i|x_i, M_k, \mathcal{D}')\right) \approx \sum_{k=1}^{K} w_k(x_i)p_{k,-i}.$$

Summing over y_i arrives at $\log\left(p(\mathcal{D}|\,\mathbf{w})\right) \approx \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} w_k(x_i)p_{k,-i}\right)$. This log likelihood coincides with the no-pooling optimization objective function (9.7).

To integrate out the hypothetical data $\mathcal{D}'$ is related to the idea of marginal data augmentation (Meng and van Dyk, 1999). Polson and Scott (2011) took a similar approach to convert the optimization-based support vector machine into a Bayesian inference.

Hierarchical stacking: discrete inputs. The log posterior density of hierarchical stacking model (9.5) contains the log likelihood defined above $\sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} w_k(x_i)p_{k,-i}\right)$, and a prior distribution on the weight matrix $\mathbf{w} = \{w_{jk}\} \in \mathcal{S}_K^J$, which we specify in the following.

We first take a softmax transformation that bijectively converts the simplex matrix space $\mathcal{S}_K^J$ to unconstrained space $\mathbb{R}^{J(K-1)}$:

$$w_{jk} = \frac{\exp(\alpha_{jk})}{\sum_{k=1}^{K} \exp(\alpha_{jk})},\ 1 \le k \le K-1,\ 1 \le j \le J; \qquad \alpha_{jK} = 0,\ 1 \le j \le J. \tag{9.8}$$

$\alpha_{jk} \in \mathbb{R}$ is interpreted as the log odds ratio of model k with reference to M_K in cell j.

We propose a normal hierarchical prior on the unconstrained model weights $(\alpha_{jk})_{k=1}^{K-1}$ conditional on hyperparameters $\mu \in \mathbb{R}^{K-1}$ and $\sigma \in \mathbb{R}_+^{K-1}$,

$$\text{prior}: \quad \alpha_{jk} \mid \mu_k, \sigma_k \sim \text{normal}(\mu_k, \sigma_k),\ k = 1, \ldots, K-1,\ j = 1, \ldots, J. \tag{9.9}$$

The prior partially pools unconstrained weights toward the shared mean $(\mu_1, \ldots, \mu_{K-1})$. The shrinkage effect depends on both the cell sample size n_j (how strong the likelihood is in cell j), and the model-specific σ_k (how much across-cell discrepancy is allowed in model k). If μ and σ are given constants, and if the posterior distribution is summarized by its mode, then hierarchical stacking contains two special cases:

- no-pooling stacking by a flat prior $\sigma_k \to \infty,\ k = 1, \ldots, K-1$.
- complete-pooling stacking by a concentration prior $\sigma_k \to 0,\ k = 1, \ldots, K-1$.

It is possible to derive other structured priors. For example, a sparse prior (e.g., Heiner et al., 2019) on simplex $(w_{j1}, \ldots, w_{jK})$ will enforce a cell-wise selection.

Instead of choosing fixed values, we view μ and σ as hyperparameters and aim for a full Bayesian solution: to describe the uncertainty of all parameters by their joint posterior distribution $p(\alpha, \mu, \sigma | \mathcal{D})$, letting the data to tell how much regularization is desired.

To accomplish this Bayesian inference, we assign a hyperprior to (μ, σ):

$$\text{hyperprior}: \quad \mu_k \sim \text{normal}(\mu_0, \tau_\mu), \quad \sigma_k \sim \text{normal}^+(0, \tau_\sigma), \quad k = 1, \ldots, K-1, \tag{9.10}$$

where $\text{normal}^+(0, \tau_\sigma)$ stands for the half-normal distribution supported on $[0, \infty)$ with scale param-

eter τ_σ.

Putting the pieces (9.5), (9.9), (9.10) together, up to a normalization constant that has been omitted, we attain a joint posterior density of all free parameters $\alpha \in \mathbb{R}^{J\times K}, \mu \in \mathbb{R}^{K-1}, \sigma \in \mathbb{R}_+^{K-1}$:

$$\log p(\alpha,\mu,\sigma|\mathcal{D}) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} w_k(x_i) p_{k,-i}\right) + \sum_{k=1}^{K-1}\sum_{j=1}^{J} \log p^{\text{prior}}\left(\alpha_{jk}|\mu_k,\sigma_k\right) + \sum_{k=1}^{K-1} \log p^{\substack{\text{hyper}\\ \text{prior}}}(\mu_k,\sigma_k). \quad (9.11)$$

Unlike complete and no-pooling stacking, which are typically solved by optimization, the maximum a posteriori (MAP) estimate of (9.11) is not meaningful: the mode is attained at the complete-pooling subspace $\alpha_{jk} = \mu_k, \sigma_k = 0, \forall j,k$, on which the joint density is positive infinity. Instead, we sample (α,μ,σ) from this joint density (9.11) using Markov chain Monte Carlo (MCMC) methods and compute the Monte Carlo mean of posterior draws $\overline{w}_{jk}$, which we will call *hierarchical stacking* weights.

The final posterior predictive density of outcome $\tilde{y}$ at any input location $(\tilde{x},\tilde{z})$ is

$$\text{final predictions}: p(\tilde{y}|\tilde{x},\tilde{z},\mathcal{D}) = \sum_{k=1}^{K} \overline{w}_k(\tilde{x}) \int_{\Theta_k} p(\tilde{y}|\tilde{x},\tilde{z},\theta_k,M_k) p(\theta_k|M_k,\mathcal{D}) d\theta_k. \quad (9.12)$$

Using a point estimate $\overline{w}_{jk}$ is not a waste of the joint simulation draws. Because equation (9.12) is a linear expression on w_k, and because of the linearity of expectation, using $\overline{\mathbf{w}}$ is as good as using all simulation draws. Nonetheless, for the purpose of post-processing, approximate cross validation, and extra model check and comparison, we will use all posterior simulation draws; see discussion in Section 9.7.

9.4 Hierarchical stacking: continuous and hybrid inputs

The next step is to include more structure in the weights, which could correspond to regression for continuous predictors, nonexchangeable models for nested or crossed grouping factors,

nonparametric prior, or combinations of these.

Additive model. Hierarchical stacking is not limited to discrete cell-divider x. When the input x is continuous or hybrid, one extension is to model the unconstrained weights additively:

$$\begin{aligned} w_{1:K}(x) &= \text{softmax}(w^*_{1:K}(x)), \\ w^*_k(x) &= \mu_k + \sum_{m=1}^{M} \alpha_{mk} f_m(x),\ k \le K-1,\ w^*_K(x) = 0, \end{aligned} \tag{9.13}$$

where $\{f_m : \mathcal{X} \to \mathbb{R}\}$ are M distinct features. Here we have already extracted the prior mean μ_k, representing the "average" weight of model k in the unconstrained space. The discrete model (9.9) is now equivalent to letting $f_m(x) = \mathbb{1}(x = m)$ for $m = 1, \ldots, J$. We may still use the basic prior (9.9) and hyperprior (9.10):

$$\alpha_{mk} \mid \sigma_k \sim \text{normal}(0, \sigma_k), \quad \mu_k \sim \text{normal}(\mu_0, \tau_\mu), \quad \sigma_k \sim \text{normal}^+(0, \tau_\sigma). \tag{9.14}$$

We provide `Stan` (Stan Development Team, 2020) code for this additive model and discuss practical hyperparameter choice in Appendix 9.8.

Because the main motivation of our paper is to convert the one-fit-all model-averaging algorithm into open-ended Bayesian modeling, the basic shrinkage prior above should be viewed as a starting point for model building and improvement. Without trying to exhaust all possible variants, we list a few useful prior structures:

- *Grouped hierarchical prior.* The basic model (9.14) is limited to have a same regularization σ_k for all α_{mk}. When the features $f_m(x)$ are grouped (e.g., f_m are dummy variable from two discrete inputs; states are grouped in regions), we achieve group specific shrinkage by replacing (9.14) by

$$\alpha_{mk} \mid \mu_k, \sigma_{gk} \sim \text{normal}(0, \sigma_{g[m]k}),\ \sigma_{gk} \sim \text{normal}^+(0, \tau_\sigma),\ \mu_k \sim \text{normal}(\mu_0, \tau_\mu),$$

where $g[m] = 1, \dots G$ is the group index of feature m.

- *Feature-model decomposition.* Alternatively we can learn feature-dependent regularization by

$$\alpha_{mk} \mid \mu_k, \sigma_k, \lambda_m \sim \text{normal}(0, \sigma_k \lambda_m),\ \lambda_m \sim \text{InvGamma}(a, b),\ \sigma_k \sim \text{normal}^+(0, \tau_\sigma).$$

- *Prior correlation.* For discrete cells, we would like to incorporate prior knowledge of the group-correlation. For example in election forecast (Section 9.6), we have a rough sense of some states being demographically close, and would expect a similar model weights therein. To this end, we calculate a prior correlation matrix $\Omega_{J\times J}$ from various sources of state level historical data, and replace the independent prior (9.9) by a multivariate normal (MVN) distribution,

$$(\alpha_{1k}, \dots, \alpha_{jk}) \mid \sigma, \Omega, \mu \sim \text{MVN}\left((\mu_k, \dots, \mu_k), \text{diag}(\sigma_k) \times \Omega\right). \tag{9.15}$$

  The prior correlation is especially useful to stabilize stacking weights in small cells.

- *Crude approximation of input density.* When applying the basic model (9.13) to continuous inputs $x = (x^1, \dots, x^D) \in \mathbb{R}^D$, instead of a direct linear regression $f_d(x) = x^d$, we recommend a coordinate-wise ReLU-typed transformation:

$$\{f : f_{2d-1}(x) = (x^d - \text{med}(x^d))_+,\ \ f_{2d}(x) = (\text{med}(x^d) - x^d)_-,\ d \le D\}, \tag{9.16}$$

  where $\text{med}(x^d)$ is the sample median of x^d. The pointwise model predictive performance typically relies on the training density $P_X^{\text{train}}(\tilde{x})$: The more training data seen nearby, the better predictions. The feature (9.16) is designed to be a crude approximation of log marginal input densities.

Choice of features and exploratory data analysis. Choosing the division of (x, z) in discrete inputs is now a more general problem on how to construct features $f_m(x)$, or a variable selection problem in a regression (9.13). In ordinary statistical modeling, we often start variable selection by exploratory data analysis. Here we cannot directly associate model weights w_{ki} with observable quantities. Nevertheless, we can use the paired pointwise log predictive density difference $\Delta_{ki} = (\log p_{k,-i} - \log p_{K,-i})$ as an exploratory approximation to the trend of $\alpha_k(x_i)$. A scatter plot of Δ_{ki} against x may suggest which margin of x is likely important. For example, the dependence of Δ_{ki} on whether x_i is in the bulk or tail is an evidence for our previous recommendation of the rectified features.

As more variables x are allowed to vary in the stacking model, model averaging is more prone to over-fitting. Pointwise stacking typically has a large noise-to-signal ratio not only due to model similarity, but also a high variance of pointwise model evaluation: the approximate leave-one-out cross validation possesses Monte Carlo errors; even if we run exact leave-one-out, or use an independent validation set in lieu of leave-one-out, we only observe one y_i for one x_i (if x is continuous) such that $\log p_{k,-i}$ is at best an one-sample-estimate of $\mathbb{E}_{\tilde{y}|x_i}(\log p(\tilde{y}|x_i, M_k))$ with non-vanishing variance. If $f_m(\cdot)$ is flexible enough, then the sample optimum of no-pooling stacking (9.7) always degenerates to pointwise model selection that pointwisely picks the model that "best" fits current realization of y_i: $w_{\arg\max_k p_{k,-i}}(x_i) = 1$, which is purely over-fitting.

Even in companion with hierarchical priors, we do not expect to include too many features on which stacking weights depend on. In our experiments, an additive model with discrete variables and rectified continuous variables without interaction is often adequate. After standardizing all features such that $\mathrm{Var}(f_m(x)) = 1$, we typically use a generic informative prior setting $\tau_\mu = \tau_\sigma = 1$ in experiments. With a moderate or large number of features/cells, M, it is sensible to scale the hyperprior $\tau_\sigma = O(\sqrt{1/M})$, or adopt other feature-wise shrinkage priors such as horseshoe for better regularization.

Gaussian process prior. An alternative way to generalize both the discrete prior in Section 9.3 and the prior correlation (9.15) is Gaussian process priors. To this end we need $K-1$ covariance kernels $\mathcal{K}_1, \ldots, \mathcal{K}_{K-1}$, and place priors on the unconstrained weight $\alpha_k(x)$, viewed as an $\mathcal{X} \to \mathbb{R}$ function: $\alpha_k(x) \sim \mathcal{GP}(\mu_k, \mathcal{K}_k(x))$. The discrete prior is a special case of a Gaussian process via a zero-one kernel $\mathcal{K}_k(x_i, x_j) = \sigma_k \mathbb{1}(x_i = x_j)$. Due to the previously discussed measurement error and the preference on stronger regularization for continuous x, we recommend simple exponentiated quadratic kernels $\mathcal{K}_k(x_i, x_j) = a_k \exp(-\left((x_i - x_j)/\rho_k\right)^2)$ with an informative hyperprior that avoids too small or too big length-scale ρ_k, and too big a_k. We present an example in Section 9.6.

Time series and longitudinal data. Hierarchical stacking can easily extend to time series and longitudinal data. Consider a time series dataset where outcomes y_i come sequentially in time $0 \le t_i \le T$. The joint likelihood is not exchangeable, but still factorizable via $p(y_{1:n}|\theta) = \prod_{i=1}^n p(y_i|\theta, y_{1:(i-1)})$. Therefore, assuming some stationary condition, we can approximate the expected log predictive densities of the next-unit unseen outcome by historical average of one-unit-ahead log predictive densities, defined by

$$p_{k,-i} \coloneqq \int_{\Theta_k} p(y_i|x_i, y_{1:(i-1)}, x_{1:(i-1)}, \theta_k, M_k) p(\theta_k|y_{1:(n-1)}, x_{1:(n-1)}) d\theta_k.$$

In hierarchical stacking, we only need to replace the regular leave-one-out predictive density (9.6) by this redefined $p_{k,-i}$, and run hierarchical stacking (9.11) as usual. Using importance sampling based approximation (Bürkner et al., 2020), we also make efficient computation without the need to fit each model n times.

If we worry about time series being non-stationary, we can reweight the likelihood in (9.11) by a non-decreasing sequence π_i: $n \sum_{i=1}^n \left(\pi_i \log\left(\sum_{k=1}^K w_k(x_i) p_{k,-i}\right)\right) / \sum_{i=1}^n \pi_i$, so as to emphasize more recent dates. For example, $\pi_i = 1 + \gamma - (1 - t_i/T)^2$, where a fixed parameter $\gamma > 0$ determines how much influence early data has. By appending $x \coloneqq (x, t)$, the stacking weight can vary across the time variable, too.

In Section 9.6, we present an election example with longitudinal polling data (40 weeks time

series $\times$ 50 states). For the i-th poll (already ordered by date), we encode state index into input $x_i = 1, \ldots, 50$, all other poll-specific variables z_i, data t_i, and poll outcome y_i. We compute the one-week-ahead predictive density $p_{k,-i} \coloneqq \int p(y_i|x_i, z_i, \mathcal{D}_{-i}, M_k) p(\theta_k|\mathcal{D}_{-i}, M_k) d\theta_k$ where the dataset $\mathcal{D}_{-i} = \{(y_l, x_l, z_l) : t_l \leq t_i - 7\}$ contains polls from all states up to one week before date t_i.

Stacking under covariate shift. So far we have adopted an IID view: the training and out-of-sample data are from the same distribution. Yet another appealing property of hierarchical stacking is its immunity to *covariate shift* (Shimodaira, 2000), a ubiquitous problem in non-representative sample survey, data-dependent collection, causal inference and many other areas.

If the distribution of inputs x in the training sample, $p_X^{\text{train}}(\cdot)$, differs from these predictors' distribution in the population of interest, $p_X^{\text{pop}}(\cdot)$ (p_X^{pop} is absolutely continuous with respect to p_X^{train}), and if $p(z|x)$ and $p(y|x, z)$ remain invariant, then we do *not* need to adjust weight estimate from (9.11), because it has already been pointwisely optimized.

By contrast, complete-pooling stacking is aimed at average risk. Under covariate shift, the sample mean of leave-one-out score in the k-th model, $\frac{1}{n}\sum_{i=1}^n \log p(\tilde{y}|\tilde{x}, M_k)$, is no longer a consistent estimate of population elpd. To adjust, we can run importance sampling (Sugiyama and Müller, 2005; Sugiyama et al., 2007; Yao et al., 2018a) and reweigh the i-th term in the objective (9.3) proportional to the inverse probability ratio $p_X^{\text{pop}}(x_i)/p_X^{\text{train}}(x_i)$. Even in the ideal situation when both p_X^{pop} and p_X^{train} are known, the importance weighted sum has in general larger or even infinite variance (Vehtari et al., 2019b), thereby turning down the effective sample size and convergence rate in complete-pooling stacking (toward its population optima (10.2)). When p_X^{train} is unknown, the covariate reweighting is more complex while hierarchical stacking circumvents the need of explicit modeling of p_X^{train}.

When we are interested only at one fixed input location $p_X^{\text{pop}}(\tilde{x}) = \delta(x_0)$, hierarchical stacking is ready for *conditional* predictions, whereas no-pooling stacking and reweighed-complete-pooling stacking effectively discard all $x_i \neq x_0$ training data in their objectives, especially a drawback when x_0 is rarely observed in the sample.

We may combine hierarchical stacking and poststratification to estimate the overall average risk. Suppose observations are from an online survey and we identify that the survey has selection bias on certain demographic variables x, such as age, gender and education, we can then fit multiple models and add hierarchical stacking to combine models. Finally we use poststratification weight $p^{\text{pop}}_{x=j}$ to compute the average log predictive density in the population $\sum_{j=1}^{J} p^{\text{pop}}_{X}(x=j)(\sum_{i:x_i=j} \log \sum_k (\hat{w}(x_i) p(y_i|D_{-i}, M_k))/\sum_{i:x_i=j} 1)$. Although it is plausible to further reweigh the hierarchical-stacking likelihood in order to achieve double robustness, here as per the protocol of multilevel modeling (Gelman and Little, 1997), we recommend poststratification only after model inference and hierarchical stacking.

Generally, a data-dependent-design is conditionally ignoble if all relevant covariates are included in the model and modeled properly. In our hierarchical stacking context, if the only goal is to adjust for the covariate-shift, we conjecture that the "most efficient" hierarchical stacking weight should only depend on the input through the propensity score $f(x) = \Pr(\text{test}|x)$: the probability of the an input location belonging to the test data. Our proposed method might be especially useful in causal inference, which we will leave for future investigation.

As a caveat, this algorithm-wise invariance does not ensure that the stacked prediction is good for the population of interest: if all individual models fit the test data terribly, model averaging cannot rescue them; when propensity score in the last paragraph is not properly estimated, the average treatment effect derived from hierarchical stacking cannot be right. Perhaps an analogy would be that the logistic regression is invariant under a retrospective (case-control) and a prospective (cohort) design, but it does not ensure that the logistic regression always yields good predictions, because the actual data does not necessarily have a logistic link function or a linear response.

For the task of averaging-models-with-covariate-shift, one open question is to compare the following 2×2 combination of methods:

(a) each individual model likelihood is pointwisely importance-weighted,

(b) each individual model is itself a hierarchical (multilevel) model that includes all relevant covariates,

×

(i) importance-weighted stacking,

(ii) hierarchical stacking.

Any one component appears a sensible adjustment for covariate-shift. Do we have double robustness when combining them? What is the best combination?

9.5 Related literature

Stacking (Wolpert, 1992; Breiman, 1996b; LeBlanc and Tibshirani, 1996), or what we call *complete-pooling stacking* in this chapter, has long been a popular method to combine learning algorithms, and has been advocated for averaging Bayesian models (Clarke, 2003; Clyde and Iversen, 2013; Le and Clarke, 2017; Yao et al., 2018a). This simple approach has been applied in various areas such as recommendation system, epidemiology (Bhatt et al., 2017), network modeling (Ghasemian et al., 2020), and post-processing in Monte Carlo computation (Tracey and Wolpert, 2016; Yao et al., 2020b). Stacking can be equipped with any scoring rules, while the present chapter focuses on the logarithm score by default.

Our theory investigation in Section 10.2 is inspired by the discussion of how to choose candidate models by Clarke (2003) and Le and Clarke (2017). In L^2 loss stacking, they recommended "independent" models in terms of posterior point predictions $(\mathbb{E}(\tilde{y}|\tilde{x}, M_1), \ldots, \mathbb{E}(\tilde{y}|\tilde{x}, M_K))$ being independent. When combining Bayesian predictive distributions, the correlations of the posterior predictive mean is not enough to summarize the relation between predictive distributions (Pirš and Štrumbelj, 2019), hence we consider the local separation condition instead.

Allowing a heterogeneous stacking model weight that changes with input x is not a new idea. Feature-weighted linear stacking (Sill et al., 2009) constructs data-varying model weights of the k-the model by $w_k(x) = \sum_{m=1}^{M} \alpha_{km} f_m(x)$, and α_{km} optimizes the L^2 loss of the point predictions of the weighted model. This is similar to the likelihood term of our additive model specification in Section 9.4, except we model the unconstrained weights. The direct least-square optimization solution from feature-weighted linear stacking is what we label *no-pooling stacking*.

