Two Numerical Methods for Approximating High-Dimensional Posterior Distributions

Two Numerical Methods for Approximating High-Dimensional Posterior Distributions

A DISSERTATION PRESENTED
BY
JAMESON QUINN
TO
THE DEPARTMENT OF STATISTICS

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE SUBJECT OF
STATISTICS

HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS

MARCH 2020

Dissertation advisor: Luke Miratrix Jameson Quinn

Two Numerical Methods for Approximating High-Dimensional Posterior Distributions

Abstract

The three chapters within this dissertation are largely self-contained, though chapter 3 does build on the ideas and work of chapter 2. The underlying similarities and connections are discussed in the foreword, but the content may be summarized separately:

Chapter 1 Online data assimilation in time series models over a large spatial extent is an important problem in both geosciences and robotics. Such models are intrinsically high-dimensional, rendering naive particle filter algorithms ineffective. I present a novel particle-based algorithm for online approximation of the filtering problem on such models, using the fact that each locus affects only nearby loci at the next time step. The algorithm constructs hybrid particles at time t using an MCMC that combines values obtained by progressing various particles at time $t-1$, using custom-built proposal and acceptance probabilities. I show simulation results that suggest the error of this algorithm is uniform in both space and time, with a lower bias, though higher variance, as compared to a previously-proposed algorithm. Since this variance may be fixable with more computing power, this tradeoff is promising.

Chapter 2 Variational inference is a way to estimate posterior distributions, especially in cases such as models with many latent variables that make MCMC difficult. Existing techniques such as mean-field methods can fail to account for

posterior correlations, leading to downward bias in estimates of posterior variance. We present a novel technique, Laplace Family Variational Inference, for creating posterior estimates with more-realistic posterior correlation structures. We show that this technique outperforms Gaussian mean-field variational inference in two models: one simple two-variable model and one model based on a multi-site study. We give results of the latter model on real data for an educational intervention, Early College High Schools.

Chapter 3 Ecological inference – inferring individual-level quantities from group-level data – appears in many contexts, but is particularly key to demonstrating violations of the US Voting Rights Act. In this setting, the standard approach to solving the ecological inference problem is King's EI. We extend the EI framework in two ways. First, we give a flexible Bayesian model of voting behavior that can be easily customized for different scenarios. Second, we show how to use the techniques from the Chapter 2, along with some observation-dependent reparametrizations, to perform variational inference on our model. We demonstrate this on simulated data based on actual racial voting patterns in the 2016 Presidential election in North Carolina. We show that this technique is comparably accurate to existing methods. Our model, however, easily permits extensions which would allow for increased power and/or addressing open questions in ecological inference.

Contents

Author List

The following author contributed to Chapters 2 and 3: Mira Bernstein.

Listing of figures

This thesis is dedicated to Shasta.

Acknowledgments

Thank you above all to Luke, Mira, Rogelia, Ixchel, and Talya, without whom this work would not have been possible.

To Luke: it is, of course, an unusual position to have considered you a friend before you were an advisor and teacher. Yet you have continued to excel in all three roles, which is more than I deserve. I can't express how grateful I am. And from what I can see, you're a pretty damn good friend, advisor, and teacher to others, too.

To Mira: I would not have dreamed of finding such an ideal collaborator and coauthor. Your ideas and insight have enriched this work. Your ability to entertain my flights of intuitive fancy, while nevertheless keeping us grounded in mathematical rigor, has been invigorating. And through it all, both your generosity of spirit and your staunch intellect have made working with you a delight. I'd love to have the many chances it will take to repay you.

A Rogelia: por supuesto, te debo gracias sin cesar por tu paciencia y apoyo. Pero aunque el agradecimiento llene mi corazon, lo que más le brota es el amor siempre joven por tu presencia, por tus chistines, y por la seguridad que tengo en ti.

To Ixchel: I could say that you are of course the pride of my life, but that's not the point here. Or, I could thank you for your patience with me as the deadline approaches and I've gotten "loopy", but that's boring. When I think of how I

hope you are reflected in these pages, I think of how, while your love and respect is something that inspires me to be my best, your clear-eyed humor also reminds me not to take myself too seriously. So, thank you.

To Talya: in my mind, it's late at night, and you're patiently listening to yet another iteration of "my thesis is broken but I think I can fix it". I feel your warmth and support. Thank you.

To Pierre: While any mistakes remain my own, every part of this thesis has been strengthened by knowing how you will always respectfully insist that all the details make sense. Truly, I'm grateful.

To Gary: thanks for being willing to play along with my initial inflated claims about what my method could do. I know that the current work does not yet fully reach that promise, but I still hope it will, and I'd love to be able to keep working with you as I bring it there.

To my parents: In every possible way, this wouldn't exist without you. Thank you.

To the grad students and faculty of the Harvard statistics department: I truly believe that there is no better place I could have been learning for these years. Thank you for too many reasons to name.

Conditioning is the soul of statistics.

Joe Blitzstein

0 Foreword

Statistics is the discipline that deals with the use of quantitative evidence. Thus, as Professor Blitzstein reminds us, the key step is to condition on that evidence, using Bayes' theorem. For observed evidence E and a hypothesis H from the space of possible hypotheses $\mathcal{H}$:

$$P(H|E) = \frac{P(E|H)P(H)}{\int_{H' \in \mathcal{H}} P(E|H')P(H')dH'}$$

Usually, the hardest part about finding the posterior distribution on the left is dealing with the denominator on the right, the normalizing constant. This becomes especially tough when $\mathcal{H}$, the space of possible hypotheses, is high-dimensional, so that attempting to numerically approximate an integral over all possible hypotheses is essentially hopeless — at least, without additional constraints or assumptions.