It is also not a new idea to add regularization and optimize the penalized loss function. For L^2 loss stacking, Breiman (1996b) advocated non-negative constraints. In the context of combining Bayesian predictive densities, a simplex constraint is necessary. Reid and Grudic (2009) investigated to add L^1 or L^2 penalty, $-\lambda||w||_1$ or $-\lambda||w||_2$, into complete-pooling stacking objective (9.3). Yao et al. (2020b) assigned a Dirichlet(λ), $\lambda > 1$ prior to the complete-pooling stacking weight vector w to ensure strict concavity of the objective function. Sill et al. (2009) mentioned the use of L^2 penalization in feature-weighted linear stacking, which is equivalent to setting a fixed prior for all free parameters $\alpha_{km} \sim \text{normal}(0, \tau), \forall k, m$, whose solution path connects between uniform weighing and no-pooling stacking by tuning τ. All of these schemes are shown to reduce over-fitting with appropriate amount of regularization, while the tuning is computation intensive. In particular, each stacking run is built upon one layer of cross validation to compute the expected pointwise score in each model $p_{k,-i}$, and this extra tuning would require to fit each model $n(n-1)$ times for each tuning parameter value evaluation if both done in exact leave-one-out way. Fushiki (2020) approximated this double cross validation for L^2 loss complete-pooling stacking with L^2 penalty on w, beyond which there was no general efficient approximation.

Hierarchical stacking treats $\{\mu_k\}$ and $\{\sigma_k\}$ as parameters and sample them from the joint density. Such hierarchy could be approximated by using L^2 penalized point estimate with a different tuning parameter in each model, and tune all parameters ($\{\sigma_k\}_{k=1}^{K-1}$ for the basic model, or $\{\sigma_{mk}\}_{m=1,k=1}^{M,\ K-1}$ for the product model). But then this intensive tuning is equivalent to approximate the Type-II MAP of hierarchical stacking in an inefficient grid-searching fashion (in contrast to gradient based MCMC).

Another popular family of regularization in stacking enforces sparse weights (e.g., Zhang and Zhou, 2011; Şen and Erdogan, 2013; Yang and Dunson, 2014), which include sparse and grouped sparse priors on the unconstrained weights, and sparse Dirichlet prior on simplex weights. The goal is that only a limited number of models are expressed. From our discussion in Chapter 10, all models are somewhere useful, hence we are not aimed for model sparsity—The concavity of log scoring rules implicitly resists sparsity; The posterior mean of hierarchical stacking weights w_{jk}

is in general is never sparse. Nevertheless, when sparsity is of a concern for memory saving or interpretability, we can run hierarchical stacking first and then apply projection predictive variable selection (Piironen and Vehtari, 2017a) afterwards to the posterior draws from the stacking model (9.11) and pick a sparse (or cell-wise sparse) solution.

In contrast to fitting individual models in parallels before model averaging, an alternative approach is to fit all models jointly in a bigger mixture model. Kamary et al. (2019) proposed a Bayesian hypothesis testing by fitting an encompassing model $p(y|w,\theta) = \sum_{k=1}^{K} w_k p(y|\theta_k, M_k)$. The mixture model requires to simultaneously fit model parameters and model weights $p(w_{1,\dots,K}, \theta_{1,\dots,K}|y)$, of which the computation burden is a concern when K is big. Yao et al. (2018a) illustrated that (complete-pooling) stacking is often more stable than full-mixture, especially when sample size is small and some models are similar. Nevertheless, our formulation of hierarchical stacking agrees with Kamary et al. (2019) in terms of sampling from the posterior marginal distribution of $p(w|y)$ in a full-Bayesian model.

Hierarchical stacking bears a resemblance with "mixture of experts" (Jacobs et al., 1991; Jordan and Jacobs, 1994; Svensën and Bishop, 2003) in terms of using a local mixture to form ensembles: $p(y|x, w(x), \theta) = \sum_{k=1}^{K} w_k(x) p(y|x, \theta_k, M_k)$ However, hierarchical stacking differs from the mixture of experts, or mixture modeling at large, in the following ways:

1. Mathematically, stacking fits individual separately, hence the posterior predictive distribution of outcome y at location x can be expressed by

$$p(y|x, \mathcal{D}) = \int \left(\sum_{k=1}^{K} w_k(x) \int p(y|x, \theta_k, \mathcal{D}, M_k) p(\theta_k|\mathcal{D}) d\theta_k \right) p(\mathbf{w}\,|\mathcal{D}) d\,\mathbf{w}\,.$$

   But in mixture modeling, individual models and the weights are fitted jointly. The final posterior predictive distribution is

$$p(y|x, \mathcal{D}) = \int \cdots \int \left(\sum_{k=1}^{K} w_k(x) p(y|x, \mathcal{D}, M_k) \right) p(\theta_1, \dots, \theta_K, \mathbf{w}\,|\mathcal{D}) d\theta_1 \dots d\theta_K d\,\mathbf{w}\,.$$

2. Conceptually, the mixture model/mixture of experts is a way to build a model, while stacking or hierarchical stacking is a tool for model averaging. From a statistical workflow perspective, we would like to view a complete workflow containing (a) model building, (b) inference, and (c) model check, model evaluation, postprocessing, model comparison and model averaging. Of course, part of the message this paper conveys is that model evaluation is itself a model, such that there is no technical isolation between them. However, this distinction on when the method is called in the workflow implies what set of individual models this averaging method would apply to. As we wrote in the related literature section: "rather than to compete with a mixture-of-experts on combining weak learners, hierarchical stacking is more recommended to combine a mixture-of-experts with other sophisticated models."

3. A mixture of experts infers experts and gating models at the same time. It is possible to decouple these two tasks computationally. For example, in each M-step in the EM algorithm, the update on these two components are separated. But even so, the computation cost is typically super linear on the number of parameters. Because a mixture of experts needs to fit all models and gates in a single super model, it is often not feasible to obtain the exact posterior distribution. Rather its inference relies on a MAP or a variational approximation.

   In contrast, hierarchical stacking fits each individual model in advance, and the weights are trained separately. This division largely reduces computation cost. Of course, this division also comes with less flexibility: If the true data generating process is truly a mixture model, then fitting individual component $p(\theta_k|\mathcal{D})$ separately is wrong and stacking cannot remedy it. As we have argued, the right attitude is that we use stacking to combine plausible models that would otherwise be used as the final belief model.

   In the modern big data regime, we may often build models as big as possible. When individual models have already sat next to the computation-feasibility frontier, it is not viable to include them in an even bigger mixture model and fit all parameters jointly.

4. Another potential drawback of the joint training procedure in mixture modeling is the weak-

identification problem, which is especially a concern when individual models are "similar" or the sample size is not big enough. To verify this argument, Figure 6 in Yao et al. (2018a) compares complete-pooling stacking and mixture modeling to combine subset regressions. Despite a much longer inference time (about 30 more times when using MCMC) than stacking, it has worse predictive performance than stacking. When the sample size is small, the mixture model is too complex to fit, and is worse than any other model averaging and selection methods we have considered. That said, this example does not mean mixture modeling is useless, and the identification issues could be improved by a strong prior, but it does manifest the flaw of treating the mixture modeling as a panacea for model averaging.

5. Hierarchical stacking or stacking can be used to average over computation instabilities: Even conditioning on a single model, the posterior may contain multiple modes; the optimization may research different results due to random initialization, stochastic gradients, data partition, etc. Our GP example in the next section illustrates this idea in the context of posterior multimodality. By contrast, a mixture of experts itself often has a multimodal posterior (Svensën and Bishop, 2003).

6. There is a slight difference in how individual models are operationalized: the majority of literature on mixture-of-experts draws a distinction between classifications and regressions: for discrete variables, it is the probabilities from individual models that are additive; for continuous outcomes, it is the outcome itself that is additive, which is equivalent to a convolution of predictive densities. In our paper, there is no difference between classifications and regressions: we always additively combine the predictive densities.

7. The original mixture of experts method introduced by Jacobs et al. (1991) did not have explicit regularization on gate parameters $\mathbf{w}$ and was solved by maximum likelihood, which thereby could be called "no-pooling" (but we have deleted this oversimplification in related literature due to the reason AE has suggested).

   In the later development of "hierarchical mixture of experts"(Jordan and Jacobs, 1994), the

word "hierarchy" is related to the hierarchy of models, which are often represented by a binary tree. Rather, in this paper, the word "hierarchy" is more about the partition of data. That said, as a multinomial logistic regression is equivalent to a sequence of logistic regressions, the binary tree representation in hierarchical mixture of experts is not intrinsic.

Using Bayesian inference (variational Bayes) to train a mixture of experts was first proposed by Waterhouse et al. (1996), and also developed by Svensën and Bishop (2003). Besides the general concern on how close the variational approximation is to the exact posterior (e.g., Yao et al., 2018b, especially a concern here because of posterior multimodality), this variational approach also needs a conjugate prior, for which the default choice in Svensën and Bishop (2003) is (using our notation):

$$\sigma_k^2 \sim \text{InvGamma}(10^{-2}, 10^{-4}),$$

which has some undesired properties according to Gelman (2006).

8. Hierarchical stacking is also featured by its use of built-in leave-one-out likelihood. If a mixture of experts is used to combine a simple model and a significantly overfitting model, the MLE is prone to assigning all weights to the overfitting one.

9. Our full-Bayesian formulation makes it straightforward to apply hierarchical stacking to temporal, spatial, panel, or hierarchical data. In the polling example, we replace the leave-one-out score with leave-seven-day ahead score to reflect our prediction task on the election day. Because of the default conditional IID view, such extension to complex data structure has long been an open problem for the mixture of experts (Yuksel et al., 2012).

Lastly, it is natural that both the mixture of experts and stacking have limitations and usefulness. Some of our full-Bayesian formulation and the informative prior choice could be useful to motivate further development in the mixture of experts, too.

9.6 Examples

We present three examples. The well-switching example demonstrates an automated hierarchical stacking implementation with both continuous and categorical inputs. The Gaussian process example highlights the benefit of hierarchical stacking when individual models are already highly expressive. The election forecast illustrates a real-data example with a complex data structure. We evaluate the proposed method on several metrics, including the mean log predictive density on holdout data, conditional log predictive densities, and the calibration error.

Well-switching in Bangladesh. We work with a dataset used by Vehtari et al. (2017) who demonstrated cross validation. A survey with a size of $n = 3020$ was conducted on residents from a small area in Bangladesh that was affected by arsenic in drinking water. Households with elevated arsenic levels in their wells were asked whether or not they were interested in switching to a neighbor's well, denoted by y. To predict this well-switching behavior, we collect a series of household information x, including the detected arsenic concentration value in the well, the distance to the closest known safe well, the household's education level, and whether any household members is in community organizations. The first two inputs are continuous and the remaining two are categorical variables.

We fit a series of plausible logistic regressions, starting by an additive model including all covariates x in model 1. In model 2, we replace one input well arsenic level, by its logarithm value. In model 3 and 4, we add cubic spline basis functions with ten knots of well arsenic level and well-distance respectively in input variables. In model 5 we replace the categorical education stage by using years of schooling as a continuous variable.

Using the additive model specification (9.13) and default prior (9.14), we model the unconstrained weight $\alpha_k(x)$ by a linear regression of all categorical inputs and all rectified continuous inputs (9.16). In this example the categorical input has eight distinct levels based on the product of education (four levels) and community participation (binary).

For comparison, we consider three alternative approaches: (a) complete-pooling stacking (b)

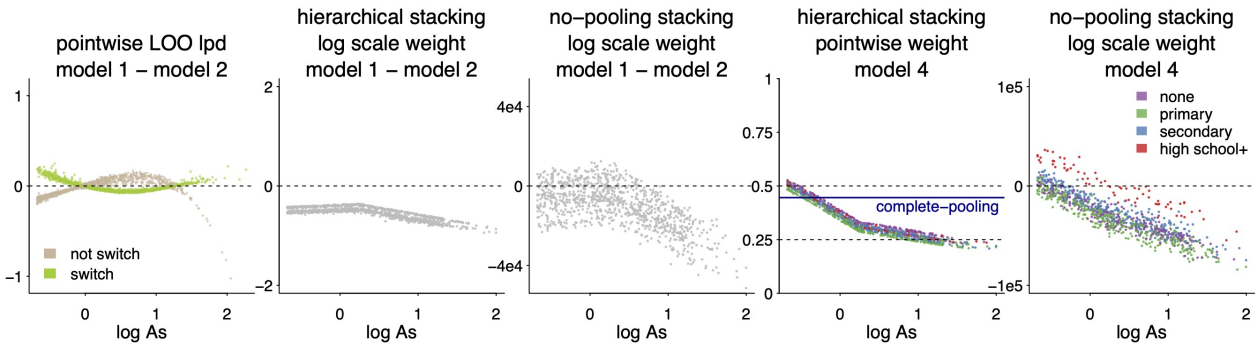


Figure 9.1: *(1) Pointwise difference of leave-one-out log scores between models 1 and 2, plotted against log arsenic. Model 1 poorly fits points with high arsenic. (2) Posterior mean of pointwise unconstrained weight difference between models 1 and 2, $\alpha_2(x) - \alpha_1(x)$ in hierarchical stacking. (3) Pointwise log weight difference between models 1 and 2 in no-pooling stacking. (4) Posterior mean of $w_4(x)$, the weight assigned to model 4, in hierarchical stacking, displayed against log arsenic and education levels. There are few sample with high school education and above, whose effect on model weights is pooled toward the shared mean. The blue line is the complete-pooling stacking. (5) The unconstrained weight of model 4, $\alpha_4(x)$, in no-pooling stacking. The "high school" effect stands out and the resulting model weights w are nearly all zeroes and ones.*

no-pooling stacking, or the maximum likelihood estimate of (9.13), and (c) model selection that picks model with the highest leave-one-out log predictive densities. We split the data into training set ($n_{\text{train}} = 2000$) and an independent holdout test set.

The leftmost panel in Figure 9.1 displays the pointwise difference of leave-one-out log scores for model 1 and 2 against log arsenic values in training data. Intuitively, model 1 fits poorly for data with high arsenic. In line with this evidence, hierarchical stacking assigns model 1 an overall low weight, and especially low for the right end of the arsenic levels. The second panel shows the pointwise posterior mean of unconstrained weight difference between model 1 and 2, $\alpha_2(x) - \alpha_1(x)$, against the arsenic values in training data. The no-pooling stacking reveals a similar direction that model 1's weight should be lower with a higher arsenic value, but for lack of hierarchical prior regularization, the fitted $\alpha_2(x) - \alpha_1(x)$ is orders of magnitude larger (the third panel). As a result, the realized pointwise weights w are nearly either zero or one.

The rightmost two columns in Figure 9.1 display the fitted pointwise weights of model 4 against log arsenic values and education level in test data. Because only a small proportion (7%) of respondents had high school education and above, the no-pooling stacking weight for this

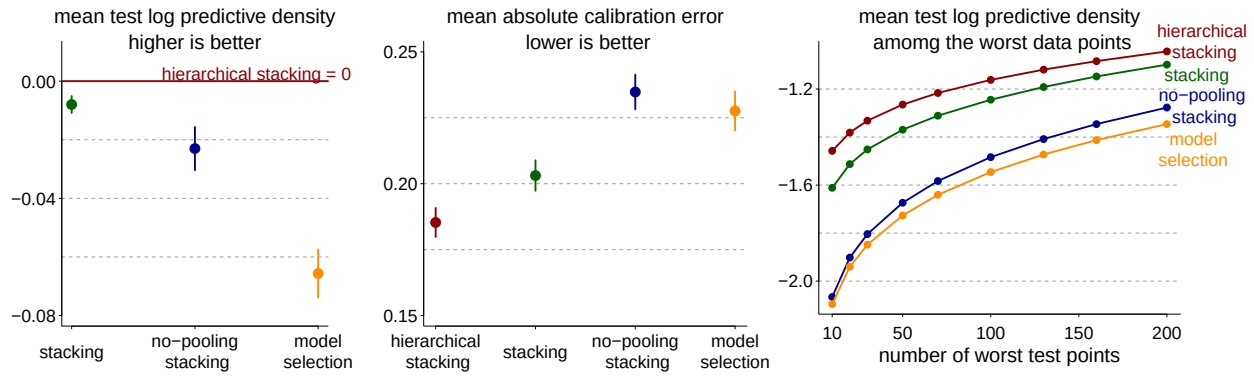


Figure 9.2: *We evaluate hierarchical, complete-pooling and no-pooling stacking, and model selection on three metrics: (a) average log predictive densities on test data, where we set the hierarchical stacking as benchmark 0, (b) calibration error: discrepancy between the predicted positive probability and realized proportion of positives in test data, averaged over 20 equally spaced bins, and (c) average log predictive destines among the* $10 \leq n_0 \leq 200$ *worst test data points. We repeat 50 random training-test splits with training size 2000 and test size 1020.*

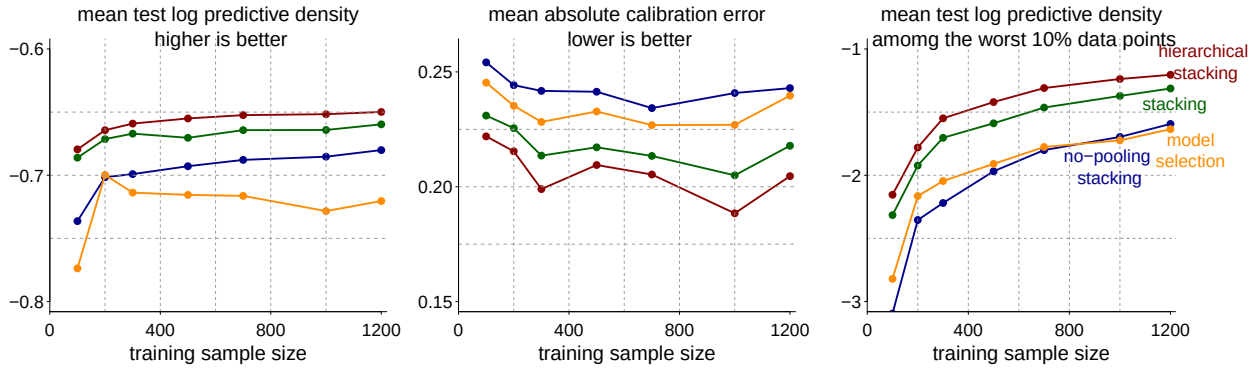


Figure 9.3: *Same comparisons as Figure 9.2, with training sample size varying from 100 to 1200.*

category is largely determined by small sample variation. Hierarchical stacking partially pools this "high school" effect toward the shared posterior mean of all educational levels, and the realized hierarchical stacking model weights do not clearly depend on education levels.

We evaluate model fit on three metrics. (a) The log predictive densities averaged over test data. In the first panel of Figure 9.2, we set hierarchical stacking as baseline and all other methods attain a lower predictive densities. (b) The L_1 calibration error. We set 20 equally spaced bin between 0 and 1. For each bin and each learning algorithm, we collect test data points whose model-predicted positive probability falling in that bin, and compute the absolute discrepancy between realized proportion of positives in test data and the model-predicted probabilities. The middle panel in

Figure 9.2 displays the resulted calibration error averaged over 20 bins. The proposed hierarchical stacking has the lowest error. No-pooling stacking has the highest calibration error despite of its higher overall log predictive densities than model selection, suggesting a prediction overconfidence. (c) We compute the average log predictive densities of four methods among the n_0 most shocking test data points (the ones with lowest predictive densities conditioning on a given method) for n_0 varying from 10 to 200 and the total test data has size 1020. As exhibited in the last panel in Figure 9.2, the proposed hierarchical stacking consistently outperforms all other approaches for all n_0: a robust performance for the worst case scenario. To reduce randomness, all evaluation metrics above are averaged over 50 random training-test splits.

Figure 9.3 presents the same comparisons of four methods while the training sample size n_{train} varies from 100 to 1200 (averaged over 50 random training-test splits). In agreement with the heuristic in Figure 10.1, the most complex method, no-pooling stacking, performs especially poorly with small sample size. By contrast the easiest method, model selection, reaches its peak elpd quickly with moderate sample size but cannot keep improving as training data size grows. The proposed hierarchical stacking performs the best in this setting under all metrics.

Gaussian process regression weighted by another Gaussian process. The local model averaging (9.12) tangles a x-dependent weight $w(x)$ and x-dependent individual prediction $p(y|x, M_k)$. If the individual model $y|x, M_k$ is big enough to have already exhausted "all" variability in input x, is there still a room for improvement by modeling local model weights $w(x)$? The next example suggests a positive answer.

Consider a regression problem with scalar observations $y_i = f(x_i) + \epsilon_i, i = 1, ..., n$, at input locations $\{x_i\}_{i=1}^n$, and ϵ_i are independent noises. The training model is a Gaussian process regression on the latent functions f with zero mean and squared exponential covariance:

$$y_i = f(x_i) + \epsilon_i,\ \epsilon_i \sim \text{normal}(0, \sigma),\ f(x) \sim \mathcal{GP}\left(0, a^2 \exp\left(-\frac{(x - x')^2}{\rho^2}\right)\right). \tag{9.17}$$

We adopt training data from Neal (1998). They were generated such that the posterior distribution

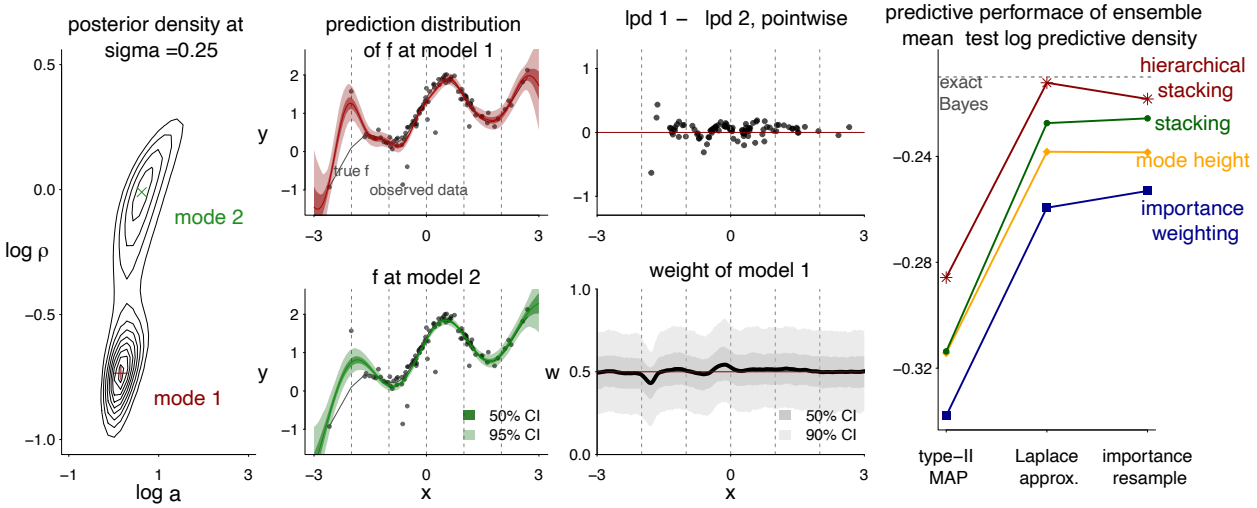


Figure 9.4: *From left to right, Column 1: posterior density at $\sigma = 0.25$. At least two modes exist. Column 2: predictive distribution of y from two modes. Column 3: the pointwise companion of log predictive density of the Laplace approximations at two modes, and the hierarchical stacking weight of mode 1. Column 4: the test data mean predictive densities of the weighted model, where individual components in the final model consists of either the MAP, Laplace approximation, or importance sampling around the two modes, and the weighting methods include hierarchical stacking, complete-pooling stacking, mode heights and importance weighing.*

of hyperparameters $\theta = (a, \rho, \sigma)$ contains at least two isolated modes (see the first panel in Figure 9.4). We consider three mode-based approximate inference of $\theta|y$: (a) Type-II MAP, where we pick the local modes of hyperparameters that maximizes the marginal density $\hat{\theta} = \arg\max p(\theta|y)$, and further draw local variables $f|\hat{\theta}, y$, (b) Laplace approximation of $\theta|y$ around the mode, and (c) importance resampling where we draw uniform samples near the mode and keep sample with probability proportional to $p(\theta|y)$. In the existence of two local modes $\hat{\theta}_1, \hat{\theta}_2$, we either obtain two MAPs, or two nearly-nonoverlapped draws, further leading to two predictive distributions. Yao et al. (2020b) suggests to use complete-pooling stacking to combine two predictions, which shows advantages over other ad-hoc weighting strategies such as mode heights or importance weighting.

Visually, mode 1 has smaller length scale, more wiggling and attracted by training data. Because of a better overall fit, it receives higher complete-stacking weights. However, the wiggling tail makes its extrapolation less robust. We now run hierarchical stacking with x-dependent weight $w_k(x)$ for

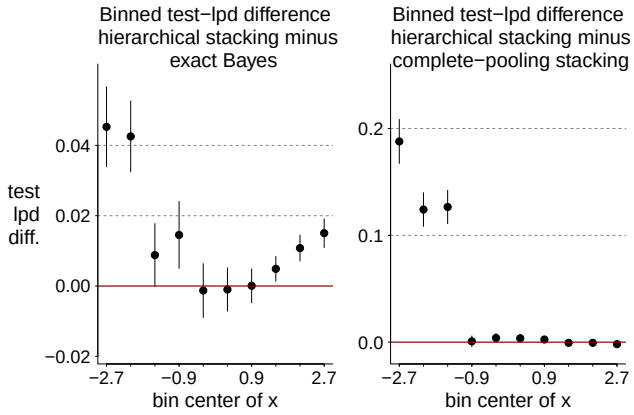


Figure 9.5: *Comparison between hierarchical stacking and stacking of two locally Laplace approximation (left) or a long-chain exact Bayes from the true model (right). We compare the binned test log predictive densities over 10 equally spaced bins on* $(-3, 3)$. *A positive value means hierarchical stacking has a better fit than the compared method.*

mode $k = 1, 2$ by placing a Gaussian process prior on unconstrained weight $\text{logit}(w_1(x))$,

$$w_1(x) = \text{invlogit}\,(\alpha(x)), \;\; \alpha(x) \sim \mathcal{GP}(0, K(x)).$$

Despite the same GP prior, it is not related to the training regression model (9.17). To evaluate how good the weighted ensemble is, we generate independent holdout test data $(\tilde{x}_i, \tilde{y}_i)$. Both training and test inputs, x and $\tilde{x}$, are distributed from $\text{normal}(0, 1)$. As presented in the rightmost panel in Figure 9.4, for all three approximate inferences, hierarchical stacking always has a higher mean test log predictive density than complete pooling stacking and other weighting schemes.

In this dataset, exact MCMC is able to explore both posterior modes in model (9.17) after a long enough sampling. Gaussian process regression equipped with exact Bayesian inference can be regarded as the "always true" model here. Hierarchical stacking achieves a similar average test data fit by combining two Laplace approximations. Furthermore, hierarchical stacking has better predictive performance under covariate shift. To examine local model fit, we generate another independent holdout test data. This time the test inputs $\tilde{x}$ are from $\text{uniform}(-3, 3)$. We divide the test data into 10 equally spaced bins, and compute the mean test data log predictive density inside each bin. Compared with exact inference, hierarchical stacking has a comparable performance in

the bulk region of x, while it yields higher predictive densities in the tail, suggesting a more reliable extrapolation.

U.S. presidential election forecast. We explore the use of hierarchical stacking on a practical example of forecasting polls for the 2016 United States presidential election. Since the polling data are naturally divided into states, it provides a suitable platform for hierarchical stacking in which model weights vary on states.

To create a pool of candidate models, we first concisely describe the model of Heidemanns et al. (2020), an updated dynamic Bayesian forecasting model (Linzer, 2013) for the presidential election, and then follow up with different variations of it. Let i be the index of an individual poll, y_i the number of respondents that support the Democratic candidate and n_i the number of respondents who support either the Democratic or the Republican candidate in the poll. Let $s[i]$ and $t[i]$ denote the state and time of poll i respectively. The model is expressed by

$$
\begin{aligned}
y_i &\sim \text{Binomial}(\theta_i, n_i), \\
\theta_i &= \begin{cases} \text{logit}^{-1}(\mu^b_{s[i],t[i]} + \alpha_i + \zeta^{\text{state}}_i + \xi_{s[i]}), & i \text{ is a state poll,} \\ \text{logit}^{-1}(\sum_{s=1}^{S} u_s \mu^b_{s,t[i]} + \alpha_i + \zeta^{\text{national}}_i + \sum_{s=1}^{S} u_s \xi_s), & i \text{ is a national poll,} \end{cases}
\end{aligned}
\tag{9.18}
$$

where superscripts denote parameter names, and subscripts their indexes. The term μ^b is the underlying support for the Democratic candidate and α_i, ζ, and ξ represent different bias terms. α_i is further decomposed into

$$
\alpha_i = \mu^c_{p[i]} + \mu^r_{r[i]} + \mu^m_{m[i]} + z\epsilon_{t[i]}, \tag{9.19}
$$

where μ^c is the house effect, μ^r polling population effect, μ^m polling mode effect, and ϵ an adjustment term for non-response bias. Furthermore, an AR(1) process prior is given to the μ^b: $\mu^b_t | \mu^b_{t-1} \sim \text{MVN}(\mu^b_{t-1}, \Sigma^b)$, where Σ^b is the estimated state-covariance matrix and μ^b_T is the estimate from the fundamentals.

Although we believe this model reasonably fits data, there is always some researcher degrees

of freedom in model building and a room for improvement. Our pool of candidates consists of eight models. M_1: The fundamentals-based model by Abramowitz (2008). M_2: The model by Heidemanns et al. (2020). M_3: M_2 without the fundamentals prior, $\mu_T^b = 0$. M_4: M_2 with an AR(2) structure, $\mu_t^b | \mu_{t-1}^b, \mu_{t-2}^b \sim \text{MVN}(0.5\mu_{t-1}^b \mu_{t-2}^b, \Sigma^b)$. M_5: simplify M_2 without polling population effect, polling mode effect, and the adjustment trend for non-response bias, $\alpha_i = \mu_{p[i]}^c$. M_6: M_2 where we added an extra regression term $\beta_{\text{stock}} \text{stock}_{t[i]}$ into model (9.18) using the S&P 500 index at the time of poll i. M_7: M_2 without the entire shared bias term, $\alpha_i = 0$. M_8: M_2 without hierarchical structure on states.

We equip hierarchical stacking with either the basic independent prior (9.9) or the state-correlated prior (9.15). The state correlation Ω is estimated using a pool of state-level macro variables, and has already been used in some of the individual models to partially pool state level polling. We plug this pre-estimated prior correlation in the correlated stacking prior (9.15), and refer to it "hierarchical stacking with correlation" in later comparisons.

Since the data are longitudinal, we evaluate different pooling approaches using a one-week-ahead forecast with an expanding window for each conducted poll. We extract the fitted one-week ahead predictions from each individual model, and train hierarchical stacking, complete-pooling and no-pooling stacking to find the optimal model weights, and evaluate the combined models by computing their mean log predictive densities on the unseen test data next week. To account for non-stationarity discussed in Section 9.4, we only use the last four week prior to prediction day for training model averaging. In the end we obtain a trajectory of this back-testing performance of hierarchical stacking, complete-pooling and no-pooling stacking and single model selection.

The left-hand side of Figure 9.6[3] shows the seven-day running-average of the one-week-ahead back-test log predictive density from models combined with various approaches. The right-hand side of Figure 9.6 shows the overall cumulative one-week-ahead back-test log predictive density. We set the uncorrelated hierarchical stacking to be a constant zero for reference. Hierarchical stacking performs the best, followed by stacking, no-pooling stacking, and model selection respectively. The

[3]Figure 9.6, 9.7, 9.8 were created by Gregor Pirš.

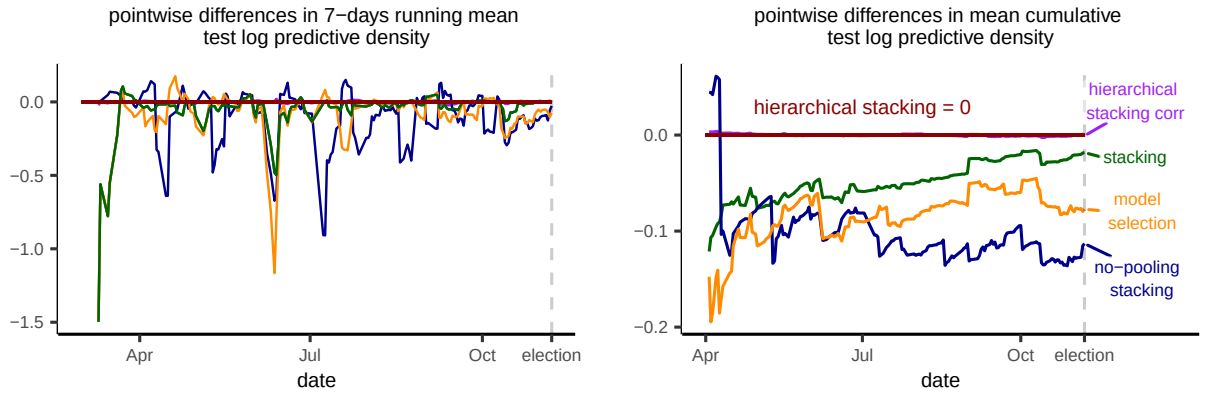


Figure 9.6: *Left: pointwise differences in 7-day running mean log predictive densities on one-week-ahead test data, where we set the hierarchical stacking as benchmark 0. Right: pointwise differences in cumulative average predictive log density by date. The advantage of hierarchical stacking is most noticeable toward the beginning, where there are fewer polls available.*

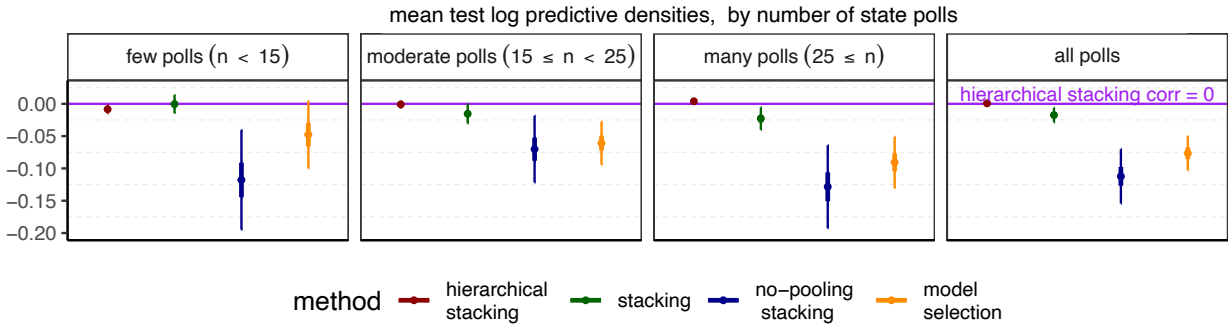


Figure 9.7: *Mean test log predictive densities with 50% and 95% confidence intervals, among subsets of states with few, moderate and many number of state polls, and among all states. Correlated hierarchical stacking is set as reference 0. It is better than independent hierarchical stacking when data are scarce. Complete-pooling stacking is close to hierarchical stacking in small states, but worse in bigger states.*

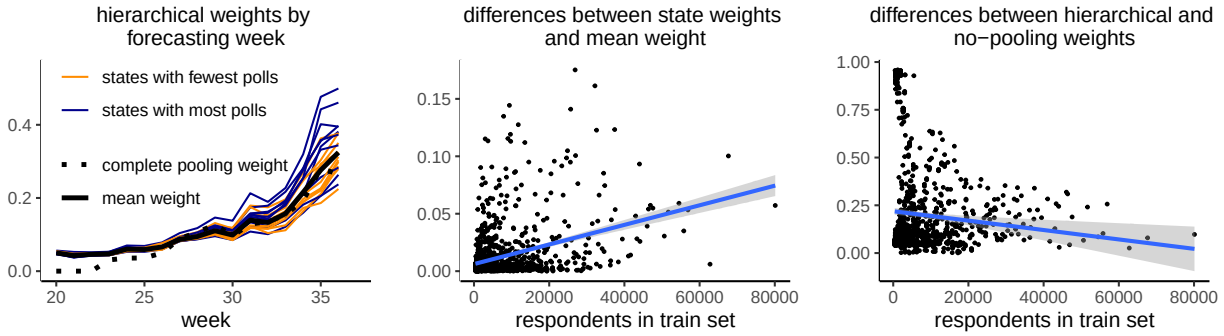


Figure 9.8: *Hierarchical stacking weights for M_1 in the polling example. Left: weights for M_1 of the 10 states with fewest polls and with most polls over time. Dotted line shows the complete-pooling stacking weight and the solid black line is the nationwide mean weight. States with fewer polls are shrunken more toward the mean. Middle: absolute differences between state-wise hierarchical stacking weights and the nationwide mean, against number of respondents. The blue line is the linear trend reference. States with smaller sample sizes are more pooled to the mean. Right: absolute differences between hierarchical stacking and no-pooling stacking weights, generally decreasing with bigger sample sizes.*

advantage of hierarchical stacking is highest at the beginning and slowly decreases the closer we get to election day. As we move closer to the election, more polls become available, so the candidate models become better and also more similar since some models only differ in priors. As a result, all combination methods eventually become more similar. No-pooling stacking has high variance and hence performs the worst out of all combination methods. Hierarchical stacking with correlated prior performs similarly to the independent approach, with a minor advantage at the beginning of the year, where the prior correlation stabilizes the state weights where the data are scarce, and later we see this advantage more discernible in individual states.

To examine small area estimations, we divided states into three categories based on how many state polls were conducted. Figure 9.7 shows the overall mean pointwise differences in test log predictive densities divided by these categories, along with a fourth panel over all states. No-pooling stacking performs the worst in all panels. An explanation for that could be that we are using a four-week moving window to tackle non-stationarity, which might not contain enough data for the no-pooling method. The variance of the no-pooling is amended by the hierarchical approach, which performs on par with stacking with scarcer data and outperforms it otherwise.

We can also observe the advantage of using prior correlations in hierarchical stacking in the first panel. With a large number of state polls available, for example close to election day in Florida and North Carolina, no-pooling stacking performs well. In states with fewer polls, no-pooling stacking is unstable. Hierarchical stacking alleviates this instability, while retaining enough flexibility for a good performance with large data come in.

Figure 9.8 illustrates how cell size affects the pooling effect. The first panel show the hierarchical stacking state-wise weights for the first candidate model w_{1j} as a function of date. For either early-date forecasts or states with few polls, hierarchical stacking weights are more pooled toward the shared nationwide mean. The middle and right panels compare the difference between state-wise hierarchical stacking weights and the nationwide mean, or with no-pooling weights, against the total number of respondents for each state and prediction date. The cells with more observed data are less pooled and closer to their no-pooling optimums, and vice versa.

9.7 Discussion

Robustness in small areas. The input-varying model averaging is designed to improve both the overall averaged prediction $\mathbb{E}_{\tilde{y},\tilde{x}}(\log p(\tilde{y}|\tilde{x}))$ and conditional prediction $\mathbb{E}_{\tilde{y}|\tilde{x}=x_0}(\log p(\tilde{y}|\tilde{x}))$, whereas these two tasks are subject to a trade-off in complete-pooling stacking.

In addition, the partial pooling prior (9.9) borrows information from other cells, which stabilizes model weights in small cells where there are not enough data for no-pooling stacking. For a crude mean-field approximation, the likelihood in the discrete model (9.11)$\approx \prod_{j,k} \text{normal}(\alpha_{jk}^{\text{mode}}, \lambda_{jk})$, where $\alpha^{\text{mode}} = \arg\max_\alpha \sum_{i=1}^n \log\left(\sum_{k=1}^K w_k(x_i) p_{k,-i}\right)$ is the unconstrained no-pooling stacking weight, and $-\lambda_{jk}^{-2} = \frac{\partial^2}{\partial \alpha_{jk}^2}|_{\text{mode}} \sum_{i=1}^n \log\left(\sum_{k=1}^K w_k(x_i) p_{k,-i}\right)$ is the diagonal element of the Hessian. Because α_{jk} appears in n_j terms of the summation, $\lambda_{jk} = O(n_j^{-1/2})$ for a given k. Combined with the prior $\alpha_{jk} \sim \text{normal}(\mu_k, \sigma_k)$, the conditional posterior mean of the k-th model weight in the j-th cell is the usual precision-weighed average of the no-pooling optimum and the shared mean: $\hat{\alpha}_{jk}|\lambda_{jk}, \sigma_k, \mu_k \approx (\lambda_{jk}^{-2}\alpha_{jk}^{\text{mode}} + \sigma_k^{-2}\mu_k)(\lambda_{jk}^{-2} + \sigma_k^{-2})^{-1}$. Hence for a given model k, $|\alpha_{jk}^{\text{mode}} - \hat{\alpha}_{jk}| = O(n_j^{-1})$: larger pooling usually occurs in smaller cells. This pooling factor is in line with Figure 9.8

and the general hierarchical model theory (Gelman and Pardoe, 2006). Our full-Bayesian solution also integrates out μ_k and σ_k, which further partially pools across models.

The possibility of partial pooling across cells encourages open-ended data gathering. In the election polling example, even if a pollster is only interested in the forecast of one state, they could gather polling data from everywhere else, fit multiple models, evaluate models on each state, and use hierarchical stacking to construct model averaging, which is especially applicable when the state of interest does not have enough polls to conduct a meaningful model evaluation individually. In this context swing states naturally have more state polls, so that the small-area estimation may not be crucial, but in general we conjecture that the hierarchical techniques can be useful for model evaluation and averaging in a more general domain adaptation setting. Without going into extra details, hierarchical models are as useful to make inferences from a subset of data (small-area estimation) as to generalize data to a new area (extrapolation). When the latter task is the focus, hierarchical stacking only needs to redefine the leave-one-data-out predictive density (9.6) by leave-one-cell-out $p_{k,-i} \coloneqq \int_{\Theta_k} p(y_i|\theta_k, x_i, z_i, M_k) p\left(\theta_k | M_k, \{(x_{i'}, z_{i'}, y_{i'}) : x_{i'} \neq x_i\}\right) d\theta_k$.

Using hierarchical stacking to understand local model fit. We use hierarchical stacking not only as a tool for optimizing predictions but also as a way to understand problems with fitted models. The fact that hierarchical stacking is being used is already an implicit recognition that we have different models that perform better or worse in different subsets of data, and it can valuable to explore the conditions under which different models are fitting poorly, reveal potential problems in the data or data processing, and point to directions for individual-model improvement.

Vehtari et al. (2017) and Gelman et al. (2020) suggested to examine the pointwise cross-validated log score $\log p_{k,-i}$ as a function of x_i, and see if there is a pattern or explanation for why observations are harder to fit than others. For example, the first panel of Figure 9.1 seems to indicate that Model 1 is incapable to fit the rightmost 10–15 non-switchers. However, $\log p_{k,-i}$ contains a non-vanishing variance since y_i is a single realization from $p_t(y|x_i)$. Despite its merit in exploratory data analysis, it is hard to tell from the raw cross validation scores that whether Model

1 is incapable of fitting high arsenic, or is merely unlucky for these few points. The hierarchical stacking weight $w(x)$ provides a smoothed summary of how each model fits locally in x and comes with built-in Bayesian uncertainty estimation. For example, in Figure 9.4, $\log p_{1,-i} - \log p_{2,-i}$ has a slightly inflated right tail, but this small bump is smoothed by stacking, and the local weight therein is close to $(0.5, 0.5)$.