This thesis concerns itself with two novel methods for approximating two different types of high-dimensional posterior distributions. In chapter 1, the posterior in question is a filtering distribution; the best guess of the state of a system with known dynamics, conditional on an ongoing time series of data that must be assimilated over time. Chapters 2 and 3 deal with a more-traditional statistical problem of posterior distributions of model parameters. Specifically, chapter 2 gives methods for a general class of latent variable models, and chapter 3 applies these methods to the specific problem of ecological inference.

Creating numerical methods like these is a finicky job. It involves building an algorithm with several steps when, in one or more of those steps, exact solutions are difficult or impossible to come by. Thus, one must find good approximations, while keeping a good intuitive grasp of how they interact holistically. Meanwhile, one must keep juggling the different roles necessary — mathematician, programmer, data wrangler, writer, and project manager.

Of course, I did not invent either of the basic methods expounded here out of whole cloth; they are based on prior work. But in both cases, they combine several original ideas and/or original applications of existing ideas into a coherent and working whole.

I would not have been able to accomplish all this without the help and support of the people mentioned in the acknowledgements above. In particular, Mira Bernstein has been a close collaborator and colleague in my work on the second and third chapters. Still, in each chapter, the key idea is my own: in chapter 2, using Laplace families for variational inference, and in chapter 3, applying these techniques to ecological inference.

0.1 Commonalities between the two methods discussed

The basic problem of estimating high-dimensional posterior distributions is a difficult one. In low-dimensional cases, it may be possible to use simple approximate numerical integration to estimate the normalizing constant. But in high-dimensional cases, the variance of such estimates becomes unmanageable.

Thus, one is forced to find and take advantage of additional regularities, constraints, or reasonable assumptions on the problem domain to make it tractable.

Though the two methods introduced in this thesis are different in nearly all of their particulars, they have several aspects in common. In both cases, I address a relatively broad class of problems with practical importance, problems which existing methods struggle to resolve. Then, in both cases, I take advantage of two constraints or assumptions about the problem space; in each case, this includes one that relates to the dependency structure of the data, and one that relates to the dynamics and/or distributions involved for individual data points. Through these constraints or assumptions, I am able to reduce problems that are nearly impossible to ones that are merely difficult.

The two basic methods themselves also have some overall features in common. In both cases, there is one initial core idea, but in order to get it to work, we have had to adapt other subsidiary ideas in order to create a full, working algorithm. And in both cases, the resulting final algorithm I lay out here works and shows broad promise, but I see possibilities for further adaptations and/or refinements in order to apply it in a more practical sense.

0.2 The problem domains are important and general

For chapter 1, the domain is data assimilation. That is, the goal is an online algorithm to incorporate information from a time series of observations of an evolving system, and use it to build a coherent understanding of its possible current state. This has broad and important applications in applied settings: for instance, in modeling geophysical processes such as weather, or in allowing robots to maintain a model of their surroundings. The problems that current techniques have with high-dimensional problems of this type are well-known, and I hope that this thesis shows that my general approach is very promising.

For chapters 2 and 3, the domain is latent variable models for independent units. Such models are common in many scientific settings, both experimental

and observational. It remains to be seen whether MCMC, SMC, or variational inference will ultimately prove to be the best workhorse for these problems, but if it is the latter, I believe that something like the approach given here will be part of the way forward. And even if, ultimately, we find ways to resolve the difficulties with MCMC or SMC in these high-dimensional cases, some of the steps I've used here (such as analytic amortization) may still prove useful.

Specifically in chapter 3, I've focused on a problem relating to voting. While this is far less general a case than chapters 1 and 2, it is still of paramount interest. For me, personally, voting-related issues were what inspired me to pursue a doctorate in the first place, and I intend to continue exploring this area.

0.3 How to make each of the problems tractable: constraints and assumptions

In chapter 1, the key constraint that I've used to make the problem tractable is spatial structure. For instance, in the case of a weather model, a cloud at one location at time t may affect nearby locations at time $t + \epsilon$, but can not affect far-off locations until more time has passed. In practice, this means that the estimated posterior should respect the correlation structure between nearby locations, but can ignore that between far-off ones. This is what allows my "Finkelstein" approach, of cutting locations apart and then putting them back together in a principled way, to work.

In chapters 2 and 3, the key structure I take advantage of is the conditional independence of each unit's observations, given global parameters and observable unit characteristics (covariates).

0.4 Secondary ideas and contributions, and directions for future work

In chapter 1, the central idea is to modify Rebeschini and van Handel's block particle filter algorithm by using the forward probability in order to ensure that

the resulting hybrid particles are more plausible; that they do not have problematic "seams" between loci with different implied histories (smoothing distributions). In order to make this basic idea work, I have created a Metropolis-Hastings-like MCMC algorithm, with proposal and acceptance probabilities that have been built from principled mathematical arguments and at least somewhat tuned for practical performance. Bringing together this MCMC over multiple loci, with a Horvitz-Thompson estimator over multiple histories to approximate the acceptance probability, is, I believe, an interesting combination, and one that might be applicable elsewhere.

Meanwhile, there is further work to be done to ensure this idea reaches its full promise. I've begun to explore how to modify it for cases when the system dynamics are deterministic or nearly so, such that the version presented here fails for lack of ergodicity. I also have ideas about using concepts from unscented Kalman filters to improve the proposal distribution, and thus the mixing rate, of the Finkelstein MCMC.

In chapters 2 and 3, the central idea is to use observed information to create a variational guide family that respects posterior correlations without needing to fit an excessive number of parameters. The ideas about how to combine this with analytic amortization and subsampling, including taking advantage of the "free" step of Newton's method in the amortization, came up along the way. Still, it may be possible to separate these "secondary" ideas from Laplace family variational inference, and to use them in the context of SMC or MCMC. There are also further improvements in the amortization functions, as well as simple computational optimizations, that I have not had time to include in this work, but that I hope to add later.

In particular, for chapter 3, the point of creating an extensible model is to try out extensions, while the current work has only gone so far as validating the basic methodology. It is clear that this method has some ability to infer real patterns in the data. My interest in voting issues was what inspired me to approach a PhD in the first place, and so I will continue to work on these issues.