Retrieving a formal likelihood from an optimization objective. The implication of hierarchical stacking (9.11) being a formal Bayesian model is that we can evaluate its posterior distribution as with in a regular Bayesian model. For example, we can run (approximate) leave-one-out cross validation of the the stacking posterior $p(w|\mathcal{D}_{-i}) \propto p(w|\mathcal{D})/p(y_i|x_i, w) = p(w|\mathcal{D})/\left(\sum_{k=1}^K w_k(x_i)p_{k,-i}\right)$. In practice, we only need to fit the stacking model (9.11) once, collect a size-S MCMC sample of stacking parameters from the full posterior $p(w|\mathcal{D})$, denoted by $\{(w_{k1}(x_i), \ldots, w_{kS}(x_i))\}_{i,k}$, compute the PSIS-stabilized importance ratio of each draw $r_{is} \approx \left(\sum_{k=1}^K w_{ks}(x_i)p_{k,-i}\right)^{-1}$, and then compute the mean leave-one-out cross validated log predictive density to evaluate the overall out-of-sample fit of the final stacked model:

$$\begin{aligned}\text{elpd}^{\text{loo}}_{\text{stacking}} &= \sum_{i=1}^n \log \int_{\mathcal{S}_K} p(y_i|x_i, w(x_i))p(w(x_i)|\mathcal{D}_{-i})d(w(x_i)) \\ &\approx \sum_{i=1}^n \log \frac{\sum_{s=1}^S \left(r_{is}\sum_{k=1}^K w_{ks}(x_i)p_{k,-i}\right)}{\sum_{s=1}^S r_{is}}. \end{aligned} \tag{9.20}$$

As discussed in Section 9.5, the same task of out-of-sample prediction evaluation in an optimization-based stacking requires double cross validation (refit the model $n(n-1)$ times if using leave-one-out), but now becomes almost computationally free by post-processing posterior draws of stacking.

The Bayesian justification above applies to log score stacking. In general we cannot convert an arbitrary objective function into a log density—its exponential is not necessarily integrable; Even if it is, the resulted density does not necessarily match a relevant model. Take linear regression for example, the ordinary least square estimate $\arg\min_\beta \sum_{i=1}^n (y_i - x_i^T\beta)^2$ equivalents the maximum likelihood estimate of β from a probabilistic model $y_i|x_i, \beta, \sigma \sim \text{normal}(x_i^T\beta, \sigma)$

with flat priors. But the directly adapted "log posterior density" from the the negative L^2 loss, $\log p(\beta|y) = -\sum_{i=1}^{n}(y_i - x_i^T\beta)^2 + C$, differs from the Bayesian inference of the latter probabilistic model unless $\sigma \equiv 1$. The hierarchical stacking framework may still work for other scoring rules in practice, while we leave their Bayesian justification for future research.

Statistical workflow for black-box algorithms. Unlike our previous work (Yao et al., 2018a) that merely applied stacking to Bayesian models, the present chapter converts optimization-based stacking itself into a formal Bayesian model, analogous to reformulating a least-squares estimate into a normal-error regression. Breiman (2001) distinguished between two cultures in statistics: the *generative modeling* culture assumes that data come from a given stochastic model, whereas the *algorithmic modeling* treats the data mechanism unknown and advocates black-box learning for the goal of predictive accuracy. As a method that Breiman himself introduced, stacking is arguably closer to the algorithmic end of the spectrum, while our hierarchical Bayesian formulation pulls it toward to the generative modeling end.

Such a formulation is appealing for two reasons. First, the generative modeling language facilitates flexible data inclusion during model averaging. For example, the election forecast model contains various outcomes on state polls and national polls from several pollsters, and pollster-level, state-level, and national-level fundamental predictors as well as prior state-level correlations. It is not clear how methods like bagging or boosting can include all of them. Data do not have to conveniently arrive in independent (x_i, y_i) pairs and compliantly await an algorithm to train upon. Second, instead of a static one-fit-all algorithm, hierarchical stacking is now part of a statistical workflow (Gelman et al., 2020) that is subject to model criticisms and model improvements—we can incorporate other Bayesian shrinkage priors as add-on components without reinventing them, and we can use approximate leave-one-out cross validation (9.20) to check the out-of-sample performance of the final stacking model and compare multiple priors. Looking ahead, the success of this work suggests that we can use generative Bayesian modeling to improve other black-box prediction algorithms.

9.8 Software implementation

We summarize our formulation of hierarchical stacking by pseudo code 8.

Algorithm 8: Hierarchical stacking

Data: y: outcomes; x: input on which the stacking weights vary, z: other inputs; $p_{k,-i}$: approximate leave-one-out predictive densities of the k-th model and i-th data.
Result: input-dependent stacking weight $\hat{w}(x) : \mathcal{X} \to \mathcal{S}_K$; combined model.

1 Sample from the joint densities $p(\alpha, \mu, \sigma|\mathcal{D})$ in hierarchical stacking model (9.11);
2 Compute posterior mean of $w_k(\tilde{x})$ at any $\tilde{x}$, and make predictions $p(\tilde{y}|\tilde{x}, \tilde{z})$ by (9.12).

To code the basic additive model, we prepare the input covariate $X = (X_{\text{discrete}}, X_{\text{continuous}})$, where X_{discrete} is discrete dummy variable, and $X_{\text{continuous}}$ are remaining features (already rectified as in (9.16)). The dimension of these two parts are $d_{\text{continuous}}$ and d_{discrete}.

Here we use the "grouped hierarchical priors" (Section 9.7) with only two groups, distinguishing between continuous and discrete variables.

$$w_{1:K}(x) = \text{softmax}(w^*_{1:K}(x)), \;\; w^*_k(x) = \sum_{m=1}^{M} \alpha_{mk} f_m(x) + \mu_k, \;\; k \le K-1, \;\; w^*_K(x) = 0,$$

$$\alpha_{mk} \mid \mu_k, \sigma_{k1} \sim \text{normal}(0, \sigma_{k1}), \; k = 1, \ldots, K-1, \; m = 1, \ldots, d_{\text{discrete}},$$

$$\alpha_{mk} \mid \mu_k, \sigma_{k2} \sim \text{normal}(0, \sigma_{k2}), \; k = 1, \ldots, K-1, \; m = d_{\text{discrete}} + 1, \ldots, d_{\text{discrete}} + d_{\text{continuous}},$$

$$\mu_k \sim \text{normal}(\mu_0, \tau_\mu), \quad \sigma_{k1} \sim \text{normal}^+(0, \tau_{\sigma 1}), \sigma_{k2} \sim \text{normal}^+(0, \tau_{\sigma 2}), \quad k = 1, \ldots, K-1.$$

This model is encoded by the following `Stan` (Stan Development Team, 2020) program:

```
data {
int<lower=1> N; // number of data points
int<lower=1> d; //number of input variables
int<lower=1> d_discrete; // number of discrete dummy inputs
int<lower=3> K;  // number of models
//when K=2, replace softmax by inverse-logit for higher efficiency
matrix[N,d] X;   // predictors
//including continuous and discrete in dummy variables, no constant
matrix[N,K] lpd_point;  //the input pointwise  predictive density
real<lower=0> tau_mu;
real<lower=0> tau_discrete;//overall regularization for discrete variables
real<lower=0> tau_con;//overall regularization for continuous variables
```

```
}
transformed data {
matrix[N,K] exp_lpd_point=exp(lpd_point);
}
parameters {
vector[K-1] mu;
real mu_0;
vector<lower=0>[K-1] sigma;
vector<lower=0>[K-1] sigma_con;
vector[d-d_discrete] beta_con[K-1];
vector[d_discrete] tau[K-1]; // using non-centered parameterization
}
transformed parameters {
vector[d] beta[K-1];
simplex[K] w[N];
matrix[N,K] f;
for (k in 1:(K-1))
beta[k] = append_row(mu_0*tau_mu+mu[k]*tau_mu+  sigma[k]*tau[k],
sigma_con[k]*beta_con[k]);
for (k in 1:(K-1))
f[,k] = X * beta[k];
f[,K] = rep_vector(0, N);
for (n in 1:N)
w[n] = softmax(to_vector(f[n, 1:K]));
}
model {
for (k in 1:(K-1)){
tau[k] ~ std_normal();
beta_con[k] ~ std_normal();
}
mu ~ std_normal();
mu_0 ~ std_normal();
sigma ~ normal(0, tau_discrete);
sigma_con ~ normal(0,tau_con);
for (i in 1:N)
target += log(exp_lpd_point[i,] * w[i]); //log likelihood
}
```

To run this stacking program on model fits, we can fit all individual models in `Stan`, and extract their leave-one-out likelihoods $\{p_{k,-i}\}$ by efficient approximation in R-package `loo` (Vehtari et al., 2019a):

```
library(loo)  # https://mc-stan.org/loo/
lpd_point <- matrix(NA, nrow(X), K)
for (k in 1:K) {
fit_stan <- stan(stan_moodel = model_k, data = ...) #x may differ in models
```

```
log_lik <- extract_log_lik(fit_stan, merge_chains = FALSE)
lpd_point[,k] <- loo(log_lik, r_eff = relative_eff(exp(log_lik)))$pointwise
}
```

Finally we run hierarchical stacking as a regular Bayesian model in Stan.

```
library(rstan)  # https://mc-stan.org/rstan/
fit_stacking <- stan("stacking.stan", # save the stacking code to a stan file.
data = list(X = X, N = nrow(X), d=ncol(X),
d_discrete=d_discrete, lpd_point=lpd_point, K=ncol(lpd_point),
tau_mu = 1, tau_sigma = 1, tau_discrete= 0.5, tau_con = 1))
# extract posterior simulations of normalized weights:
w_fit <- extract(fit_stacking, pars='w')$w
```

Chapter 10. Characterizing the diversity of models

"Truth is so precious that she should always be attended by a bodyguard of lies."

—Winston Churchill

10.1 All models are wrong, but some are somewhere useful

With an $\mathcal{M}$-closed view (Bernardo and Smith, 1994), one of the candidate models is the true data generating process, whereas in the more realistic $\mathcal{M}$-open scenario, none of the candidate models is completely correct, hence models are evaluated to the extent that they interpret the data.

The expectation of a strictly proper scoring rule, such as the expected log predictive density (elpd), is maximized at the correct data generating process. However, the extent to which a model is "true" is contingent on the input information we have collected. Consider an input-outcome pair (x, y) generated by

$$x \in [0,1],\ y \in \{0,1\}, \quad x \sim \text{uniform}(0,1),\ \Pr(y = 1|x) = x.$$

If the input x is not observed or is omitted in the analysis, then $M_1 : y \sim \text{Bernoulli}(0.5)$ is the only correct model and is optimal among all probabilistic predictions of y unconditioning on x. But this marginally *true* model is strictly worse than a *misspecified* conditional prediction, $M_2 : \Pr(y = 1|x) = \sqrt{x}$, since the expected log predictive densities are $\log(0.5) = -0.69$ and $-\frac{7}{12} = -0.58$ respectively after averaged over x and y. The former model is true purely because it ignores some predictors.

This wronger-model-does-better example does not contradict the log score being strictly proper, as we are changing the decision space from measures on y to conditional measures on $y|x$. But this example does underline two properties of model evaluation and averaging. First, we have little

interest in a binary model check. The hypothesis testing based model-being-true-or-false depends on what variables to condition on and is not necessarily related to model fit or prediction accuracy. In a non-quantum scheme, a really "everywhere true" model that has exhausted all potentially unobserved inputs contains no aleatory uncertainty. Second, the model fits typically vary across the input space. In the Bernoulli example, despite its larger overall error, M_1 is more desired near $x \approx .5$, and is optimal at $x = .5$.

For theoretical interest, we define the conditional (on $\tilde{x}$) expected (on $\tilde{y}|\tilde{x}$) log predictive density in the k-th model, $\text{celpd}_k(\tilde{x}) \coloneqq \int_{\mathcal{Y}} p_t(\tilde{y}|\tilde{x}) \log p(\tilde{y}|\tilde{x}, M_k) d\tilde{y}$. If $\{\text{celpd}_k\}_{k=1}^K$ are known, we can divide the input space $\mathcal{X}$ into K disjoint sets based on which model has the *locally* best fit (When there is a tie, the point is assigned the smallest index, and $\mathcal{I}$ stands for "input"):

$$\mathcal{I}_k \coloneqq \{\tilde{x} \in \mathcal{X} : \text{celpd}_k(\tilde{x}) > \text{celpd}_{k'}(\tilde{x}), \forall k' \neq k\},\ k = 1, \ldots, K. \tag{10.1}$$

In this Bernoulli example, $\mathcal{I}_1 = [0.25, 0.67]$.

10.2 What does model dissimilarity mean, why does stacking help, and why can hierarchical stacking help more

The consistency of leave-one-out cross validation ensures that complete-pooling stacking (9.3) is asymptotically no worse than model selection in predictions (Clarke, 2003; Le and Clarke, 2017), hence justified by Bayesian decision theory. The theorems we establish in Section 10.2 go a step further, providing lower bounds on the utility gain of stacking and hierarchical stacking. In short, model averaging is more pronounced when the model predictive performances are locally separable, but in the same situation we can improve the linear mixture model from learning locally which model is better, so that the stacking is a step toward model improvement rather than an end to itself. We illustrate with a theoretical example in Appendix 10.3 and provide proofs in Appendix 10.4.

In this subsection, we focus on the oracle expressiveness power of model selection and averaging, and their input-dependent version. $\mathbf{w}^{\text{stacking,cp}}$ refers to the complete-pooling stacking weight in the

population:

$$\mathbf{w}^{\text{stacking,cp}} := \arg\max_{\mathbf{w}\in\mathcal{S}_K} \text{elpd}(\mathbf{w}),$$
$$\text{elpd}(\mathbf{w}) = \int_{\mathcal{X}\times\mathcal{Y}} \log\left(\sum_{k=1}^{K} w_k p(\tilde{y}|M_k,\tilde{x})\right) p_t(\tilde{y},\tilde{x})d\tilde{y}d\tilde{x}. \tag{10.2}$$

Apart from the heuristic that model averaging is likely to be more useful when candidate models are more "dissimilar" or "distinct" (Breiman, 1996b; Clarke, 2003), we are not aware of rigorous theories that characterize this "diversity" regarding the effectiveness of stacking. It seems tempting to use some divergence measure between posterior predictions from each model as a metric of how close these models are, but this is irrelevant to the true data generating process.

We define a more relevant metric on how individual predictive distributions can be *pointwisely* separated. The description of a forecast being good is probabilistic on both $\tilde{x}$ and $\tilde{y}$: an overall bad forecast may be lucky at an one-time realization of outcome $\tilde{y}$ and covariate $\tilde{x}$. We consider the input-output product space $\mathcal{X}\times\mathcal{Y}$ and divide it into K disjoints subsets ($\mathcal{J}$ stands for "joint"):

$$\mathcal{J}_k := \{(\tilde{x},\tilde{y})\in\mathcal{X}\times\mathcal{Y} : p(\tilde{y}|M_k,\tilde{x}) > p(\tilde{y}|M_{k'},\tilde{x}), \forall k'\neq k\},\ k=1,\ldots,K.$$

In this framework, we call a family of predictive densities $\{p(\tilde{y}|M_k,\tilde{x})\}_{k=1}^K$ to be locally separable with a constant pair $L>0$ and $0\leq\epsilon<1$, if

$$\sum_{k=1}^{K}\int_{(\tilde{x},\tilde{y})\in\mathcal{J}_k} \mathbb{1}\Big(\log p(\tilde{y}|M_k,\tilde{x}) < \log p(\tilde{y}|M_{k'},\tilde{x}) + L,\ \forall k'\neq k\Big) p_t(\tilde{y},\tilde{x})d\tilde{y}d\tilde{x} \leq \epsilon. \tag{10.3}$$

Stacking is sometimes criticized for being a black box. The next two theorems link stacking weight to a probabilistic explanation. Unlike Bayesian model averaging (Hoeting et al., 1999) that computes the probability of a model being "true", stacking is more related to $\Pr(\mathcal{J}_k)$: the probability of a model being the locally "best" fit, with respect to the true joint measure $p_t(\tilde{y},\tilde{x})$.

Theorem 10.1 (probabilistic limit of stacking weights)**.** *When the separation condition* (10.3)

holds, the complete pooling stacking weight is approximately the probability of the model being the locally best fit:

$$w_k^{\text{stacking,cp}} \approx w_k^{\text{approx}} := \Pr(\mathcal{J}_k) = \int_{\mathcal{J}_k} p_t(\tilde{y}, \tilde{x}) d\tilde{y} d\tilde{x}, \tag{10.4}$$

in the sense that the objective function is nearly optimal:

$$|\operatorname{elpd}(\mathbf{w}^{\text{approx}}) - \operatorname{elpd}(\mathbf{w}^{\text{stacking,cp}})| \leq O(\epsilon + \exp(-L)). \tag{10.5}$$

Further, a model is only ignored by stacking if its winning probability is low.

Theorem 10.2 (model bound when stacking weight is sparse)**.** *When the separation condition* (10.3) *holds, and if the k-th model has zero weight in stacking, $w_k^{\text{stacking,cp}} = 0$, then the probability of its winning region is bounded by:*

$$\Pr(\mathcal{J}_k) \leq (1 + (\exp(L) - 1)(1 - \epsilon) + \epsilon)^{-1}. \tag{10.6}$$

The right-hand side can be further upper-bounded by $\exp(-L) + \epsilon$.

The separation condition (10.3) trivially holds for $\epsilon = 1$ and an arbitrary L, or for $L = 0$ and an arbitrary ϵ, though in those cases the bounds (10.5) and (10.6) are too loose. To be clear, we only use the closed form approximation (10.4) for theoretical assessment.

The next theorem bounds the utility gain from shifting model selection to stacking:

Theorem 10.3 (oracal expressiveness bound of stacking)**.** *Under the separation condition* (10.3)*, let $\rho = \sup_k \Pr(\mathcal{J}_k)$, and a deterministic function $g(L, K, \rho, \epsilon) = L(1 - \rho)(1 - \epsilon) - \log K$, then the utility gain of stacking is lower-bounded by*

$$\operatorname{elpd}_{\text{stacking,cp}} - \sup_k \operatorname{elpd}_k \geq \max\left(g(L, K, \rho) + O(\exp(-L) + \epsilon), 0\right).$$

Evaluating $\mathcal{J}_k$ requires access to $\tilde{y}|\tilde{x}$ and $\tilde{x}$. Though both terms are unknown, the roles of $\tilde{x}$ and $\tilde{y}$ are not symmetric: we could bespoke the model in preparation for a future prediction at a

given $\tilde{x}$, but cannot be tailored for a realization of $\tilde{y}$. To be more tractable, we consider the case when the variation on $\tilde{x}$ predominates the uncertainty of model comparison, such that $\mathcal{J}_k \approx \mathcal{I}_k \times \mathcal{Y}$, where $\mathcal{I}_k$ is defined in (10.1). More precisely, we define a strong local separation condition with a distance-probability pair (L, ϵ):

$$\sum_{k=1}^{K} \int_{\tilde{x}\in\mathcal{I}_k} \int_{\mathcal{Y}} \mathbb{1}\Big(\log p(\tilde{y}|M_k, \tilde{x}) < \log p(\tilde{y}|M_{k'}, \tilde{x}) + L,\ \forall k' \neq k \Big) p_t(\tilde{y}, \tilde{x}) d\tilde{y} d\tilde{x} \leq \epsilon. \tag{10.7}$$

We define $\rho_{\mathcal{X}} = \sup_k \Pr(\mathcal{I}_k)$. Under condition (10.7), $\rho_{\mathcal{X}}$ and ρ will be close. If we know the input space division $\{\mathcal{I}_k\}$, we can select model M_k for and only for $x \in \mathcal{I}_k$, which we call pointwise selection. The predictive density is

$$p(\tilde{y}|\tilde{x}, \mathcal{I}, \text{pointwise selection}) = \sum_{k=1}^{K} \mathbb{1}(\tilde{x} \in \mathcal{I}_k) p(\tilde{y}|\tilde{x}, M_k). \tag{10.8}$$

As per Theorem 10.3, for a given pair of L and ϵ, the smaller is ρ, the higher improvement ($K(1-\epsilon)(1-\rho)$) can stacking achieve against model selection: the situation in which no model always predominates. Thus, the effectiveness of stacking can indicate heterogeneity of model fitting. Next, we show that the heterogeneity of model fitting provides an additional utility gain if we shift from stacking to pointwise selection:

Theorem 10.4 (oracal expressiveness bound of pointwise selection)**.** *Under the strong separation condition* (10.7)*, and if the divisions* $\{\mathcal{I}_k\}$ *are known exactly, then the extra utility gain of pointwise selection has a lower bound,*

$$\text{elpd}_{\text{pointwise selection}} - \text{elpd}_{\text{stacking,cp}} \geq -\log \rho_{\mathcal{X}} + O(\exp(-L) + \epsilon).$$

For any input location $x_0 \in \mathcal{X}$, if we can approach the theoretically pointwise optimal $\hat{w}(x_0) \in \mathcal{S}_K$, then applying Theorem 10.3 to the space $\{x_0\}\times\mathcal{Y}$ suggests the advantage of pointwise averaging (9.4) against pointwise selection (10.8).

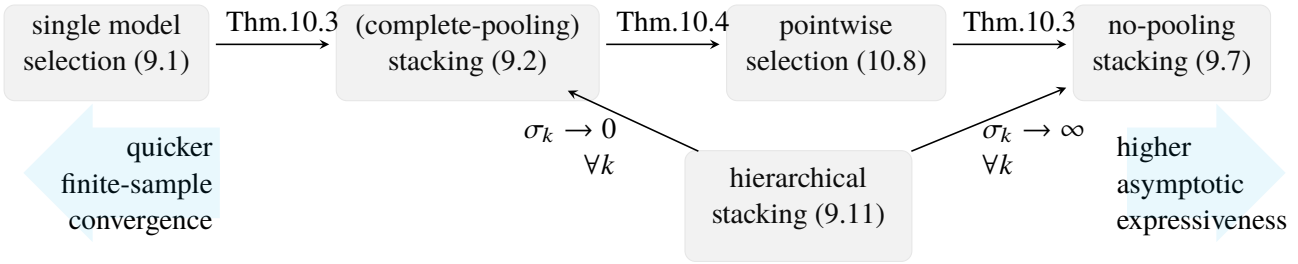


Figure 10.1: *Evolution of methods. First row from left to right: the methods have a higher degree of freedom to ensure a higher asymptotic predictive accuracy, the gain of which is bounded by the labeled theorems. Meanwhile, complex methods come with a slower convergence rate. The hierarchical stacking is a generalization of all remaining methods by assigning various structured priors, and adapts to the complexity-expressiveness tradeoff by hierarchical modeling.*

The relevance to the methodology evolution. The potential utility gain from Theorems 10.3 and 10.4 is the motivation behind the input-varying model averaging. Despite this asymptotic expressiveness, the finite sample estimate remains challenging. We do not know $\mathcal{I}_k$ or $\mathcal{J}_k$; We may use leave-one-out cross validation to estimate the overall model fit elpd_k, but in the pointwise version we want to assess conditional model performance—The more data coming in, the more input locations need to assess. The asymptotic expressiveness comes with an increasing complexity, as the free parameters in single model selection, complete-pooling stacking, pointwise selection, and no-pooling stacking are a single model index, a length-K simplex, an vector of pointwise model selection index $\{1, 2, \ldots, K\}^{\mathcal{X}}$, and a matrix of pointwise weight $(\mathcal{S}_K)^{\mathcal{X}}$. It is natural to apply the hierarchical shrinkage prior to handle this complexity-expressiveness tradeoff.

10.3 An illustrative example

Before theorem proofs, we first consider a theory example. It can be solved with closed form solution, and illustrates how the Theorems 10.1–10.4 apply. The construction also verifies some bounds we have derived are tight.

As shown in Figure 10.2, the true data generating process (DG) of the outcome is $y \sim \text{uniform}(-3, 1)$, and there are two given (pre-trained) models with spike-and-slab predictive distri-

butions

$$M_1 : y \sim .99 \text{ uniform}(-4, 0) + .01 \text{ uniform}(0, 2),$$
$$M_2 : y \sim .99 \text{ uniform}(0, 2) + .01 \text{ uniform}(-4, 0),$$

which yield piece-wise constant predictive densities

$$p_1(y) = 0.99/4\mathbb{1}(y \in [-4, 0]) + 0.01/2\mathbb{1}(y \in [0, 2]),$$
$$p_2(y) = 0.99/2\mathbb{1}(y \in [0, 2]) + 0.01/4\mathbb{1}(y \in [-4, 0]).$$

Using our notation in Section 10.2, the region in which M_1 predominates is $\mathcal{J}_1 = [-4, 0]$, and M_2 outperforms on $\mathcal{J}_2 = (0, 2]$ (the conventions send the tie $\{0\}$ to M_1). We count their masses with respect to the true DG: $\Pr(\mathcal{J}_1) = 3/4$ and $\Pr(\mathcal{J}_2) = 1/4$.

Complete-pooling stacking solves

$$\max_{w \in \mathcal{S}_2} \int_{-3}^{1} 1/4 \log\big((w_1 0.99/4 + w_2 0.01/4)\mathbb{1}(y \in [-4, 0]) + (w_2 0.99/2 + w_1 0.01/2)\mathbb{1}(y \in [0, 2])\big) dy.$$

The exact optimal weight is $w_1 = 0.755$, close to the mass $\Pr(\mathcal{J}_1) = 0.75$ and is irrelevant to the height of each regions. For instance, if the right bump in M_2 shrinks to the interval $(0, 1)$ (i.e., $y \sim 0.99$ uniform$(0, 1) + 0.01$ uniform$(-4, 0)$), then the winning margin therein is twice as big, while the winning probability as well as the stacking weight remains nearly unchanged.

At the pointwise level, stacking behaves as a plurality voting system: as long a model "wins" a sub-region (subject to a prefixed threshold L in condition (10.3)), the *winner take all* and its winning margin no long matters.

By contrast, likelihood based model averaging techniques such as Bayesian model averaging (BMA, Hoeting et al., 1999) and pseudo-Bayesian model averaging (Yao et al., 2018a) are analogies of *proportional representation*: every count of the winning margin matters. For illustration, we

vary the slab probability δ in Model 1 and 2:

$$M_1 \mid \delta : y \sim (1-\delta) \times \text{uniform}(-4, 0) + \delta \times \text{uniform}(0, 2),$$
$$M_2 \mid \delta : y \sim (1-\delta) \times \text{uniform}(0, 2) + \delta \times \text{uniform}(-4, 0).$$

The left column in Figure 10.3 visualizes the predictive densities from these two models at $\delta = 0.2$, 0.33, and 0.45.

When the slab probability δ increases from 0 to 0.5, these two models are closer and closer to each other, measured by a smaller $\text{KL}(M_1, M_2)$. The $(0.5, 1)$ counterpart is similar, though not exactly symmetric. We compute stacking weight and the expected pseudo-BMA weight with sample size n: $w_1^{\text{BMA}}(n, \delta) = \left(1 + \exp\left(n\,\mathbb{E}_{y|\delta} \log p_2(y) - n\,\mathbb{E}_{y|\delta} \log p_1(y)\right)\right)^{-1}$.

Interestingly, pseudo-BMA weight $w_1^{\text{BMA}}(n, \delta)$ is strictly decreasing as a function of $\delta \in (0, 1)$. This is because when $\delta \to 0^+$, log predictive density of model 2 in the left part $\log(\delta/4) \to \infty$ can be arbitrarily small, and the influence of this bad region dominates the overall performance of model 2. By contrast, stacking weight is monotonic non-increasing on $(0, 0.5)$ (strictly decreasing on $(0, 1/3)$, and remains flat afterwards)—the opposite direction of BMA. Stacking simply recognizes model 1 winning the $[-3, 0]$ interval and does not haggle over how much it wins.

In addition, when $\delta = 1/3$, M_1 becomes a uniform density on $[-4, 2]$. When $\delta \in (1/3, 1/2)$, model 2 is not only strictly worse than model 1, but also provides no extra information for model

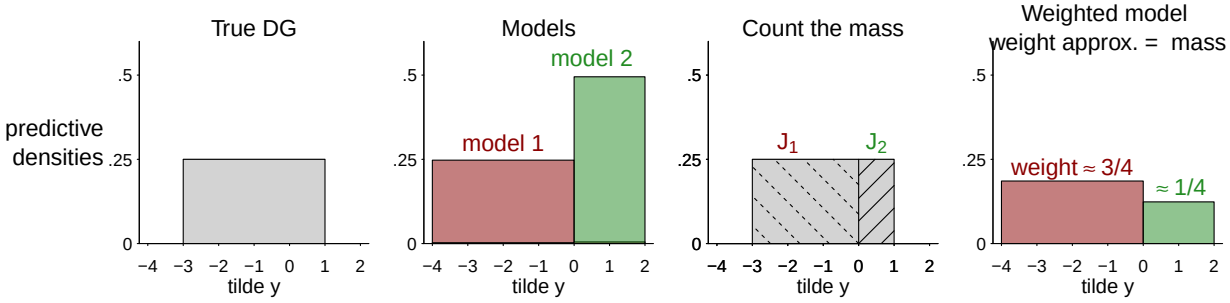


Figure 10.2: *The true data is generated from uniform*$(-3, 1)$ *and the there are two models with spikes and slabs on intervals* $(-4, 0)$ *and* $(0, 2)$ *respectively.* $\mathcal{J}_1$ *and* $\mathcal{J}_2$ *in Theorem 10.1 are* $[-4, 0]$ *and* $(0, 2]$*, with DG probabilities 3/4 and 1/4. The stacking weights are approximately these two probabilities, and irrelevant to how high the winning margins are.*

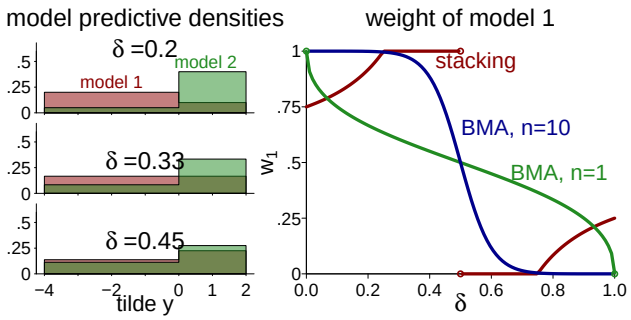


Figure 10.3: *Left: Pointwise predictive density $p(\tilde{y}|M_1$ or $M_2)$ when the slab probability δ is chosen 0.2, 1/3 and 0.45. Right: Weight of model 1 in complete-pooling stacking (not defined at $\delta = 0.5$) and pseudo-BMA (sample size n=1 or 10, not defined at $\delta = 0$ or 1) as a function of the slab probability δ. They evolve in the opposite direction. Besides, stacking weights are more polarized when models are more similar.*

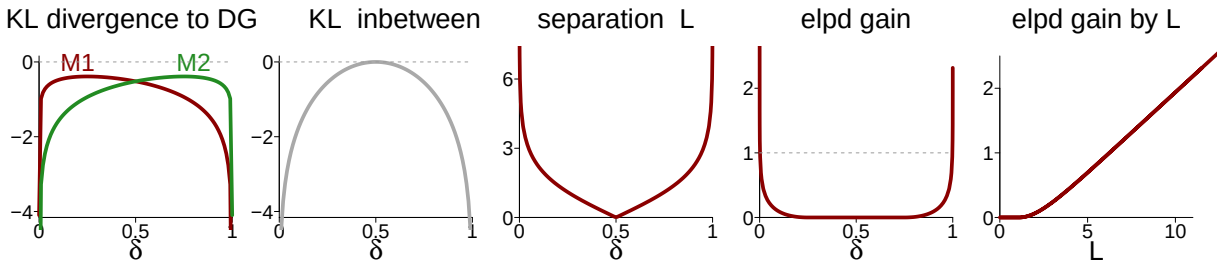


Figure 10.4: *From left: (1) KL divergence between model 1 or model 2 and data generating process. (2) KL divergence between model 1 and model 2. (3) Separation constant L. (4) Stacking elpd gain compared with the best individual model. (5) Stacking elpd gain as a function of L.*

averaging. Hence stacking assigns it weight zero.

The first two panels in Figure 10.4 show the KL divergence from model 1 or from model 2 to data generating process, and the KL divergence between model 1 and model 2. The third panel is the largest separation constant L for which the separation condition (10.3) holds. The last two panels show the stacking epld gain (compared with the best individual model) as a function of δ and L. This constructive example reflects the worst case for it matches the theoretical lower bound $g^*(L, K, \rho, \epsilon) = \log(\rho) + (1-\rho)(\log(1-\rho) - \log(K-1))$ (here $L = L, K = 2, \rho = 1/4, \epsilon = 0$) in Theorem 10.3, which verifies that the latter bound is tight.

When $\delta \in [1/3, 1/2)$, Model 2 still wins on the interval $\mathcal{J}_2 = (0, 2]$ with the separation constant $\epsilon = 0$ and $L \leq \log 2$ (the winning margin is maximized at $\delta = 1/3$). Nevertheless, a zero stacking weight and a non-zero winning area do not contradict Theorem 10.1. Indeed, Theorem 10.2 precisely bounds the mass of the winning region when stacking weight is zero. We provide self-contained theorem proofs in the next section.

10.4 Theoretical derivation

Theorem. *We call K predictive densities $\{p(\tilde{y} = \cdot|\tilde{x} = \cdot, M_k)\}_{k=1}^K$ to be locally separable with a constant pair $L > 0$ and $0 < \epsilon < 1$ with respect to the true data generating process $p_t(\tilde{y}, \tilde{x})$, if*

$$\sum_{k=1}^{K} \int_{(\tilde{x},\tilde{y})\in\mathcal{J}_k} \mathbb{1}\Big(\log p(\tilde{y}|\tilde{x}, M_k) < \log p(\tilde{y}|M_{k'}, x) + L,\ \forall k' \neq k \Big) p_t(\tilde{y}, \tilde{x}) d\tilde{y} d\tilde{x} \leq \epsilon.$$

For a small ϵ and a large L, the stacking weights that solve (9.3) *is approximately the proportion of the model being the locally best model:*

$$w_k^{\text{stacking}} \approx w_k^{\text{approx}} := \Pr(\mathcal{J}_k) = \int_{\mathcal{J}_k} p_t(\tilde{y}|\tilde{x}) p(\tilde{x}) d\tilde{y} d\tilde{x}.$$

in the sense that the objective function is nearly optimal:

$$|\text{elpd}(w^{\text{approx}}) - \text{elpd}(w^{\text{stacking}})| \leq O(\epsilon + \exp(-L)).$$

Proof. The expected log predictive density of the weighted prediction $\sum_k w_k p_k(\cdot|x)$ (as a function of w) is

$$\begin{aligned}
\text{elpd}(w) &= \int_{\mathcal{X}\times\mathcal{Y}} \log\left(\sum_{l=1}^{K} w_l p_l(\tilde{y}|\tilde{x})\right) p_t(\tilde{y}|\tilde{x})p(\tilde{x})d\tilde{x}d\tilde{y} \\
&= \sum_{k=1}^{K}\int_{\mathcal{J}_k} \log\left(\sum_{l=1}^{K} w_l p_l(\tilde{y}|\tilde{x})\right) p_t(\tilde{y}|\tilde{x})p(\tilde{x})d\tilde{x}d\tilde{y} \\
&= \sum_{k=1}^{K}\int_{\mathcal{J}_k} \log\left(w_k p_k(\tilde{y}|\tilde{x}) + \sum_{l\neq k} w_l p_l(\tilde{y}|\tilde{x})\right) p_t(\tilde{y}|\tilde{x})p(\tilde{x})d\tilde{x}d\tilde{y} \\
&= \sum_{k=1}^{K}\int_{\mathcal{J}_k} \left(\log\left(w_k p_k(\tilde{y}|\tilde{x})\right) + \log\left(1 + \sum_{l\neq k}\frac{w_l p_l(\tilde{y}|\tilde{x})}{w_k p_k(\tilde{y}|\tilde{x})}\right)\right) p_t(\tilde{y}|\tilde{x})p(\tilde{x})d\tilde{x}d\tilde{y}.
\end{aligned}$$

The expression is legit for any simplex vector $w \in \mathcal{S}$ that does not contain zeros. We will treat zeros later. For now we only consider a dense weight: $\{w \in \mathcal{S} : w_k > 0, k = 1, \ldots K\}$.

Consider a surrogate objective function (the first term in the integral above):

$$\begin{aligned}
\text{elpd}^{\text{surrogate}}(w) &= \sum_{k=1}^{K}\int_{\mathcal{J}_k} \log\left(w_k p_k(\tilde{y}|\tilde{x})\right) p_t(\tilde{y}|\tilde{x})p(\tilde{x})d\tilde{x}d\tilde{y} \\
&= \sum_{k=1}^{K}\int_{\mathcal{J}_k} \left(\log w_k + \log p_k(\tilde{y}|\tilde{x})\right) p_t(\tilde{y}|\tilde{x})p(\tilde{x})d\tilde{x}d\tilde{y} \\
&= \sum_{k=1}^{K} \log w_k \int_{\mathcal{J}_k} p_t(\tilde{y}|\tilde{x})p(\tilde{x})d\tilde{x}d\tilde{y} + \sum_{k=1}^{K}\int_{\mathcal{J}_k} \log p_k(\tilde{y}|\tilde{x}) p_t(\tilde{y}|\tilde{x})p(\tilde{x})d\tilde{x}d\tilde{y} \\
&= \sum_{k=1}^{K} \left(\Pr(\mathcal{J}_k)\log w_k\right) + \text{constant}.
\end{aligned}$$

Ignoring the constant term above (the expected cross-entropy between each conditional prediction and the true DG), to maximize the surrogate objective function is equivalent to maximize $\sum_{k=1}^{K}\Pr(\mathcal{J}_k)\log w_k$, we call this function elbo(w), the *evidence lower bound*. To optimize $\text{elpd}^{\text{surrogate}}$ is equivalent to optimize elbo. We show that this elbo function has a closed form

optimum. Using Jensen's inequality,

$$\begin{aligned}
\text{elbo}(w) &= \sum_{k=1}^{K} \Pr(\mathcal{J}_k) \log w_k \\
&= \sum_{k=1}^{K} \Pr(\mathcal{J}_k) \log \frac{w_k}{\Pr(\mathcal{J}_k)} + \sum_{k=1}^{K} \Pr(\mathcal{J}_k) \log \Pr(\mathcal{J}_k) \\
&\leq \log \left(\sum_{k=1}^{K} \Pr(\mathcal{J}_k) \frac{w_k}{\Pr(\mathcal{J}_k)} \right) + \sum_{k=1}^{K} \Pr(\mathcal{J}_k) \log \Pr(\mathcal{J}_k) \\
&= \sum_{k=1}^{K} \Pr(\mathcal{J}_k) \log \Pr(\mathcal{J}_k).
\end{aligned}$$

The equality is attained at $w_k = \Pr(\mathcal{J}_k),\ k = 1, \ldots, K$, which reaches our definition of w^{approx} in Theorem 10.1.

What remains to be proved is that the surrogate objective function is close to the actual objective. We divide each set $\mathcal{J}_k$ into two disjoint subsets $\mathcal{J}_k = \mathcal{J}_\circ \cup \mathcal{J}_k^\bullet$, for

$$\begin{aligned}
\mathcal{J}_k^\circ &\coloneqq \{(\tilde{x}, \tilde{y}) \in \mathcal{J}_k : \log p(\tilde{y}|M_k) < \log p(\tilde{y}|M_{k'}, x) + L\}\,; \\
\mathcal{J}_k^\bullet &\coloneqq \{(\tilde{x}, \tilde{y}) \in \mathcal{J}_k : \log p(\tilde{y}|M_k) \geq \log p(\tilde{y}|M_{k'}, x) + L\}\,.
\end{aligned}$$

The separation condition ensures $\sum_{k=1}^{K} \Pr(\mathcal{J}_k^\circ) \leq \epsilon$.

Let $\Delta(w) = \text{elpd}(w) - \text{elpd}^{\text{surrogate}}(w)$. For any fixed simplex vector w, this absolute difference

of the objective function is bounded by

$$
\begin{aligned}
|\Delta(w)| &= \left| \sum_{k=1}^{K} \int_{\mathcal{J}_k} \left(\log \left(1 + \sum_{l \neq k} \frac{w_l p_l(\tilde{y}|\tilde{x})}{w_k p_k(\tilde{y}|\tilde{x})} \right) \right) p_t(\tilde{y}|\tilde{x}) p(\tilde{x}) d\tilde{x} d\tilde{y} \right| \\
&\leq \sum_{k=1}^{K} \int_{\mathcal{J}_k} \left| \log \left(1 + \sum_{l \neq k} \frac{w_l p_l(\tilde{y}|\tilde{x})}{w_k p_k(\tilde{y}|\tilde{x})} \right) \right| p_t(\tilde{y}|\tilde{x}) p(\tilde{x}) d\tilde{x} d\tilde{y} \\
&= \sum_{k=1}^{K} \left(\int_{\mathcal{J}_k^\circ} + \int_{\mathcal{J}_k^\bullet} \right) \left| \log \left(1 + \sum_{l \neq k} \frac{w_l p_l(\tilde{y}|\tilde{x})}{w_k p_k(\tilde{y}|\tilde{x})} \right) \right| p_t(\tilde{y}|\tilde{x}) p(\tilde{x}) d\tilde{x} d\tilde{y} \\
&\leq \sum_{k=1}^{K} \int_{\mathcal{J}_k^\circ} \log(1 + \sum_{l \neq k} \frac{w_l}{w_k}) p_t(\tilde{y}|\tilde{x}) p(\tilde{x}) d\tilde{x} d\tilde{y} + \sum_{k=1}^{K} \int_{\mathcal{J}_k^\bullet} \sum_{l \neq k} \frac{w_l}{w_k} \frac{p_l(\tilde{y}|\tilde{x})}{p_k(\tilde{y}|\tilde{x})} p_t(\tilde{y}|\tilde{x}) p(\tilde{x}) d\tilde{x} d\tilde{y} \\
&\leq \left(\sum_{k=1}^{K} \sum_{l \neq k} \frac{w_l}{w_k} \right) \left(\sum_{k=1}^{K} \int_{\mathcal{J}_k^\circ} p_t(\tilde{y}|\tilde{x}) p(\tilde{x}) d\tilde{x} d\tilde{y} + \sum_{k=1}^{K} \int_{\mathcal{J}_k^\bullet} \frac{p_l(\tilde{y}|\tilde{x})}{p_k(\tilde{y}|\tilde{x})} p_t(\tilde{y}|\tilde{x}) p(\tilde{x}) d\tilde{x} d\tilde{y} \right) \\
&\leq \left(\sum_{k=1}^{K} \frac{1 - w_k}{w_k} \right) (\epsilon + \exp(-L)).
\end{aligned}
$$

The exact optima of objective function is w^{stacking}. Using the inequality above twice,

$$
\begin{aligned}
0 \leq \text{elpd}(w^{\text{stacking}}) - \text{elpd}(w^{\text{approx}}) &\leq |\text{elpd}^{\text{surrogate}}(w^{\text{stacking}}) - \text{elpd}(w^{\text{stacking}})| \\
&\quad + |\text{elpd}^{\text{surrogate}}(w^{\text{approx}}) - \text{elpd}(w^{\text{approx}})| \\
&\quad + \text{elpd}^{\text{surrogate}}(w^{\text{stacking}}) - \text{elpd}^{\text{surrogate}}(w^{\text{approx}}) \\
&\leq |\Delta(w^{\text{approx}})| + |\Delta(w^{\text{stacking}})| \\
&\leq \sum_{k=1}^{K} \left(\frac{1 - w_k^{\text{approx}}}{w_k^{\text{approx}}} + \frac{1 - w_k^{\text{stacking}}}{w_k^{\text{stacking}}} \right) (\epsilon + \exp(-L)).
\end{aligned}
$$

It almost finished the proof expect for the simplex edge where w_k^{stacking} or w_k^{approx} attains zero. Without loss of generality, if $w_1^{\text{approx}} = 0, w_k^{\text{approx}} \neq 0, \forall k \neq 1$, which means $p(\tilde{y}|M_k, \tilde{x})$ is always inferior to some other model. This will only happy if $p(\tilde{y}|M_k, \tilde{x})$ is almost sure zero (w.r.t $p_t(\tilde{y}, \tilde{x})$) hence we can remove model 1 from the list, and the same $O(\epsilon + \exp(-L))$ bound applies to remaining model $2, \ldots, K$. If there are more than one zeros, repeat until all zeros are removed.