1

A High-Dimensional Particle Filter Algorithm

1.1 Background

Filtering problems arise in many applied contexts, whenever noisy observations over time must be combined, using an explicit dynamical model, into a best-guess distribution of a current state. In cases where the system being modeled involves processes over a large spatial extent, such as models of weather or other large-scale fluid dynamics, filtering is also called data assimilation.[5] This is an active area of research, with broad applications in predictive geoscience[6][55] and robotics[52]. In fact, it is considered to be among the

central problems in both of these disciplines.[1]

The basic filtering problem is as follows. We model the state of the system at time t as a random variable X_t. In our context, X_t will have values $x_{\substack{l \\ t}}$ at each spacial locus l, for a large number of loci. We assume $X_0, ..., X_T$ form a Markov chain with known and sampleable densities for both the initial state (π_0) and transition function (P, which maps states or densities at time $t-1$ to densities at time t). We also have a series of observations $Y_1, ..., Y_T$, and we assume that each Y_t depends only on the corresponding X_t according to a known and sampleable observation density $f(Y_t|X_t)$. This is shown graphically in Figure 1.1.1.

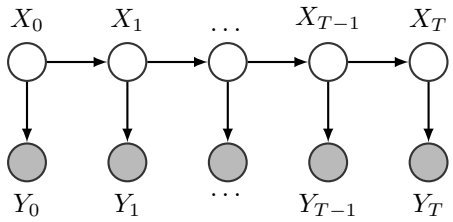


Figure 1.1.1: Graphical model of a low-dimensional filtering problem, with time going horizontally from left to right.

At each time step t, we wish to estimate the filtering distribution: that is, the probability density π_t of $X_t|\{Y_1, ..., Y_t\}$. Because the X_t are Markovian, and Y_t depends only X_t, we can write π_t recursively as

$$\pi_t(\cdot) = \boldsymbol{E}_{X_{t-1}\sim\pi_{t-1}}[P(X_t \in \cdot|Y_t, X_{t-1})] \tag{1.1}$$

To minimize subscripts, we adopt the following notation:

- We abbreviate the filtering distributions π_{t-1} and π_t by τ and π respectively.

[1]According to the papers cited above, there is "much focus in the [geoscience] literature on the assimilation of data and numerical models pertain[ing] to the sampling of high-dimensional probability density functions" [6], and "The SLAM [simultaneous location and mapping] problem is generally regarded as one of the most important problems in the pursuit of building truly autonomous mobile robots."[52]

- We abbreviate samples from X_{t-1} and X_t by $\boldsymbol{x}$ and $\boldsymbol{z}$ respectively.
- When $\boldsymbol{y}_{t-1}$ is not relevant, we write $\boldsymbol{y}$ for $\boldsymbol{y}_t$.
- Superscripts should not be read as exponentiation for these and similar entities.

The recursive formula for π_t suggests the possibility of online calculation, with only a constant computing time required to update from $\tau = \pi_{t-1}$ to $\pi = \pi_t$. However, unless we assume a particular parametric form for π, there is no finite set of sufficient statistics that could stand in for the full distribution. Thus, aside from very simple special cases, exact calculation is impossible; we look for an approximation instead.

A widely-used recursive algorithm for approximating the filtering distribution is the bootstrap particle filter. Assuming we have a sampleable distribution $\hat{\tau}$ at time $t-1$ that approximates the true filtering distribution τ, we proceed as follows:

1. Sample M iid particles $\boldsymbol{x}^{1..M}$ from $\hat{\tau}$. (Note that if we have been following the algorithm up to step $t-1$, then $\hat{\tau}$ takes the form of step (3) below.)
2. For each $\boldsymbol{x}^i$, progress it to get candidate particle $\boldsymbol{z}^i \sim \mathsf{P}\boldsymbol{x}^i$.
3. Find weights $w^i \equiv f(\boldsymbol{y}|\boldsymbol{z}^i)$. The set of weighted particles forms

$$\hat{\pi} \equiv \frac{\sum_{i=1}^{M} w^i \delta(\boldsymbol{z}^i)}{\sum_{i=1}^{M} w^i}.$$

   Here $\delta(a)$ is the Dirac delta density; for example, $\frac{1}{2}(\delta(0) + \delta(1))$ is the Bernoulli distribution with $p = \frac{1}{2}$.

A key property of the particle filter algorithm is that, for large enough M, the Monte Carlo error remains under control, even as the time steps accumulate.[11] Specifically, let $\hat{\pi}_t^M$ be the approximation to π_t obtained using M particles, as above. Let $\mathcal{F}$ be the set of functions from the domain of π_t to $(-1, 1)$. For each

$\mathrm{f} \in \mathcal{F}$, denote $\boldsymbol{E}_{X\sim\pi_t}(\mathrm{f}(X))$ and $\boldsymbol{E}_{X\sim\hat{\pi}_t^M}(\mathrm{f}(X))$ by $\pi_t(\mathrm{f})$ and $\hat{\pi}_t^M(\mathrm{f})$ respectively. Then

$$\sup_{\mathcal{F}} \boldsymbol{E}|\pi_t(\mathrm{f}) - \hat{\pi}_t^M(\mathrm{f})| \leq \frac{C}{\sqrt{M}}, \tag{1.2}$$

for some constant C that *does not depend on* t. The outer expectation here is taken over the randomness of the algorithm itself; that is, considering the distribution $\hat{\pi}$ as itself a random variable, while π is fixed.[44, p. 2814]

Now suppose that we are interested in modeling processes with a large spatial extent. For instance, in a weather model, one might use a lattice of points that cover the region of interest, with various continuous values (temperature, humidity, pressure, wind, etc.) recorded at each locus. If X_t contains information about L separate spatial loci, and the state space at each locus has dimension K, then the full state space of X_t has dimension KL. In practical applications, this can easily be 10^7 or more.[54]

To see why, consider a schematic diagram of the model (Figure 1.1.2), where the state of the system at time t and locus l is denoted $x_{\substack{l\\t}}$.

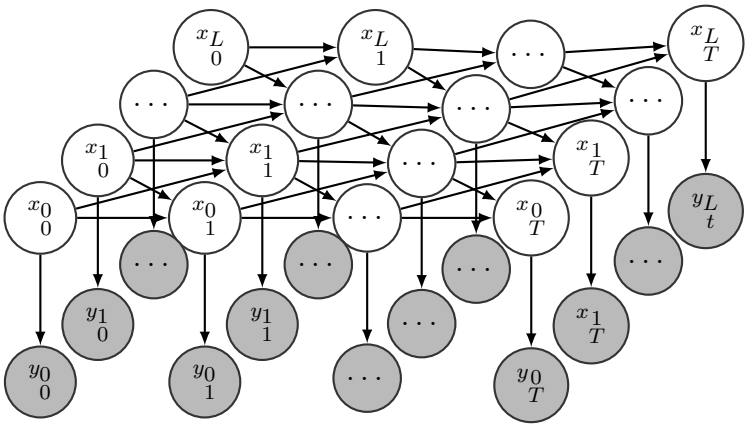


Figure 1.1.2: Graphical model of a (simple) high-dimensional filtering problem. Time is still represented left-to-right, and a single spatial dimension is represented by diagonal sets of simultaneous variables.

Note the following assumptions implicit in the diagram:

- $(y_{\underset{t}{l}}|x_{\underset{t}{l}}) \perp\!\!\!\perp x_{k\neq l}$. This assumption will be used in the following for simplicity, although I believe it can be relaxed in practice with only minor additional complications.

- More crucially, $x_{\underset{t}{l}}$ depends on $x_{\underset{t-1}{k}}$ only for k in some small spatial "neighborhood" $\mathcal{N}(l)$ of l. The precise composition of $\mathcal{N}(l)$ depends on model assumptions as well as the way that the L loci are positioned in space. The diagram depicts the case where the loci are all laid out along a single spatial dimension, so that the immediate neighbors of locus l are loci $l-1$ and $l+1$. In practical applications, the loci would more likely be connected in a 2D or 3D grid.

I will discuss this locality of dynamics assumption further below, as it is key to the performance of the algorithm I propose in this paper.

Technically speaking, the error bounds in 1.2 still apply: errors are stable over time and inversely proportional to the square root of the number of particles ($\propto 1/\sqrt{M}$). But it is widely recognized that the bootstrap particle filter is no longer a practical solution in this context, due to weight degeneracy.[5 1][7][3] The problem is that, in the resampling step, the majority of the weight will tend to be carried by only a small fraction of the proposed particles. To see why, note that the log likelihood of each particle is the sum of its log likelihood at each locus. Since these terms are roughly independent, as L increases, the empirical distribution of log likelihoods of the particles comes to resemble a Gaussian distribution with variance that scales linearly with L. Thus the weights come to be distributed approximately according to a log-normal distribution, whose skewness increases exponentially with L. Thus, the fraction of particles with above-average weight will shrink exponentially with L. [2]

In fact, without an exponentially large number of particles, not only will one of them tend to have more weight than all the others, but it is likely that there is some missing value whose weight would dominate even our best particle. At this

[2]Bickel *et al.*[7] formalize this line of argument, showing that the variance of the log likelihood may be seen as an estimate of the effective state dimension depicted by the measurements $\boldsymbol{y}$.

point the particle filter ceases to be a useful approximation of the filtering distribution. That is, although the constant C in Equation 1.2 does not depend on time, it does grow exponentially with L. This is what is known as *curse of dimensionality* for particle filters.

1.1.1 Existing state of the art (Rebeschini and van Handel, 2015)

There have been various proposals for dealing with this curse of dimensionality in general. In fact, there are three different recent survey articles reviewing and comparing these, by Septier and Peters[49], Morzfield *et al.*[36], and Farchi and Bocquet.[15] Some of these prior methods do not use the locality of dynamics assumption, which I believe limits their effectiveness. This includes Gilks and Berzuini,[17] who suggest rejuvenating particles with MCMC steps, targeted to the filtering distribution, to avoid the duplication problem from resampling; and Goodsill and Clapp,[18] who propose using bridging densities such as annealed quasi-filtering densities to solve the problem of lack of overlap between the progressed density $\mathsf{P}\tau \equiv (\boldsymbol{z}|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1})$ and the likelihood $f(\boldsymbol{y}_t|\boldsymbol{z})$.

The two previous proposals that do use locality of dynamics come from Poterjoy[40][41] and from Rebeschini and van Handel.[44] Of these two, Rebeschini and van Handel's proposal is more generally applicable, so I will explain it further below. Poterjoy, on the other hand, suggests a scheme that, as given, is limited to situations of sparse observations; it uses estimated covariance matrices to blend resampled particle filter values at a local scale with un-resampled values at a meso-scale.

Rebeschini and van Handel's proposal is called the block particle filter; also sometimes termed the localized particle filter. In simple terms, they replace the global resampling step of the bootstrap particle filter with a local resampling step which constructs new particles by resampling neighborhoods independently.

They begin their theoretical discussion by offering an overall point of view of the problem which very much inspired the current work. They focus on the decay of correlations between local values as spatial distance increases, which

they say is "in essence a spatial counterpart of the much better-understood [temporal] stability property of nonlinear filters". This decay of correlations, discussed further below, is a product of the locality-of-dynamics assumption shown in the diagrams above.

Rebeschini and van Handel partition the loci $1, \ldots, L$ of the progressed particles into J zones $\{\mathcal{Z}_j\}$, where each zone consists of a small number of (contiguous) loci. They then weight and resample values from each zone independently. This produces what I would call "Frankenstein" particles, sewn together from zone-sized pieces of different particles. Because of the reweighting and resampling, each piece tends to fit well with observations locally, but at the "seams" between zones, values at neighboring loci often come from progressing different particles from time $t-1$.