Next, we deal with $w_1^{\text{stacking}} = 0, w_k^{\text{stacking}} \neq 0, \forall k \neq 1$. If $w_1^{\text{approx}} = 0$, too, then we have solved

in the previous paragraph. If not, Theorem 10.2 shows that w_1^{approx} has to be a small order term:

$$\Pr(\mathcal{J}_1) \le (1 + (\exp(L) - 1)(1 - \epsilon) + \epsilon)^{-1} < \exp(-L) + \epsilon.$$

We leave the proof of this inequality in Theorem 10.2.

The contribution of the first model in the surrogate model is at most $\Pr(\mathcal{J}_1) \log \Pr(\mathcal{J}_1)$. After we remove the first model from the model list, with the surrogate model elpd changes by at most a small order term, not affecting the final bound. Because the separation condition with constant (ϵ, L) applies to model $1, \ldots, K$, and due to lack of a competition source, the same separation condition applies to model $2, \ldots, K$ and the same bound applies.

Theorem. *When the separation condition* (10.3) *holds, and if the k-th model has zero weight in stacking, $w_k^{\text{stacking}} = 0$, then the probability of its winning region is bounded by:*

$$\Pr(\mathcal{J}_k) \le (1 + (\exp(L) - 1)(1 - \epsilon) + \epsilon)^{-1}.$$

The right hand side can be further upper-bounded by $\exp(-L) + \epsilon$.

Proof. Without loss of generality, assume $w_1^{\text{stacking}} = 0$. Let $p_0(\tilde{y}|\tilde{x}) = \sum_{k=2}^K w^{\text{stacking}} p_k(\tilde{y}|\tilde{x})$. Consider a constrained objective $\widetilde{\text{elpd}}(w_1) = \mathbb{E}(\log(w_1 p_1(\tilde{y}|\tilde{x}) + (1 - w_1) p_0(\tilde{y}|\tilde{x})))$ where the expectation is over both $\tilde{y}$ and $\tilde{x}$ as before. Because the max is attained at $w_1 = 0$ and because $\log(\cdot)$ is a concave function, the derivative at any $w_1 \in [0, 1]$ is

$$\frac{d}{dw_1} \widetilde{\text{elpd}}(w_1) = \mathbb{E}_{\tilde{y},\tilde{x}} \left(\frac{p_1(\tilde{y}|\tilde{x}) - p_0(\tilde{y}|\tilde{x})}{w_1 p_1(\tilde{y}|\tilde{x}) + (1 - w_1) p_0(\tilde{y}|\tilde{x})} \right) \le 0.$$

That is

$$
\begin{aligned}
0 \geq & \mathbb{E}\left(\frac{p_1(\tilde{y}|\tilde{x}) - p_0(\tilde{y}|\tilde{x})}{p_0(\tilde{y}|\tilde{x})}\right) \\
= & \Pr(\mathcal{J}_1)\,\mathbb{E}[\frac{p_1(\tilde{y}|\tilde{x}) - p_0(\tilde{y}|\tilde{x})}{p_0(\tilde{y}|\tilde{x})}|\mathcal{J}_1] + (1 - \Pr(\mathcal{J}_1))\,\mathbb{E}[\frac{p_1(\tilde{y}|\tilde{x}) - p_0(\tilde{y}|\tilde{x})}{p_0(\tilde{y}|\tilde{x})}|\mathcal{J}_0] \\
\geq & \Pr(\mathcal{J}_1)\,\mathbb{E}[\frac{p_1(\tilde{y}|\tilde{x}) - p_0(\tilde{y}|\tilde{x})}{p_0(\tilde{y}|\tilde{x})}|\mathcal{J}_1] - (1 - \Pr(\mathcal{J}_1)).
\end{aligned}
$$

Rearrange this inequality arrives at

$$
1 \geq \Pr(\mathcal{J}_1)\left(1 + \mathbb{E}[\frac{p_1(\tilde{y}|\tilde{x}) - p_0(\tilde{y}|\tilde{x})}{p_0(\tilde{y}|\tilde{x})}|\mathcal{J}_1]\right) \geq \Pr(\mathcal{J}_1)(1 + (\exp(L) - 1)(1 - \epsilon) + \epsilon).
$$

Hence, the model with weight zero cannot have a large probability to predominate all other models,

$$
\Pr(\mathcal{J}_1) \leq (1 + (\exp(L) - 1)(1 - \epsilon) + \epsilon)^{-1} < \exp(-L) + \epsilon.
$$

Theorem. *Let* $\rho = \sup_{1 \leq k \leq K} \Pr(\mathcal{J}_k)$*, and two deterministic functions* g *and* g^* *by*

$$
\begin{aligned}
& g(L, K, \rho, \epsilon) = L(1 - \rho)(1 - \epsilon) - \log K \\
\leq & g^*(L, K, \rho, \epsilon) = L(1 - \rho)(1 - \epsilon) + \rho \log(\rho) + (1 - \rho)(\log(1 - \rho) - \log(K - 1)).
\end{aligned}
$$

Assuming the separation condition (10.3) *holds for all* $k = 1, \ldots, K$*, then the utility gain of stacking is further lower-bounded by*

$$
\mathrm{elpd}_{\mathrm{stacking}} - \mathrm{elpd}_k \geq \max\left(g^*(L, K, \rho) + \mathcal{O}(\exp(-L) + \epsilon), 0\right).
$$

Proof. As before, we consider the approximate weights: $w_k^{\mathrm{approx}} = \Pr(\mathcal{J}_k)$, and the surrogate elpd

$\text{elpd}^{\text{surrogate}}(w) = \sum_{k=1}^{K} \int_{\mathcal{J}_k} \log\left(w_k p_k(\tilde{y}|\tilde{x})\right) p_t(\tilde{y}|\tilde{x}) p(\tilde{x}) d\tilde{x} d\tilde{y}.$

$$
\begin{aligned}
&\text{elpd}^{\text{surrogate}}(w^{\text{approx}}) - \text{elpd}_k \\
=&\sum_{l=1}^{K} \int_{\mathcal{J}_l} \log\left(\Pr(\mathcal{J}_l) p_l(\tilde{y}|\tilde{x})\right) p_t(\tilde{y}|\tilde{x}) p(\tilde{x}) d\tilde{x} d\tilde{y} - \sum_{l=1}^{K} \int_{\mathcal{J}_l} \log\left(p_k(\tilde{y}|\tilde{x})\right) p_t(\tilde{y}|\tilde{x}) p(\tilde{x}) d\tilde{x} d\tilde{y} \\
=&\sum_{l=1}^{K} \int_{\mathcal{J}_l} \left(\log \Pr(\mathcal{J}_l) + \log p_l(\tilde{y}|\tilde{x}) - \log p_k(\tilde{y}|\tilde{x})\right) p_t(\tilde{y}|\tilde{x}) p(\tilde{x}) d\tilde{x} d\tilde{y} \\
=&\sum_{l=1}^{K} \Pr(\mathcal{J}_l) \log \Pr(\mathcal{J}_l) + \sum_{l=1}^{K} \mathbb{1}(l \neq k) \int_{\mathcal{J}_l} \log(p_l(\tilde{y}|\tilde{x}) - \log p_k(\tilde{y}|\tilde{x})) p_t(\tilde{y}|\tilde{x}) p(\tilde{x}) d\tilde{x} d\tilde{y} \\
=&\sum_{l=1}^{K} \Pr(\mathcal{J}_l) \log \Pr(\mathcal{J}_l) + \sum_{l=1}^{K} \mathbb{1}(l \neq k) \left(\int_{\mathcal{J}_l^{\circ}} + \int_{\mathcal{J}_l^{\bullet}}\right) \log(p_l(\tilde{y}|\tilde{x}) - \log p_k(\tilde{y}|\tilde{x})) p_t(\tilde{y}|\tilde{x}) p(\tilde{x}) d\tilde{x} d\tilde{y} \\
\geq&\sum_{l=1}^{K} \Pr(\mathcal{J}_l) \log \Pr(\mathcal{J}_l) + (1-\epsilon)(1-\rho) L - \epsilon \\
\geq&\rho \log \rho + (1-\rho) \log \frac{1-\rho}{K-1} + (1-\epsilon)(1-\rho) L - \epsilon \\
=&g(L, K, \rho, \epsilon) - \epsilon.
\end{aligned}
$$

The last inequality comes from the fact that the entropy term $\sum_{l=1}^{K} \Pr(\mathcal{J}_l) \log \Pr(\mathcal{J}_l)$ attains its minimal at $(\rho, (1-\rho)/(K-1), \ldots, (1-\rho)/(K-1))$ due to the convexity of $x \log x$. We may further use a lower bound $\rho \log \rho + (1-\rho) \log 1 - \rho/K - 1 \geq -\log(K)$, which arrives in $g(\cdot)$.

Finally, using the proof of Theorem 10.1,

$$|\text{elpd}^{\text{surrogate}}(w^{\text{approx}}) - \text{elpd}^{\text{stacking}}(w^{\text{stacking}})| \leq O(\exp(-L) + \epsilon),$$

we get $\text{elpd}_{\text{stacking}} - \text{elpd}_k \geq g^*(L, K, \rho) + O(\exp(-L) + \epsilon)$. Because selection is always a specials case of averaging, the utility is further bounded below by 0.

From the constrictive example (Appendix 10.3), $g^*(\cdot)$ is a tight bound. We use the looser bound $g(\cdot)$ in the main chapter for its simpler form.

Theorem. *Under the strong separation assumption*

$$\sum_{k=1}^{K} \int_{\tilde{x}\in\mathcal{I}_k} \int_{\tilde{y}\in\mathcal{Y}} \mathbb{1}\Big(\log p(\tilde{y}|M_k, x) < \log p(\tilde{y}|M_{k'}, x) + L,\ \forall k' \neq k^*(x)\Big) p_t(\tilde{y}|x, D) d\tilde{y} \leq \epsilon,$$

and if the sets $\{\mathcal{I}_k\}$ *are known exactly, then we can construct pointwise selection*

$$p(\tilde{y}|x, \text{pointwise selection}) = \sum_{k=1}^{K} \mathbb{1}(x \in \mathcal{I}_k) p(\tilde{y}|x, M_k).$$

Its utility gain is bounded from below by

$$\text{elpd}_{\text{pointwise selection}} - \text{elpd}_{\text{stacking}} \geq -\log \rho_{\mathcal{X}} + O(\exp(-L) + \epsilon).$$

Proof.

$$\begin{aligned}
&\text{elpd}_{\text{pointwise selection}} - \text{elpd}^{\text{surrogate}}(w^{\text{approx}}) \\
&= \sum_{l=1}^{K} \int_{\mathcal{I}_l} \left(\log p_l(\tilde{y}|\tilde{x}) - \log\left(\Pr(\mathcal{I}_l) p_l(\tilde{y}|\tilde{x})\right)\right) p_t(\tilde{y}|\tilde{x}) p(\tilde{x}) d\tilde{x} d\tilde{y} \\
&= -\sum_{l=1}^{K} \Pr(\mathcal{I}_l) \log \Pr(\mathcal{I}_l) \\
&\geq -\sum_{l=1}^{K} \Pr(\mathcal{I}_l) \log \rho_x \\
&= -\log \rho_x.
\end{aligned}$$

Finally, from the proof of Theorem 10.1,

$$|\text{elpd}^{\text{surrogate}}(w^{\text{approx}}) - \text{elpd}^{\text{stacking}}(w^{\text{stacking}})| \leq O(\exp(-L) + \epsilon).$$

Conclusion and future directions

A. Alternatives to Bayes rule (can we learn a better epistemic uncertainty)

As we have previously discussed in Chapter 8, epistemic uncertainty comes into inference and prediction due to a finite amount of data. Given a model and an unknown parameter θ, the epistemic uncertainty is reflected by the posterior distribution $p(\theta|y)$. When inference is conducted through MCMC methods, we have an extra Monte Carlo error when computing any posterior integrals with respect to these destines.

Post-processing has long been used to shrink the Monte Carlo or approximation errors. In Chapter 5 and Chapter 6, we explore Rao-Blackwellization and path sampling that reduce Monte Carlo error for any normalization constant estimate. In Chapter 3, we post-process variational approximation draws using PSIS to assess its accuracy.

However, to recover the exact posterior $p(\theta|y)$ is not the ultimate goal. The "full Bayes" solution is only optimal under the true model and when averaging over the prior distribution. Being open-minded to model misspecification, it will be more desired to find some "good" probabilistic inference, denoted by $\tilde{p}(\theta|y)$, such that the posterior prediction $p(\tilde{y}|y) = \int \tilde{p}(\tilde{y}|\theta)p(\theta|y)d\theta$ is "good". We have mathematically defined this framework in equation (8.2).

Along with this idea, a second type of post-processing is designed to re-calibrating Bayesian epistemic uncertainty. In Chapter 8, we use stacking to reweigh separated component in the posterior density $\tilde{p}(\theta|y, w) = \sum_k w_k p_k^{\text{Bayes}}(\theta|y)$. In Chapter 6, we consider a tempered modification: $\tilde{p}(\theta|y, \lambda) \propto p^{\text{Bayes}}(\theta|y)^{\lambda}$. We will discuss more aggregation forms in Problem D.

When we initially developed this method, we were sometimes questioned by "Is it a Bayes procedure?". We responded to them by prioritizing Bayesian decision theory against Bayesian inference. Such tensions can only be resolved by considering Bayesian logic as a tool, a way of

revealing inevitable misfits and incoherences in our model assumptions, rather than as an end in itself.

An alternative, yet gradually more preferred, defense is through a model expansion perspective. Take stacking for example: Yes, BMA is full Bayes, but stacking is full Bayes inference too if we consider the augmented model $p(\theta, w, y)$ defined in equation (9.11). Likewise in tempering, an orthodox Bayesian may only admit the validity of $p(\theta|y)$, the point at which temperature λ is a constant one. But as soon as we expand the model by $p(\theta, \lambda, y) \propto c(\lambda)p(\theta, y)^{\lambda}$, the posterior of temperature becomes a valid inference, and collecting all draws of λ draws amounts to a BMA solution that average over temperatures, whereas the original posterior $p(\theta|y)$ becomes a narrow slice (type-II MAP) of the extended model that ignores the variation on temperature. From this point of view, post-processing techniques such as chain-stacking and full temperature trajectory do not sabotage the Bayes' rule—they expand the model.

Related to this idea, stacking is not only a model averaging tool, but also an inference paradigm, sitting in parallel to Bayes and empirical Bayes. Consider one *single* model with data y and parameter θ. There are three related paradigms to draw inference:

- In a full-Bayes view, the posterior inference has to be

$$\log p(\theta|y) = \sum_i \log p(y_i|\theta) + \log p_{\text{prior}}(\theta) + \text{Constant}.$$

- From the (optimization-based) stacking point of view, the goal of inference is to better fit the data with respect to some predictive metric. Equipped with the leave-one-out log score, stacking estimates the best parameter by

$$\hat{\theta} = \arg\max_{\theta} \sum_i \log p(y_i|y_{-i}, \theta) + \log p_{\text{prior}}(\theta).$$

  Yao et al. (2020b) discussed this stacking-as-single-model-training idea in the context of multimodal posterior, in which full-Bayes is often overconfident. This approach is also

related to empirical Bayes.

- Yao et al. (2021b) bridges these two ends. The Bayesian version of stacking can be viewed as an single-model-inference defined by the modified log posterior density:

$$\log p(\theta|y) = \sum_i \log p(y_i|y_{-i}, \theta) + \log p_{\text{prior}}(\theta) + \text{Constant}.$$

Can we apply the redefined posterior from the last line to general inference problems?

B. $\mathcal{M}$-views for data collection (where to sample observations)

So far we have not directly studied the data inquiry part of the workflow: How to design the experiment? Where to sample observations? Should all data be collected once, or sequentially?

However, we implicitly encounter similar challenges several times. Chapter 2 discusses importance sampling, one foundation of transfer learning and importance weighted active learning. Chapter 4 discusses covariate imbalance, an example in which sampling distribution determiners the convergence rate of the treatment effect estimate. $\hat{k}$ is thereby a useful quantity for experiment design. More relevantly, in Chapter 6, the simulated tempering consists of an active learning task: how to distribute the free energy, the margin of inverse temperature, or what we called "prior" in that context. We have argued that there is an efficiency-robustness trade-off, with three distinct goals (Section 6.2) to optimize and we derive the optimal temperature margin respectively.

This tradeoff applies to many other experiment design problems. Take linear regression for example: assume the observation is (x, y) pair with $x \in [-1, 1]$, and we work with a regression model, $y_i = \beta x_i + \text{normal}(0, \sigma)$. Suppose we also know that in the population of interest, the input is x ~Uniform[-1,1]. Similar to the three goals in Section 6.2, here we may also adopt three views:

1. *Experiment design or an $\mathcal{M}$-closed view.* The objective is to minimize the variance of estimate $\hat{\beta}$, which is achieved by placing all masses of x at the two end-points $p_{\text{train}}(x) = .5\delta(-1) + .5\delta(1)$.

2. *Active learning or an $\mathcal{M}$-complete view.* Adopting a covariate shift perspective, we reweight the model to reflect the difference between training and testing covariates. That is to reweigh the log likelihood by $w_i = p_{\text{test}}(x_i)/p_{\text{train}}(x_i)$. Placing x only on -1 and 1 leads to an infinite variance on w. If the linear model is *approximately correct* such that the bias is asymptotically ignoble, we can use active learning techniques (Kanamori and Shimodaira, 2003; Sugiyama and Ridgeway, 2006) to derive the optimal p_{train} that minimizes the asymptotic variance of the importance weighted estimate: $p_{\text{train}}(x) \propto |x|$.

3. *Causal inference or an $\mathcal{M}$-open view.* In the derivation above, we still need some belief model and have to assume it to be asymptotically unbiased. With a minimal assumption view, we would like to hedge against minimax risks: what if the true model is $y = x$, except inside a tiny interval $x \in [-0.01, 0.01]$, $y = 100x$? Then the previous design will miss this part. The most conservative/minimax design is to minimize the variance of importance weights, which is 0 and is achieved at $p_{\text{train}} = p_{\text{test}} = \text{Uniform}[-1, 1]$.

In continuous tempering (Chapter 6), we adopt the third view with a little twist: the perfect identification of test and training distribution might be too demanding. As long as the importance weight has a small tail shape parameter, we are able to extrapolate training to test date, akin to the causal reasoning in Chapter 4.

In practice, we have more complex models than linear regression, and designing the optimal sampling distribution remains challenging. The reasoning above hints a viable direction to balance the robustness-efficiency tradeoff: to seek the most "efficient" training sample such that the tail shape of the importance ratios is controlled.

C. Bayesian update (when to jump out of the loop)

The decision framework would be more completed with additional sequential components. The workflow has several adaptive module including iterative model building based on previous fitting, early termination of computing when too slow, sequential data collection. When the model is fixed,

the data-coming-sequentially problem fits in the classic online learning framework where we update parameters. However, because of the persistent doubt on model specification, we would also like to update the model on the fly. When should we dynamically update or abandon the current model?

To be concrete, consider a practical challenge in election forecast (we have explored a full Bayesian model of election forecast in Chapter 9). During the election night, the state-level election outcome comes in sequence, making a real-time update of the prediction useful after some state results have been revealed.

We have two general approaches. Approach 1 is a full Bayesian solution. After fitting all polling data, the model provides a *joint* posterior predictive density of the state-level election result, say the Democratic share of among two parties, which we now denote by Pr(CA, NY ...) such that the dependence on previously fitted polling data is suppressed. An online update is defined by a conditional distribution: Pr(CA | NY = observed outcome). Similar to approximate Bayes computation, we can adopt some distance metric, and collect posterior simulation draws from the previous model fitting evaluate Pr(CA | NY $\approx$ observed outcome) using Monte Carlo draws. Alternatively, we can approximate joint unconstrained posterior density by a multivariate normal, such that any conditional update comes in a closed-form solution.

An orthogonal approach to this update is an extra regression model: to fit the actual state-level outcome using the model-based prediction as an input. For example, a linear regression: $\text{outcome}_{\text{state } i} = \beta_1 \text{prediction}_{\text{state } i} + \beta_0 + \epsilon_i$, $\epsilon_i \sim$ iid error. The independent error may be improved by a correlated multivariate distribution to account for state covariance.

These two approaches differ in how much trust they have in the model-based prediction. Approach 1 learns the systematic "polling bias" from posterior correlations, which amounts to Approach 2 with fixed $\beta_1 = 1$ and fixed residual distribution. Approach 2 can learn the systematic bias faster, but also needs more data to fit the regression model.