The precise steps of the block particle filter algorithm are as follows:

1. Given $\hat{\tau}$, a sampleable distribution that approximates the ideal filtering distribution τ, sample M iid particles $\boldsymbol{x}^{1..M}$ from $\hat{\tau}$. (Note that if $\hat{\tau}$ takes the form of the output of step (3) below, then this amounts to sampling, independently with replacement, a k_j^i for each zone $Z_j; 1 \leq j \leq J$ and particle i with probability $w_{Z_j}^{k_j^i} / \Sigma_n w_{Z_j}^n$; then putting those together to make the final particles, so that $l \in Z_j \Rightarrow z_l^i = z_l^{k_j^i}$.)

2. For each $\boldsymbol{x}^i$, progress it to get $\boldsymbol{z}^i \sim \mathsf{P}\boldsymbol{x}$.

3. Find weights for each particle for each zone, based on the local observation likelihood: $w_{Z_j}^i = \prod_{l \in Z_j} f(y_l | z_l^i)$. Then

$$\hat{\pi} \equiv \bigotimes_{j=1}^{J} \frac{\sum_{i=1}^{M} w_{Z_j}^i \delta(\boldsymbol{x}_{Z_j}^i)}{\sum_{i=1}^{M} w_{Z_j}^i}$$

Intuitively, cutting the progressed particles into zones and reconstituting them is a way to solve the exponential curse of dimensions. However, it gives rise to a different problem: it breaks any inter-locus dependencies across zone

boundaries, whether those come from pre-existing dependencies at the $t-1$ time step or are induced by cross-border dynamics during the latest time step (because the transition kernel uses a history that overlaps the boundary). These broken dependencies lead to error at the boundaries, which does not disappear even as number of particles goes to infinity.

In particular, this could lead to unrealistic dynamics near the boundaries at later time steps, especially if the forward density operator P is nonlinear. [44, p. 2829] For instance, imagine a weather model in which the hypothetical air pressure in a particle varied reasonably within each zone, but a discontinuity at the zone boundary led to a prediction of a tornado forming in the next time step. Note that the algorithm proposed in this paper avoids such discontinuities, but for unrelated reasons can be ill-suited to modeling models with nonlinear dynamics; I will address this issue in later work with a proposed extension to my algorithm.

For the block particle filter method to be useful, correlations between local values must tend to decay with distance. If such decay of correlations holds, then, far enough from a zone boundary, the dynamics return to normal.

This intuition helps explain the error bounds that these researchers prove their algorithm obeys. They show that the error for the value at a given locus l satisfies

$$|||\pi - \hat{\pi}^M|||_{\{l\}} \leq \alpha(\frac{e^{\beta|Z_l|}}{\sqrt{M}} + e^{-\gamma \inf|l-b\in Z_l^C|}) \tag{1.3}$$

where the constants $\alpha, \beta, \gamma > 0$ do not depend on t. (They define a norm for this purpose which measures the distance between random distributions; I will not reproduce this definition; here, this result is merely stated as a guide for intuition.)

One can see that there is a tradeoff: using smaller zones and/or more particles allows better guesses for a given zone to control the term $\frac{e^{\beta|Z_l|}}{\sqrt{M}}$, while using larger zones means that there are loci farther from boundary effects and thus controls the term $e^{-\gamma \inf|m-b\in Z_l^C|}$. In practice, Rebeschini and van Handel give a simple example where it would still be possible to control average error per locus, based

on finding an optimal balance between adding particles and increasing neighborhood size, but in that example the inverse of their error bound grows only logarithmically with the computing resources/number of particles — in other words, as tolerances tighten, the required number of particles can still grow exponentially. Thus, although their algorithm in practice gives error far lower than that of the bootstrap particle filter, it has not fully overcome the problem of needing exponential computing cost, especially if one wishes to achieve fixed error bounds that are lower than what comes easily with a moderate neighborhood size.

1.2 The Finkelstein solution

In this section, I will sketch out the basic outlines of a recursive algorithm in which each particle at time t is composed of values at different loci which are drawn from state vectors progressed from different particles at time $t-1$. The choice of which values for a given locus combine with which values at other loci is made by running a separate Metropolis-Hastings MCMC to create each composite particle, proposing to replace one locus value at each MCMC step. What acceptance ratio ρ to use for those proposals will be discussed in later sections.

I name this the Finkelstein algorithm, after the character Sally Finkelstein from the movie "The Nightmare Before Christmas". I have already compared the block particle filter to a Frankenstein solution, in which the progressed particles are chopped up and then randomly sewn back together. In that algorithm, the suitability of the values in each zone of each particle is measured against observation, but not against the other zones on which it borders. Finkelstein can improve on this. Though herself originally a Frankenstein's-monster-like creation of a stereotypical mad doctor, now that she has been animated, Sally is able to lose her body parts and sew them back on, and thus presumably to choose for herself only those body parts that best fit together. In my terms, Finkelstein would be able to run an MCMC process on her own body, targeting whatever

distribution she pleases.

A key aspect of this algorithm is the Metropolis-Hastings acceptance ratio ρ. When choosing a formula for such a ratio, the key question is, what distribution do we wish to target? I'll begin by showing an algorithm that targets the natural unnormalized density:

$$\begin{aligned} f_{\boldsymbol{x}\sim\tau}(\boldsymbol{z}|\boldsymbol{y}) &= \int f(\boldsymbol{z}|\boldsymbol{y},\boldsymbol{x})\tau(\boldsymbol{x})d\boldsymbol{x} \\ &\propto \int f(\boldsymbol{y},\boldsymbol{z})f(\boldsymbol{z},\boldsymbol{x})\tau(\boldsymbol{x})d\boldsymbol{x} \\ &= [\Pi_l f(y_l|z_l)] \int [\Pi_l f(z_l|\boldsymbol{x})]\tau(\boldsymbol{x})d\boldsymbol{x} \end{aligned} \tag{1.4}$$

It will turn out that targeting this density still suffers a similar curse of dimensionality as the bootstrap particle filter, so I will modify the algorithm, such that its stationary distribution is not precisely the above expression. Still, this expression is still the starting point; by approximately targeting it, I approximately target π.

Here are the steps of the basic Finkelstein algorithm. I do not include a formula for the acceptance probability ρ here; I will develop and discuss several alternatives for such a formula, based on modifications of the density above, in the following sections.

1. Assume we have $\hat{\tau}^M$, a sampleable distribution which in some sense approximates the ideal filtering distribution τ, and which must be of the form $\frac{1}{M}\sum_{i=1}^{M}\delta(\boldsymbol{x}^i)$. Note that unlike the cases of the bootstrap and block particle filters, the final $\hat{\pi}$ produced by this algorithm already has equally-weighted particles; so if $\hat{\tau}^M$ comes from $\hat{\pi}$ of a previous iteration of this algorithm, resampling is not required.

2. For each particle $\boldsymbol{x}^i$, progress it to get a full particle $\tilde{\boldsymbol{z}}^i \sim \mathsf{P}\boldsymbol{x}^i$ whose local values are known as $\tilde{z}^i_l$. (In later steps, I will assume for simplicity that

there are no duplicate values at any locus, so $i \neq j \Rightarrow \tilde{z}_l^i \neq \tilde{z}_l^j$, but relaxing this assumption should be straightforward if necessary.)

3. Find likelihood weights for each such local value, denoted $w_l^i \equiv f(y_l|\tilde{z}_l^i)$; and forward densities conditional on $\boldsymbol{x}^j$ for all j (including $j = i$), denoted $f_l^{j\to i} \equiv f_{\mathsf{P}}(\tilde{z}_l^i|\boldsymbol{x}^j)$.

4. In parallel, for $k = 1, \ldots, M$, do the following:

   (a) Create a new proposal particle by independently sampling each locus of a vector $\boldsymbol{\iota}^0 \in \{1..M\}^L$. This vector specifies the source for the value of $\tilde{z}$ that is being considered at each locus; that is, after running the MCMC for S steps, the final value $\boldsymbol{\iota}^S$ will be used to define $\boldsymbol{z}^k$ by setting $z_l^k = \tilde{z}_l^{\iota_l^S}$. The initial sampling at each locus uses the probabilities

   $$P(\iota_l^0 = i) = w_l^i / \sum_j w_l^j.$$

   (Note that these initial sampling probabilities are arbitrary and, if the MCMC successfully runs until convergence, irrelevant. The probabilities above represent a reasonable starting point that should converge reasonably well, but it may be possible to get even faster convergence through some sampling scheme that is not independent across loci.)

   (b) Run a Metropolis-Hastings MCMC chain targeting an approximation of the filtering distribution, for $s = 1, ..., S$ steps to (assumed) convergence:

      i. Choose a spatial locus $\lambda(s) \in \{1, \ldots, L\}$ uniformly at random. For brevity, I will refer to this as λ, suppressing the dependency on s, in the steps that follow.

      ii. Sample a proposed particle ι^* from which to draw the

replacement value $\tilde{z}_\lambda^{\iota^*}$ for locus λ, with probability

$$P(\iota^* = i) = v_\lambda^i = w_\lambda^i / \sum_j w_\lambda^j.$$

As with λ itself, I am omitting here the subscript s, even though this will be resampled at each step. Note that unlike $\boldsymbol{\iota}^s$, which is a vector of one integer per locus, this ι^* is only one integer, which determines the source for the proposed value only at locus λ. So for convenience in the formulas below, I will also define the vector $\boldsymbol{\iota}^{**}$ such that $\iota_\lambda^{**} = \iota^*$ and $\forall k \in \{1, ..., \lambda - 1, \lambda + 1, ..., L\} : \iota_k^{**} = \iota_k^{s-1}$

iii. Accept this proposed change with M-H probability: $1 \wedge \rho(\boldsymbol{\iota}^{s-1}, \lambda, \iota^*)$, where ρ is defined below. In case of acceptance, $\boldsymbol{\iota}^s := \boldsymbol{\iota}^{**}$; Otherwise, in case of rejection, make no change, so that $\boldsymbol{\iota}^s := \boldsymbol{\iota}^{s-1}$.

(c) Let $z_l^k = \tilde{z}_l^{\iota_l^S}$; that is, leave the particle unchanged at all locations $l \neq \lambda$.

5. The final set of particles forms

$$\hat{\pi} \equiv \frac{1}{M} \sum_{i=1}^{M} \delta[(z_1^i, ..., z_L^i)].$$

We would like to tune the proposal and acceptance probabilities so that this algorithm targets the distribution $(\boldsymbol{z}|\boldsymbol{y}, \{\boldsymbol{x}^i\})$. Insofar as we succeed, the full algorithm will be very similar to a bootstrap particle filter, which similarly targets that distribution. Thus, on a purely intuitive level, it is unsurprising that this should work if the MCMC does. But the whole process depends on the validity of the MCMC, which in turn depends on the M-H acceptance probability $\rho(\boldsymbol{\iota}^{s-1}, \lambda, \iota^*)$, left unspecified above.

1.3 Developing a working formula for ρ

In this section, I'll first develop some motivating understanding of what ρ should be like, then develop two specific formulas for ρ:

1. The first formula, ρ_{full}, is for illustrative purposes only. Though it is, by construction, asymptotically correct — that is, the algorithm using ρ_{full} approaches the correct filtering distribution as the number of particles M approaches infinity — it is unsuitable for use in practice. Not only does it lead to impractically high computational costs for a specific M, it also suffers from similar dimensionality problems as the bootstrap particle filter.