In this thesis, we have implicitly made this type of model update: we use stacking to recombine multimodal posterior, effectively changing the working model after detecting model misspecification. In general, when model fitting, building, and data collection come sequentially and

interactively, it requires more thinking or perhaps an extra meta-modeling on when to update the current working model.

D. Operationalize the model space (how to combine distributions)

With input $x \in \mathbb{R}^d$ and output $y \in \mathbb{R}^m$, we know how to learn $y = f(x)$ using various regression techniques. But what if the input source is a *distribution* rather than a *number*?

This is essentially the challenge we are facing in model averaging and model expansion. For each data point, we have k probabilistic predictions $p_k(\tilde{y})$ (for brevity we will ignore covariates for now). In (Bayesian) stacking (Section 7), we combine distributions by a *mixture*:

$$p(\cdot|w) = \sum_k w_k p_k(\cdot). \tag{11.1}$$

In tempering (Section 6) or distillation (Hinton et al., 2015), we combine distributions through a *geometry bridge*:

$$p(\cdot|w) \propto \exp\left(\sum_k w_k \log p_k(\cdot)\right).$$

In point ensembles (stacking, boosting, Bayesian predictive synthesis), we combine distributions via a *convolution* (additive model of $r.v.$):

$$\cdot|w \stackrel{d}{=} \sum_k w_k Y_k, \;\; Y_k \sim p_k(\cdot).$$

There are certainly other ways to operationalize distributions such as superposition. The choice of these operations impacts what model evaluation metric we can comfortably use. Under convolutions, the L^2 error is easy to compute, while the predictive density is hard to evaluate. Using the mixture form, the log predictive density of the combined model comes directly. These combinations of operation-form f and scoring-rule S give different aggregation method: $\max_w \mathbb{E}_y \, S(f(p_1(y), \ldots, p_k(y)|w), y)$. For example, the quantile regression averaging (Maciejowska et al., 2016) can be viewed as stacking with interval scores and convolution form.

By optimizing parameters w, we say stacking is asymptotically optimal among mixtures. Yet how can we also optimize over operations f? What is the space of operations? What is the most general combination form?

The support of these expansions is often large: a mixture of normals can approximate anything, and yet a convolution of normals can also approximate anything. To study the subtle difference of these "paths", it might useful to start with frequentist properties. For example, Kamary et al. (2019) convert hypothesis testing into a framing a mixture model that encompasses individual models as special cases, and test if some w_k in (11.1) is zero. Now that we have various paths, do they differ in terms of the convergence rate of the hypothesis testing under the null? What is the optimal operation?

E. Meta-learning (how would an AI learn statistics)

In this thesis, we advertise stacking/tempering as a semi-automatic model building/expansion tool: when we are intellectually tired after building and fitting a sequence of models, we run stacking to obtain a new model to try. Nevertheless, this procedure cannot replace all human labors in the workflow as the mixture path (11.1) explores only a limited class of models. Beyond current model aggregation tools, can we develop an automated ensemble learner that could fully explore and expand the space of models—for example, using an autoregressive (AR) model and a moving-average (MA) model to learn an ARMA model? Or using an inverse probability weighting and a regression model to learn a doubly-robust regression?

These tasks are the ultimate goal that we would want an AI, who can *learn statistics* rather than *learn parameters*, to achieve. We conjecture that stacking or hierarchical stacking might be a good starting point toward this direction, but there are plenty of rooms for improvement. Perhaps it is helpful to rethink what we are really learning when we "learn new models".

There are three levels of what a "model" stands for:

1. *A black-box prediction machine.* We feed the model with training data $\mathcal{D}$ and future input $\tilde{x}$, and the model will output the outcome prediction $p(\tilde{y}|\tilde{x}, \mathcal{D})$. This "black-box" mapping is all

that stacking is using: we do not have to know what parameters are involved in each model.

2. *A joint probability distribution.* A parametric model is mathematically defined by a joint probability distribution $p(\theta, y)$. If we are fitting mixture models or mixture of experts, the model aggregation and individual models are fitted simultaneously. Think about the probit-logit model expansion in Chapter 6. They are defined by two joint densities: $p_1(\beta_{\text{probit}}, y) = \prod_i \text{Bernoulli\_probit}(y_i|x_i\beta_{\text{probit}})p(\beta_{\text{probit}})$, and $p_2(\beta_{\text{logit}}, y) = \prod_i \text{Bernoulli\_logit}(y_i|x_i\beta_{\text{logit}})p(\beta_{\text{logit}})$. Stacking simply ignores parameters while tempering links two separated parameter spaces by

$$p(y, \beta_{\text{logit}}, \beta_{\text{probit}}|\lambda) \propto p_1^{\lambda}(\beta_{\text{logit}}, y)p_2^{1-\lambda}(\beta_{\text{probit}}, y).$$

3. *A joint probability distribution and parameters have meanings*. More importantly, the black-box prediction $p(\tilde{y}|\tilde{x}, \mathcal{D})$ ignores how the model is assembled by likelihoods and priors, while some modules are shared across models. In the probit-logit example, knowing these two models only differ in link functions, we would combine the model structure by merging the functionally similar parameters $(\beta_{\text{probit}}, \beta_{\text{logit}}) \rightarrow \beta_{\text{new}}$,

$$p(y, \beta_{\text{new}}|\lambda) \propto p_1^{\lambda}(\beta_{\text{new}}, y)p_2^{1-\lambda}(\beta_{\text{new}}, y).$$

Using this path in Figure 6.10, we argue that it is even more computationally efficient than fitting two models separately. If we have further prior knowledge on logit and probit regressions, we can do even better by

$$p(y, \beta_{\text{new}}|\lambda) \propto p_1^{\lambda}(\beta_{\text{new}}, y)p_2^{1-\lambda}(1.6 \times \beta_{\text{new}}, y).$$

Sharing parameters during model averaging enhances computation efficiency (Figure 6.10), and is a necessary step toward *model understanding*. Here we only illustrate geometric bridges, while this idea can be energized to all other operations discussed in Problem D.

All three views are useful. View 3 is arguably toward an interpretable machine learning: the model is an explanation of nature. Unless learning the meaning of parameters as well, an AI would not automatically name the model "ARMA" even after learning it from combining an AR and MA model. On the other hand, View 1 can be useful as a high-level implementation. For example, it is less meaningful to fit a mixer model or a mixture of experts on unstable computation results, random initialization, non-mixing computation, or bootstrapped runs.

In order to apply View 3 to a general model expansion/averaging setting, a sequence of open questions arises for future investigation: How can we detect similar modulus from different models? Perhaps we characterize this similarity by hierarchical modeling and partially pool across models? How can we encourage these functionally similar parameters living in different models to talk? Can we use this template in the model fitting phase such that we can have a warm start for new models during the workflow loop?

References

Abramowitz, A. I. (2008). Forecasting the 2008 presidential election with the time-for-change model. *Political Science and Politics*, 41:691–695. 210

Adams, J., Bishin, B. G., and Dow, J. K. (2004). Representation in congressional campaigns: Evidence for discounting/directional voting in U.S. Senate elections. *Journal of Politics*, 66(2):348–373. 128

Agapiou, S., Papaspiliopoulos, O., Sanz-Alonso, D., and Stuart, A. (2017). Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, 32(3):405–431. 15, 44

Agrawal, A., Fu, W., and Menzies, T. (2018). What is wrong with topic modeling? And how to fix it using search-based software engineering. *Information and Software Technology*, 98:74–88. 159

Anderson, J. L. (1996). A method for producing and evaluating probabilistic forecasts from ensemble model integrations. *Journal of Climate*, 9:1518–1530. 28

Angelino, E., Johnson, M. J., and Adams, R. P. (2016). Patterns of scalable Bayesian inference. *Foundations and Trends in Machine Learning*, 9:119–247. 150

Bafumi, J., Gelman, A., Park, D. K., and Kaplan, N. (2005). Practical issues in implementing and understanding Bayesian ideal point estimation. *Political Analysis*, 13:171–187. 143

Berger, J. O. and Pericchi, L. R. (1996). The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association*, 91:109–122. 113

Berk, R. H. (1966). Limiting behavior of posterior distributions when the model is incorrect. *Annals of Mathematical Statistics*, 37:51–58. 143

Bernardo, J. M. and Smith, A. F. (1994). *Bayesian Theory*. John Wiley & Sons. 105, 107, 116, 220

Bertsimas, D. and Tsitsiklis, J. (1993). Simulated annealing. *Statistical Science*, 8:10–15. 138

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M., and Stuart, A. (2013). Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli*, 19:1501–1534. 141

Betancourt, M. (2015). Adiabatic Monte Carlo. *arXiv:1405.3489*. 83, 84

Betancourt, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. *arXiv:1701.02434*. 60

Betancourt, M. and Girolami, M. (2015). Hamiltonian Monte Carlo for hierarchical models. In Upadhyay, S. K., Singh, U., Dey, D. K., and Loganathan, A., editors, *Current Trends in Bayesian Methodology with Applications*, pages 79–101. CRC Press. 137, 168

Bhatnagar, N. and Randall, D. (2004). Torpid mixing of simulated tempering on the Potts model. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 478–487. 99, 138

Bhatt, S., Cameron, E., Flaxman, S. R., Weiss, D. J., Smith, D. L., and Gething, P. W. (2017). Improved prediction accuracy for disease risk mapping using Gaussian process stacked generalization. *Journal of The Royal Society Interface*, 14. 197

Bishop, C. M., Lawrence, N. D., Jaakkola, T., and Jordan, M. I. (1998). Approximating posterior distributions in belief networks using mixtures. In *Advances in Neural Information Processing Systems*, pages 416–422. 151

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112:859–877. 21, 145, 172

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022. 143, 159

Blondel, A. (2004). Ensemble variance in free energy calculations by thermodynamic integration: theory, optimal "alchemical" path, and practical solutions. *Journal of Computational Chemistry*, 25:985–993. 67

Bou-Rabee, N., Eberle, A., and Zimmer, R. (2020). Coupling and convergence for Hamiltonian Monte Carlo. *Annals of Applied Probability*, 30:1209–1250. 141

Breiman, L. (1996a). Bagging predictors. *Machine learning*, 24:123–140. 114

Breiman, L. (1996b). Stacked regressions. *Machine Learning*, 24:49–64. 109, 119, 123, 139, 197, 198, 222

Breiman, L. (2001). Statistical modeling: The two cultures. *Statistical Science*, 16:199–231. 216

Bugallo, M. F., Elvira, V., Martino, L., Luengo, D., Miguez, J., and Djuric, P. M. (2017). Adaptive importance sampling: The past, the present, and the future. *IEEE Signal Processing*, 34:60–79. 97

Bürkner, P.-C., Gabry, J., and Vehtari, A. (2020). Approximate leave-future-out cross-validation for Bayesian time series models. *Journal of Statistical Computation and Simulation*, 90:2499–2523. 118, 194

Carlson, D., Stinson, P., Pakman, A., and Paninski, L. (2016). Partition functions from Rao-Blackwellized tempered sampling. In *Proceedings of the 33rd International Conference on Machine Learning*. 64, 86, 88

Carreño, L. V. G. and Winbladh, K. (2013). Analysis of user comments: An approach for software requirements evolution. In *35th International Conference on Software Engineering*, pages 582–591. 160

Chang, O., Yao, Y., Williams-King, D., and Lipson, H. (2019). Ensemble model patching: A parameter-efficient variational Bayesian neural network. *arXiv:1905.09453*. 152

Chatterjee, S. and Diaconis, P. (2018). The sample size required in importance sampling. *Annals of Applied Probability*, 28:1099–1135. 15, 42, 44, 64

Chee, J. and Toulis, P. (2018). Convergence diagnostics for stochastic gradient descent with constant learning rate. In *International Conference on Artificial Intelligence and Statistics*, pages 1476–1485. 21

Chen, L. H. and Shao, Q.-M. (2004). Normal approximation under local dependence. *Annals of Probability*, 32(3):1985–2028. 11, 18, 24, 42

Chen, M.-H. (1994). Importance-weighted marginal bayesian posterior density estimation. *Journal of the American Statistical Association*, 89:818–824. 70

Chung, Y., Rabe-Hesketh, S., Dorie, V., Gelman, A., and Liu, J. (2013). A nondegenerate penalized likelihood estimator for variance parameters in multilevel models. *Psychometrika*, 78:685–709. 170

Clarke, B. (2001). Combining model selection procedures for online prediction. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 229–249. 114

Clarke, B. (2003). Comparing Bayes model averaging and stacking when model approximation error cannot be ignored. *Journal of Machine Learning Research*, 4:683–712. 111, 133, 139, 178, 197, 221, 222

Clyde, M. and Iversen, E. S. (2013). Bayesian model averaging in the M-open framework. In *Bayesian Theory and Applications*, pages 483–498. Oxford University Press. 109, 139, 178, 197

Cook, S. R., Gelman, A., and Rubin, D. B. (2006). Validation of software for Bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, 15:675–692. 28

Cortes, C., Greenberg, S., and Mohri, M. (2019). Relative deviation learning bounds and generalization with unbounded loss functions. *Annals of Mathematics and Artificial Intelligence*, 85:45–70. 24

Cortes, C., Mansour, Y., and Mohri, M. (2010). Learning bounds for importance weighting. In *Advances in Neural Information Processing Systems*, pages 442–450. 24, 42

Cotter, S. L., Roberts, G. O., Stuart, A. M., and White, D. (2013). MCMC methods for functions: Modifying old algorithms to make them faster. *Statistical Science*, 28:424–446. 141

Crump, R. K., Hotz, V. J., Imbens, G. W., and Mitnik, O. A. (2009). Dealing with limited overlap in estimation of average treatment effects. *Biometrika*, 96:187–199. 47

Dawid, A. P. (1984). Present position and potential developments: Some personal views: Statistical theory: The prequential approach. *Journal of the Royal Statistical Society: Series A*, pages 278–292. 117

Diaconis, P. and Freedman, D. (1986a). On inconsistent Bayes estimates of location. *Annals of Statistics*, 14:68–87. 159

Diaconis, P. and Freedman, D. (1986b). On the consistency of Bayes estimates. *Annals of Statistics*, 14:1–26. 159

Dieng, A. B., Tran, D., Ranganath, R., Paisley, J., and Blei, D. (2017). Variational inference via χ upper bound minimization. In *Advances in Neural Information Processing Systems*, pages 2729–2738. 38

Dwivedi, R., Chen, Y., Wainwright, M. J., and Yu, B. (2018). Log-concave sampling: Metropolis-Hastings algorithms are fast! In *Conference on Learning Theory*, pages 793–797. 141

D'Amour, A., Ding, P., Feller, A., Lei, L., and Sekhon, J. (2020). Overlap in observational studies with high-dimensional covariates. *Journal of Econometrics*. 40

Earl, D. J. and Deem, M. W. (2005). Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7:3910–3916. 138

Epifani, I., MacEachern, S. N., and Peruggia, M. (2008). Case-deletion importance sampling estimators: Central limit theorems and related results. *Electronic Journal of Statistics*, pages 774–806. 17, 24

Finch, S. J., Mendell, N. R., and Thode Jr, H. C. (1989). Probabilistic measures of adequacy of a numerical search for a global maximum. *Journal of the American Statistical Association*, 84:1020–1023. 151

Fokoue, E. and Clarke, B. (2011). Bias-variance trade-off for prequential model list selection. *Statistical Papers*, 52:813–833. 118, 123

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139. 135

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, pages 1189–1232. 135

Fushiki, T. (2020). On the selection of the regularization parameter in stacking. *Neural Processing Letters*, pages 1–12. 198

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., and Gelman, A. (2019). Visualization in Bayesian workflow. *Journal of the Royal Statistical Society: Series A*, 182:389–402. 180

Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1050–1059. 152

Geisser, S. and Eddy, W. F. (1979). A predictive approach to model selection. *Journal of the American Statistical Association*, 74:153–160. 112

Gelfand, A. E. (1996). Model determination using sampling-based methods. In Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., editors, *Markov Chain Monte Carlo in Practice*, pages 145–162. Chapman & Hall. 112

Gelfand, A. E., Dey, D. K., and Chang, H. (1992a). Model determination using predictive distributions with implementation via sampling-based methods. In Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics 4*, pages 147–167. Oxford University Press. 17

Gelfand, A. E. and Smith, A. F. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85:398–409. 70

Gelfand, A. E., Smith, A. F., and Lee, T.-M. (1992b). Bayesian analysis of constrained parameter and truncated data problems using Gibbs sampling. *Journal of the American Statistical Association*, 87:523–532. 70

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian Analysis*, 1(3):515–534. 202

Gelman, A. (2008). The folk theorem of statistical computing. *Statistical Modeling, Causal Inference, and Social Science*. `https://statmodeling.stat.columbia.edu/2008/05/13/the_folk_theore/`. 98, 140

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. CRC Press, New York, 3rd edition. 33, 93

Gelman, A., Hwang, J., and Vehtari, A. (2014). Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, 24:997–1016. 146

Gelman, A. and Little, T. C. (1997). Poststratification into many categories using hierarchical logistic regression. *Survey Methodology*, 23:127–135. 196

Gelman, A. and Meng, X.-L. (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, 13:163–185. 59, 61, 63, 67, 77, 81, 138

Gelman, A., Meng, X.-L., and Stern, H. (1996). Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica*, pages 733–760. 144

Gelman, A. and Pardoe, I. (2006). Bayesian measures of explained variance and pooling in multilevel (hierarchical) models. *Technometrics*, 48:241–251. 214

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–472. 146, 151

Gelman, A. and Shalizi, C. R. (2013). Philosophy and the practice of Bayesian statistics. *British Journal of Mathematical and Statistical Psychology*, 66:8–38. 178

Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bürkner, P.-C., and Modrák, M. (2020). Bayesian workflow. *arXiv:2011.01808*. 5, 180, 214, 216

Gelman, A. and Yao, Y. (2021). Holes in Bayesian statistics. *Journal of Physics G: Nuclear and Particle Physics*. 5, 178

George, E. I. (2010). Dilution priors: Compensating for model space redundancy. In Berger, J. O., Cai, T. T., and Johnstone, I. M., editors, *Borrowing Strength: Theory Powering Applications–A Festschrift for Lawrence D. Brown*, pages 158–165. Institute of Mathematical Statistics. 118, 123

Gershman, S., Hoffman, M., and Blei, D. (2012). Nonparametric variational inference. In *Proceedings of the 29th International Conference on Machine Learning*. 152

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57:1317–1339. 18, 24, 37, 42, 97

Geweke, J. and Amisano, G. (2012). Prediction with misspecified models. *American Economic Review*, 102(3):482–486. 117

Geyer, C. J. (1992). Practical Markov chain Monte Carlo. *Statistical Science*, 7:473–483. 151, 153

Geyer, C. J. and Thompson, E. A. (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association*, 90:909–920. 78, 82, 98

Ghasemian, A., Hosseinmardi, H., Galstyan, A., Airoldi, E. M., and Clauset, A. (2020). Stacking models for nearly optimal link prediction in complex networks. *Proceedings of the National Academy of Sciences*, 117:23393–23400. 197

Giordano, R., Broderick, T., and Jordan, M. I. (2018). Covariances, robustness, and variational Bayes. *Journal of machine learning research*, 19. 22

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:359–378. 105, 144

Gobbo, G. and Leimkuhler, B. J. (2015). Extended Hamiltonian approach to continuous tempering. *Physical Review E*, 91:61301. 83

Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, 40:237–264. 151

Gorinova, M. I., Moore, D., and Hoffman, M. D. (2020). Automatic reparameterisation of probabilistic programs. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3648–3657. 94, 137, 168

Graham, M. M. and Storkey, A. J. (2017). Continuously tempered Hamiltonian Monte Carlo. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. 84, 88

Gumbel, E. J. (1958). *Statistics of Extremes*. Columbia University Press, New York. 13

Hansmann, U. H. (1997). Parallel tempering algorithm for conformational studies of biological molecules. *Chemical Physics Letters*, 281:140–150. 138

Heidemanns, M., Gelman, A., and Morris, G. E. (2020). An updated dynamic Bayesian forecasting model for the 2020 election. *Harvard Data Science Review*. 209, 210

Heiner, M., Kottas, A., and Munch, S. (2019). Structured priors for sparse probability vectors with application to model selection in Markov chains. *Statistics and Computing*, 29:1077–1093. 189

Hill, J. and Su, Y.-S. (2013). Assessing lack of common support in causal inference using Bayesian nonparametrics: Implications for evaluating the effect of breastfeeding on children's cognitive outcomes. *Annals of Applied Statistics*, pages 1386–1420. 39

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*. 242

Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. (1999). Bayesian model averaging: A tutorial. *Statistical Science*, pages 382–401. 109, 152, 222, 226

Hoffman, M. and Ma, Y.-A. (2020). Black-box variational inference as distilled langevin dynamics. In *Proceedings of the 37th International Conference on Machine Learning*. 138

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347. 20

Hoffman, M. D. and Gelman, A. (2014). The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15:1593–1623. 60

Huang, Z. and Gelman, A. (2005). Sampling for Bayesian computation with large datasets. *Technical Report, Columbia University*. `http://www.stat.columbia.edu/~gelman/research/unpublished/comp7.pdf`. 150

Imai, K., King, G., and Stuart, E. A. (2008). Misunderstandings between experimentalists and observationalists about causal inference. *Journal of the Royal Statistical Society: Series A*, 171:481–502. 39

Imbens, G. W. (2004). Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics*, 86:4–29. 39

Imbens, G. W. (2015). Matching methods in practice: Three examples. *Journal of Human Resources*, 50(2):373–419. 39

Imbens, G. W. and Rubin, D. B. (2015). *Causal Inference in Statistics, Social, and Biomedical Sciences*. Cambridge University Press. 45, 46

Ionides, E. L. (2008). Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311. 10, 11, 48

Jaakkola, T. S. and Jordan, M. I. (1998). Improving the mean field approximation via the use of mixture distributions. In *Learning in Graphical Models*, pages 163–173. Springer. 152

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3:79–87. 135, 199, 201

Jarzynski, C. (1997). Nonequilibrium equality for free energy differences. *Physical Review Letters*, 78:2690. 66

Johnson, L. T. and Geyer, C. J. (2012). Variable transformation to obtain geometric ergodicity in the random-walk Metropolis algorithm. *Annals of Statistics*, 40:3050–3076. 137

Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214. 199, 201

Jylänki, P., Vanhatalo, J., and Vehtari, A. (2011). Robust Gaussian process regression with a Student-t likelihood. *Journal of Machine Learning Research*, 12:3227–3257. 165

Kamary, K., Mengersen, K., Robert, C. P., and Rousseau, J. (2019). Testing hypotheses via a mixture estimation model. *arXiv:1412.2044*. 180, 199, 243

Kanamori, T. and Shimodaira, H. (2003). Active learning algorithm using the maximum weighted log-likelihood estimator. *Journal of Statistical Planning and Inference*, 116(1):149–162. 240

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220:671–680. 138

Kirkwood, J. G. (1935). Statistical mechanics of fluid mixtures. *Journal of Chemical Physics*, 3:300–313. 67

Kleijn, B. J. K. and Van der Vaart, A. W. (2012). The Bernstein-von-Mises theorem under misspecification. *Electronic Journal of Statistics*, 6:354–381. 143

Koopman, S. J., Shephard, N., and Creal, D. (2009). Testing the assumptions behind importance sampling. *Journal of Econometrics*, 149:2–11. 11, 18, 24

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2017). Automatic differentiation variational inference. *Journal of Machine Learning Research*, 18:430–474. 20, 172

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413. 177

Lavine, I., Lindon, M., and West, M. (2021). Adaptive variable selection for sequential prediction in multivariate dynamic models. *Bayesian Analysis*. 118

Le, T. and Clarke, B. (2017). A Bayes interpretation of stacking for $\mathcal{M}$-complete and $\mathcal{M}$-open settings. *Bayesian Analysis*, 12:807–829. 109, 114, 119, 139, 149, 178, 187, 197, 221

LeBlanc, M. and Tibshirani, R. (1996). Combining estimates in regression and classification. *Journal of the American Statistical Association*, 91:1641–1650. 109, 139, 197

Lee, B. K., Lessler, J., and Stuart, E. A. (2011). Weight trimming and propensity score weighting. *PloS one*, 6:e18174. 47

Lelievre, T., Rousset, M., and Stoltz, G. (2010). *Free Energy Computation: A Mathematical Perspective*. Imperial College Press, London. 63

Li, M. and Dunson, D. B. (2019). Comparing and weighting imperfect models using D-probabilities. *Journal of the American Statistical Association*, 115:1349–1360. 112

Li, Y. and Turner, R. E. (2016). Rényi divergence variational inference. In *Advances in Neural Information Processing Systems*, pages 1073–1081. 38

Liang, F., Paulo, R., Molina, G., Clyde, M. A., and Berger, J. O. (2008). Mixtures of g priors for Bayesian variable selection. *Journal of the American Statistical Association*, 103(481):410–423. 129

Linzer, D. A. (2013). Dynamic Bayesian forecasting of presidential elections in the states. *Journal of the American Statistical Association*, 108:124–134. 209

Liu, J. and Hodges, J. S. (2003). Posterior bimodality in the balanced one-way random-effects model. *Journal of the Royal Statistical Society: Series A*, 65:247–255. 143, 168, 170

Liu, Q. and Lee, J. D. (2016). Black-box importance sampling. *arXiv:1610.05247*. 48

Maciejowska, K., Nowotarski, J., and Weron, R. (2016). Probabilistic forecasting of electricity spot prices using factor quantile regression averaging. *International Journal of Forecasting*, 32:957–965. 242

MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press. 10, 17

Madigan, D., Raftery, A. E., Volinsky, C., and Hoeting, J. (1996). Bayesian model averaging. In *Proceedings of the AAAI Workshop on Integrating Multiple Learned Models*. 109, 152

Madras, N. and Zheng, Z. (2003). On the swapping algorithm. *Random Structures & Algorithms*, 22:66–97. 83

Mangoubi, O., Pillai, N. S., and Smith, A. (2018). Does Hamiltonian Monte Carlo mix faster than a random walk on multimodal densities? *arXiv:1808.03230*. 74, 137

Mangoubi, O. and Smith, A. (2017). Rapid mixing of Hamiltonian Monte Carlo on strongly log-concave distributions. *arXiv:1708.07114*. 141

Mangoubi, O. and Smith, A. (2019). Mixing of Hamiltonian Monte Carlo on strongly log-concave distributions 2: Numerical integrators. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, pages 586–595. 141

Marinari, E. and Parisi, G. (1992). Simulated tempering: A new Monte Carlo scheme. *Europhysics Letters*, 19:451. 82, 138

McAlinn, K., Aastveit, K. A., Nakajima, J., and West, M. (2020). Multivariate Bayesian predictive synthesis in macroeconomic forecasting. *Journal of the American Statistical Association*, 115:1092–1110. 118

McAlinn, K. and West, M. (2019). Dynamic Bayesian predictive synthesis in time series forecasting. *Journal of Econometrics*, 210:155–169. 118

Meng, X.-L. and van Dyk, D. A. (1999). Seeking efficient data augmentation schemes via conditional and marginal augmentation. *Biometrika*, 86:301–320. 188

Meng, X.-L. and Wong, W. H. (1996). Simulating ratios of normalizing constants via a simple identity: A theoretical exploration. *Statistica Sinica*, 6:831–860. 64

Mesquita, D., Blomstedt, P., and Kaski, S. (2020). Embarrassingly parallel MCMC using deep invertible transformations. In *Uncertainty in Artificial Intelligence*, pages 1244–1252. 150

Miller, A. C., Foti, N. J., and Adams, R. P. (2017). Variational boosting: Iteratively refining posterior approximations. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2420–2429. 152

Montgomery, J. M. and Nyhan, B. (2010). Bayesian model averaging: Theoretical developments and practical applications. *Political Analysis*, 18(2):245–270. 129

Morris, G. M., Goodsell, D. S., Halliday, R. S., Huey, R., Hart, W. E., Belew, R. K., and Olson, A. J. (1998). Automated docking using a lamarckian genetic algorithm and an empirical binding free energy function. *Journal of Computational Chemistry*, 19:1639–1662. 82

Mäntylä, M. V., Claes, M., and Farooq, U. (2018). Measuring LDA topic stability from clusters of replicated runs. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. 159

Neal, R. M. (1993). Probabilistic inference using Markov chain Monte Carlo methods. Technical report, Department of Computer Science, University of Toronto. Technical Report CRG-TR-93-1. 67, 82, 138

Neal, R. M. (1998). Regression and classification using Gaussian process priors. In Bernardo, J., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics*, volume 6, pages 475–501. Oxford University Press. 162, 165, 206

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, 11:125–139. 66, 83

Nemeth, C., Lindsten, F., Filippone, M., and Hensman, J. (2019). Pseudo-extended Markov chain Monte Carlo. In *Advances in Neural Information Processing Systems*, pages 4312–4322. 90

Oelrich, O., Ding, S., Magnusson, M., Vehtari, A., and Villani, M. (2020). When are Bayesian model probabilities overconfident? *arXiv:2003.04026*. 178

Ogata, Y. (1989). A Monte Carlo method for high dimensional integration. *Numerische Mathematik*, 55:137–157. 67

O'Hagan, A. (1987). Monte Carlo is fundamentally unsound. *Statistician*, pages 247–249. 69

O'Hagan, A. (1995). Fractional Bayes factors for model comparison. *Journal of the Royal Statistical Society: Series B*, 57:99–118. 113

Osband, I., Van Roy, B., Russo, D., and Wen, Z. (2019). Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20:1–62. 177

Owen, A. and Zhou, Y. (2000). Safe and effective importance sampling. *Journal of the American Statistical Association*, 95:135–143. 97

Owen, A. B. (2013). Monte Carlo theory, methods and examples. Online book at `http://statweb.stanford.edu/~owen/mc/`, accessed 2017-09-09. 14

Paananen, T., Piironen, J., Bürkner, P.-C., and Vehtari, A. (2021). Implicitly adaptive importance sampling. *Statistics and Computing*, 31:1–19. 97, 179

Papaspiliopoulos, O., Roberts, G. O., and Sköld, M. (2007). A general framework for the parametrization of hierarchical models. *Statistical Science*, 22:59–73. 137

Pearce, T., Zaki, M., Brintrup, A., and Neel, A. (2020). Uncertainty in neural networks: Approximately Bayesian ensembling. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*. 177

Peruggia, M. (1997). On the variability of case-deletion importance sampling weights in the Bayesian linear model. *Journal of the American Statistical Association*, 92(437):199–207. 17

Pflug, G. C. (1990). Non-asymptotic confidence bounds for stochastic approximation algorithms with constant step size. *Monatshefte für Mathematik*, 110(3):297–314. 21

Pickands, J. (1975). Statistical inference using extreme order statistics. *Annals of Statistics*, 3:119–131. 11, 12, 13

Piironen, J. and Vehtari, A. (2017a). Comparison of Bayesian predictive methods for model selection. *Statistics and Computing*, 27(3):711–735. 110, 199

Piironen, J. and Vehtari, A. (2017b). On the hyperprior choice for the global shrinkage parameter in the horseshoe prior. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. 92, 172

Piironen, J. and Vehtari, A. (2017c). Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, 11:5018–5051. 35, 92, 172, 173

Pirš, G. and Štrumbelj, E. (2019). Bayesian combination of probabilistic classifiers using multivariate normal mixtures. *Journal of Machine Learning Reserach*, 20:1–18. 197

Polson, N. G. and Scott, S. L. (2011). Data augmentation for support vector machines. *Bayesian Analysis*, 6:1–23. 188

Quiroz, M., Kohn, R., Villani, M., and Tran, M.-N. (2019). Speeding up MCMC by efficient data subsampling. *Journal of the American Statistical Association*, 114:831–843. 150

R Core Team (2020). R: A language and environment for statistical computing. `https://www.R-project.org/`. 100

Raftery, A. E. and Lewis, S. (1992a). How many iterations in the Gibbs sampler. In Bernardo, J., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics*, volume 4, pages 763–773. Oxford University Press. 151

Raftery, A. E. and Lewis, S. M. (1992b). Comment: One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo. *Statistical Science*, 7:493–497. 151

Ranganath, R., Gerrish, S., and Blei, D. (2014). Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822. 20

Ranganath, R., Tran, D., and Blei, D. (2016). Hierarchical variational models. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 324–333. 152

Reid, S. and Grudic, G. (2009). Regularized linear models in stacked generalization. In *International Workshop on Multiple Classifier Systems*, pages 112–121. 198

Rényi, A. (1961). On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1*. University of California. 23

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1278–1286. 26

Rischard, M., Jacob, P. E., and Pillai, N. (2018). Unbiased estimation of log normalizing constants with applications to Bayesian cross-validation. *arXiv:1810.01382*. 67

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407. 67

Robbins, H. E. (1968). Estimating the total probability of the unobserved outcomes of an experiment. *Annals of Mathematical Statistics*, 39:256–257. 151

Robert, C. and Casella, G. (2013). *Monte Carlo Statistical Methods*. Springer Science & Business Media, New York, 2nd edition. 64

Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Arroita, G., Hauenstein, S., Lahoz-Monfort, J. J., Schröder, B., and Thuiller, W. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, 40:913–929. 116

Roberts, G. O., Gelman, A., and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of Applied Probability*, 7:110–120. 141

Rosenbaum, P. R. and Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70:41–55. 48

Rosenbaum, P. R. and Rubin, D. B. (1984). Reducing bias in observational studies using subclassification on the propensity score. *Journal of the American statistical Association*, 79:516–524. 48

Rubin, D. B. (1981). The Bayesian bootstrap. *Annals of Statistics*, 9:130–134. 112

Rubin, D. B. and Thomas, N. (1992). Characterizing the effect of matching using linear propensity score methods with normal distributions. *Biometrika*, 79:797–809. 48

Rubin, D. B. and Thomas, N. (1996). Matching using estimated propensity scores: Relating theory to practice. *Biometrics*, pages 249–264. 48

Rubin, D. B. and Thomas, N. (2000). Combining propensity score matching with additional adjustments for prognostic covariates. *Journal of the American Statistical Association*, 95:573–585. 48

Sanz-Alonso, D. (2018). Importance sampling and necessary sample size: an information theory approach. *Journal on Uncertainty Quantification*, 6(2):867–879. 15, 44

Scarrot, C. and MacDonald, A. (2012). A review of extreme value threshold estimation and uncertainty quantification. *REVSTAT*, 10(1):33–60. 12

Schlitter, J. (1991). Methods for minimizing errors in linear thermodynamic integration. *Molecular Simulation*, 7:105–112. 67

Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and McCulloch, R. E. (2016). Bayes and big data: The consensus Monte Carlo algorithm. *International Journal of Management Science and Engineering Management*, 11:78–88. 151

Sejdinovic, D., Strathmann, H., Garcia, M. L., Andrieu, C., and Gretton, A. (2014). Kernel adaptive Metropolis-Hastings. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1665–1673. 90

Şen, M. U. and Erdogan, H. (2013). Linear classifier combination and selection using group sparse regularization and hinge loss. *Pattern Recognition Letters*, 34:265–274. 198

Shalit, U., Johansson, F. D., and Sontag, D. (2017). Estimating individual treatment effect: Generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3076–3085. 40, 46

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244. 195

Shirts, M. R. and Chodera, J. D. (2008). Statistically optimal analysis of samples from multiple equilibrium states. *Journal of Chemical Physics*, 129:124105. 64, 65

Sielken, R. L. (1973). Stopping times for stochastic approximation procedures. *Probability Theory and Related Fields*, 26(1):67–75. 21

Sill, J., Takács, G., Mackey, L., and Lin, D. (2009). Feature-weighted linear stacking. *arXiv:0911.0460*. 184, 197, 198

Sivula, T., Magnusson, M., and Vehtari, A. (2020). Uncertainty in Bayesian leave-one-out cross-validation based model comparison. *arXiv:2008.10296*. 115

Smyth, P. and Wolpert, D. (1998). Stacked density estimation. In *Advances in Neural Information Processing Systems*, pages 668–674. 121

Sriperumbudur, B. K., Fukumizu, K., Gretton, A., Schölkopf, B., and Lanckriet, G. R. (2012). On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599. 47

Stan Development Team (2020). Stan modeling language user's guide. Version 2.23. `https://mc-stan.org`. 59, 100, 136, 145, 180, 191, 217

Stephens, M. (2000). Dealing with multimodal posteriors and non-identifiability in mixture models. *Journal of the Royal Statistical Society: Series B*, 62:795–809. 143

Stroup, D. F. and Braun, H. I. (1982). On a new stopping rule for stochastic approximation. *Probability Theory and Related Fields*, 60(4):535–554. 21

Sugiyama, M., Krauledat, M., and Müller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:985–1005. 195

Sugiyama, M. and Müller, K.-R. (2005). Input-dependent estimation of generalization error under covariate shift. *Statistics and Decisions*, 23:249–280. 195

Sugiyama, M. and Ridgeway, G. (2006). Active learning in approximately linear regression based on conditional expectation of generalization error. *Journal of Machine Learning Research*, 7:141–166. 240

Sundin, I., Schulam, P., Siivola, E., Vehtari, A., Saria, S., and Kaski, S. (2019). Active learning for decision-making from imbalanced observational data. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6046–6055. 40, 46

Svensën, M. and Bishop, C. M. (2003). Bayesian hierarchical mixtures of experts. In *Uncertainty in Artificial Intelligence*. 199, 201, 202

Tian, K., Revelle, M., and Poshyvanyk, D. (2009). Using latent Dirichlet allocation for automatic categorization of software. In *6th IEEE International Working Conference on Mining Software Repositories*, pages 163–166. 159

Tipping, M. E. and Lawrence, N. D. (2005). Variational inference for Student-t models: Robust Bayesian interpolation and generalised component analysis. *Neurocomputing*, 69:123–141. 165

Tracey, B. D. and Wolpert, D. H. (2016). Reducing the error of Monte Carlo algorithms by learning control variates. In *Conference on Neural Information Processing Systems*. 197

Van Der Pas, S. L., Kleijn, B. J., and Van Der Vaart, A. W. (2014). The horseshoe estimator: Posterior concentration around nearly black vectors. *Electronic Journal of Statistics*, 8:2585–2618. 173

Vanhatalo, J., Jylänki, P., and Vehtari, A. (2009). Gaussian process regression with Student-*t* likelihood. In *Advances in Neural Information Processing Systems*, pages 1910–1918. 165

Vegetabile, B. G., Gillen, D. L., and Stern, H. S. (2020). Optimally balanced Gaussian process propensity scores for estimating treatment effects. *Journal of the Royal Statistical: Series A*, 183:355–377. 39

Vehtari, A., Gabry, J., Magnusson, M., Yao, Y., and Gelman, A. (2019a). *loo: Efficient leave-one-out cross-validation and WAIC for Bayesian models*. R package version 2.2.0. `https://mc-stan.org/loo`. 136, 149, 218

Vehtari, A., Gelman, A., and Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, 27:1413–1432. 12, 17, 115, 139, 146, 148, 149, 184, 187, 203, 214

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., and Bürkner, P.-C. (2020a). Rank-normalization, folding, and localization: An improved $\hat{R}$ for assessing convergence of MCMC. *Bayesian Analysis*. 14, 57, 92, 93, 137, 145, 146, 149

Vehtari, A., Gelman, A., Sivula, T., Jylänki, P., Tran, D., Sahai, S., Blomstedt, P., Cunningham, J. P., Schiminovich, D., and Robert, C. (2020b). Expectation propagation as a way of life: A framework for Bayesian inference on partitioned data. *Journal of Machine Learning Research*, 21:1–53. 145

Vehtari, A. and Lampinen, J. (2002). Bayesian model assessment and comparison using cross-validation predictive densities. *Neural Computation*, 14:2439–2468. 112

Vehtari, A. and Ojanen, J. (2012). A survey of Bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys*, 6:142–228. 105, 110, 187

Vehtari, A., Simpson, D., Gelman, A., Yao, Y., and Gabry, J. (2019b). Pareto smoothed importance sampling. *arXiv:1507.02646*. 5, 9, 14, 21, 24, 42, 44, 48, 51, 63, 116, 139, 148, 187, 195

Vehtari, A., Särkkä, S., and Lampinen, J. (2000). On MCMC sampling in Bayesian MLP neural networks. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, pages 317–322. 176

Wada, T. and Fujisaki, Y. (2015). A stopping rule for stochastic approximation. *Automatica*, 60:1–6. 21

Wang, F. and Landau, D. P. (2001). Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical Review Letters*, 86:2050–2053. 138

Waterhouse, S., MacKay, D., and Robinson, T. (1996). Bayesian methods for mixtures of experts. In *Advances in Neural Information Processing Systems*. 202

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, pages 681–688. 150

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5:241–259. 109, 139, 184, 197

Wong, H. and Clarke, B. (2004). Improvement over Bayes prediction in small samples in the presence of model uncertainty. *Canadian Journal of Statistics*, 32:269–283. 178

Woodard, D. B., Schmidler, S. C., and Huber, M. (2009). Conditions for rapid mixing of parallel and simulated tempering on multimodal distributions. *Annals of Applied Probability*, 19:617–640. 83

Yang, Y. and Dunson, D. B. (2014). Minimax optimal Bayesian aggregation. *arXiv:1403.1345*. 133, 198

Yang, Z. and Zhu, T. (2018). Bayesian selection of misspecified models is overconfident and may cause spurious posterior probabilities for phylogenetic trees. *Proceedings of the National Academy of Sciences*, 115:1854–1859. 112, 178

Yao, Y. (2019). Bayesian aggregation. *arXiv:1912.11218*. 6, 104, 178, 179

Yao, Y., Cademartori, C., Vehtari, A., and Gelman, A. (2020a). Adaptive path sampling in metastable posterior distributions. *arXiv:2009.00471*. 6, 55, 74, 138

Yao, Y., Mozumder, R., Bostick, B., Mailloux, B., Harvey, C. F., Gelman, A., and van Geen, A. (2021a). Making the most of imprecise measurements: Changing patterns of arsenic concentrations in shallow wells of Bangladesh from laboratory and field data. *arXiv:2101.06631*. 130

Yao, Y., Pirš, G., Vehtari, A., and Gelman, A. (2021b). Bayesian hierarchical stacking. *arXiv:2101.08954*. 6, 118, 183, 239

Yao, Y., Vehtari, A., and Gelman, A. (2020b). Stacking for non-mixing Bayesian computations: The curse and blessing of multimodal posteriors. *arXiv:2006.12335*. 6, 74, 100, 137, 197, 198, 207, 238

Yao, Y., Vehtari, A., Simpson, D., and Gelman, A. (2018a). Using stacking to average Bayesian predictive distributions (with discussion). *Bayesian Analysis*, 13:917–1007. 6, 104, 109, 111, 112, 139, 149, 153, 174, 178, 184, 195, 197, 199, 201, 216, 226

Yao, Y., Vehtari, A., Simpson, D., and Gelman, A. (2018b). Yes, but did it work?: Evaluating variational inference. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5577–5586. 5, 20, 28, 63, 172, 175, 202

Yuksel, S. E., Wilson, J. N., and Gader, P. D. (2012). Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1177–1193. 202

Zhang, J. and Stephens, M. A. (2009). A new and efficient estimation method for the generalized Pareto distribution. *Technometrics*, 51(3):316–325. 12, 13

Zhang, L. and Zhou, W.-D. (2011). Sparse ensembles using weighted combination methods based on linear programming. *Pattern Recognition*, 44:97–106. 198