2. The second formula ρ_{local} only considers values of the MCMC in some local neighborhood of $\lambda(s)$. It thus does not suffer the weight degeneracy problem of the bootstrap particle filter or ρ_{full}, while still being possible to calculate in computing time that's polynomial in number of particles, and linear in dimension and number of time steps.

In the next section, I'll develop a further formula for ρ that improves the computational characteristics, as well as proposing some other computational optimizations.

1.3.1 ρ_{full}: full acceptance probability

I will begin by using a standard Metropolis-Hastings ratio to derive a ρ_{full} that targets the (unnormalized) density 1.4. That expression is promising in one sense: it suggests that one can judge the fit of local proposals drawn from two different particles using an expression involving the transition kernel forward density. However, the fact that it involves a sum over all particles, of a product over all loci, makes ρ_{full} computationally unworkable in practice. And even if there were enough available computational power to calculate this at every step

of an MCMC, taking a product over all loci would lead to similar problems with high dimensions as those of the naive high-dimensional bootstrap particle filter.[3]

If we propose replacement z_λ^i for the value at a given locus λ with probability proportional to some value v_λ^i, we can construct a Metropolis-Hastings acceptance probability in the usual way, by multiplying a ratio of target densities (proposed over current) by a ratio of proposal densities (current over proposed).

(To avoid nested subscripts/superscripts in the following, I leave out redundant indices for locus; thus using the notation $f_k^{j\to\iota^{**}}$ rather than $f_k^{j\to\iota_k^{**}}$, using $f_k^{j\to\iota(s-1)}$ rather than $f_k^{j\to\iota_k^{s-1}}$, and using $w_k^{\iota(s-1)}$ rather than $w_k^{\iota_k^{s-1}}$.)

$$\rho_{\text{full}} \equiv \frac{[w_\lambda^{\iota^*}\Pi_{k\neq\lambda}w_k^{\iota(s-1)}]\,\Sigma_{j=1}^M[\Pi_k f_k^{j\to\iota^{**}}]}{[w_\lambda^{\iota(s-1)}\Pi_{k\neq\lambda}w_k^{\iota(s-1)}]\,\Sigma_{j=1}^M[\Pi_k f_k^{j\to\iota(s-1)}]}\cdot\frac{v_\lambda^{\iota(s-1)}\,\Sigma_{j=1}^M f_\lambda^{j\to\iota(s-1)}}{v_\lambda^{\iota^{**}}\,\Sigma_{j=1}^M f_\lambda^{j\to\iota**}} \tag{1.5}$$

Let's consider these terms from left to right, taking the version of each term as it appears in the numerator:

1. $w_\lambda^{\iota^*}$: The likelihood at the locus in question; the term which depends on $\boldsymbol{y}$. I will set $v_\lambda^{\iota^*}$ so as to cancel this out.

2. $\Pi_{k\neq\lambda}w_k^{\iota(s-1)}$: The likelihoods at other loci. These cancel out naturally.

3. $\Sigma_j[\Pi_k f_k^{j\to\iota^{**}}]$: This sum of products term is the heart of the calculation at each MCMC step. Each product is the forward likelihood of the current/proposed hybrid particle conditional on a given history; the sum of products is proportional to the forward likelihood of the current/proposed hybrid particle conditional on $\hat{\tau}^M$.

4. $v_\lambda^{\iota(s-1)}$: Weights that define the proposal distribution and can be chosen arbitrarily (up to normalization).

[3] The dimensionality problems of using ρ_{full} are similar in cause, but different in effects, to those of a bootstrap particle filter. While the bootstrap particle filter suffers from degenerate particle weights as dimension grows, the Finkelstein algorithm does not have particle weights; instead, the stationary distribution of the MCMC, and thus all particles, will be dominated by a single state whose values at all loci come from progressing just a single history.

5. $\Sigma_{j=1}^{M} f_{\lambda}^{j \to \iota(s-1)}$: The part of the proposal density that's due to $\hat{\tau}^M$; the probability density that a given value would have been in $\{\tilde{\boldsymbol{z}}_{\lambda}^{i} : 0 < l < L\}$ to be available to be sampled. To someone used to other variations of particle filtering algorithms, it may seem counterintuitive to include this term; usually, taking advantage of the forward density is a key aspect of how the algorithm works, not something that needs canceling out. However, from the perspective of a Metropolis-Hastings construction, it is necessary to include this in order to target the intended (unnormalized) density 1.4. On a more intuitive level, one might note that the forward density values $f_{\lambda}^{j \to \iota(s-1)}$ for each j appear both here and in the sum of products term in the denominator; so if one did not include this term, that would in a sense be double counting these forward density values by allowing them to cause an increased proposal density and then also increase the acceptance ratio.

The unnormalized proposal weights v_{λ}^{i} can be set at will; as stated above, I let $v_{\lambda}^{i} \equiv w_{\lambda}^{i}$, so that these terms cancel out. Now ρ is just a ratio of sums of products, multiplied by a ratio of the forward mean proposal densities $\bar{f}_{\lambda}^{i}$. I could have set v_{λ}^{i} to also cancel out $\bar{f}_{\lambda}^{i}$, but as seen later, this would slow convergence.

The quantities $\bar{f}_{\lambda}^{i}$ do not change for different MCMC chains or for different steps in each chain and can be precalculated for each i and λ in $O(LM)$ time. We thus have

$$\rho_{\text{full}} \equiv \frac{\Sigma_{j=1}^{M}[\Pi_{k=1}^{L} f_{k}^{j \to \iota^{**}}]}{\Sigma_{j=1}^{M}[\Pi_{k=1}^{L} f_{k}^{j \to \iota(s-1)}]} \frac{\Sigma_{j=1}^{M} f_{\lambda}^{j \to \iota(s-1)}}{\Sigma_{j=1}^{M} f_{\lambda}^{j \to \iota^{**}}} \tag{1.6}$$

By construction, this acceptance probability satisfies the conditions for detailed balance of a Metropolis-Hastings MCMC, and thus converges to the desired target stationary distribution 1.4 under standard regularity assumptions. At convergence, the algorithm will give M samples from the correct target distribution, but with the constraint that the value for each sample at each locus

must be available in $\tilde{\boldsymbol{z}}^{1..M}$. As $M \to \infty$, the set of values available at each locus will become dense, so the conditionality will not be restrictive. Thus, asymptotically, one would expect that each $\boldsymbol{z}^i$ should be a sample from the correct filtering distribution, conditional on $\boldsymbol{x} \sim \hat{\tau}^{M_{t-1}}$.

Yet ρ_{full} is not useful in practice, for two reasons. First, on a relatively trivial level, actually calculating the sum of products terms once for each step of the MCMC would be computationally prohibitive; though not exponential, the resources required would be extreme.

Even more importantly, unless M_{t-1} is exponentially high, $\mathsf{C}_t\mathsf{P}\hat{\tau}^{M_{t-1}}$ is not a good approximation of $\mathsf{C}_t\mathsf{P}\tau$, because of a curse of dimensionality very similar to that which causes the bootstrap particle filter to fail in high dimensions. In ρ_{full}, the acceptance probability is based on a sum over histories of products over loci of likelihoods. Following a similar logic as Bickel *et al.*[7], discussed above, for showing weight degeneracy in the bootstrap particle filter, we see that, assuming that the likelihoods associated with distantly-separated loci are roughly independent, then as dimension increases the distribution of these products over loci will approach a log-normal. Since the variance of that distribution will grow with dimension, the sum is likely to be degenerate unless number of particles grows exponentially with dimension; just one history particle will contribute more to the sum than all others put together.

Consider the example of a weather model of the continental United States, where imperfect measurements of atmospheric conditions are taken daily over a set of cities. In this case, the sum would be degenerate because, although any given proposal particle (weather map for today) might accord well with a given history (possible weather map for yesterday) for some cities, you'd nevertheless need to consider an exponentially large number of possible histories before finding one which accords well across *all* cities with a given realistic present.

In essence, rather than resolving the high-dimensionality problem at time t, we've merely pushed it off to time $t-1$; because of the high variance of the product terms $\Pi_{k=1}^{L} f_k^{j \to \iota^{**}}$, the sums will tend to be dominated by the product term for a single history j, losing most of the benefits of a high number of

particles.

1.3.2 ρ_{local}: Simplifying the product terms by focusing on local neighborhoods

The main computational burden of calculating ρ_{full} comes from the sum over histories of a product over loci. To deal with these computational issues, as well as with the degeneracy of the sum, it would be good to take this product over fewer terms. To do so, I restrict the product over loci to only consider loci in some neighborhood of the locus l which the proposal would change.

This idea gains some support from the decay of correlations property of that Rebeschini and van Handel (2015) demonstrate. This is a complex issue which occupies a significant portion of their paper, but to summarize briefly: they assume particle filters with local dynamics, and both forward densities and observation likelihoods that are strongly bounded away from zero and infinity, Given those assumptions (which they argue are probably stronger than necessary in most practical cases), they show that changing the value at locus k cannot change the conditional distribution of the value at l by more than a quantity that falls exponentially as the distance between k and l increases. Thus, it would seem logical that, in calculating an acceptance probability to target the distribution at l, one may safely ignore faraway loci k.

To use this idea for the Finkelstein algorithm, assume there is a natural distance metric $d(l,k)$ over loci; for instance, if loci were arranged in a square lattice, $d(l,k)$ could be the ℓ_1-distance. Use this distance to define neighborhood balls $\mathcal{B}_r(\lambda) \equiv \{l : d(\lambda, l) \leq r\}$, and use the natural notation that $\boldsymbol{x}_{\mathcal{B}_r(\lambda)} \equiv \{x_l : l \in \mathcal{B}_r(\lambda)\}$. Thus, the new ρ would be:

$$\rho_{\text{local}} \equiv \frac{\Sigma_{j=1}^{M}[\Pi_{k\in\mathcal{B}_r(\lambda)} f_k^{j\to\iota^{**}}]}{\Sigma_{j=1^M}[\Pi_{k\in\mathcal{B}_r(\lambda)} f_k^{j\to\iota(s-1)}]} \frac{\Sigma_{j=1}^{M} f_\lambda^{j\to\iota(s-1)}}{\Sigma_{j=1}^{M} f_\lambda^{j\to\iota^*}} \tag{1.7}$$

If this works, it will have finally conquered the curse of dimensions. For any locus l, there are only $|\mathcal{B}_r(l)|$ terms in each product; a quantity which does not

depend on the overall dimension of the problem, only on the local connectivity. We can therefore choose a fixed M large enough such that this sum of products is not degenerate (not dominated by just one of the products) for $N \equiv \max_l |\mathcal{B}_r(l)|$ loci.

However, it should be noted that with this acceptance probability, the algorithm is no longer strictly speaking Metropolis-Hastings. In particular, the overall MCMC is no longer guaranteed to obey detailed balance. If one repeatedly replaced the values of a single locus l, the MCMC would, by the standard Metropolis-Hastings construction, show detailed balance at a unique stationary distribution with a density proportional to:

$$f_l(z_l|z_1, ..., z_{l-1}, z_{l+1}, ..., z_L) \equiv (\Pi_{\lambda=1}^{L} \mathbb{1}_{z_\lambda \in \{\tilde{z}_\lambda\}}) \Sigma_{j=1}^{M} [\Pi_{k \in \mathcal{B}_r(l)} f_{\mathsf{P}}(z_k|\boldsymbol{x}^j)] \tag{1.8}$$

This function is not only of z_l, but of all z_k such that $k \in \mathcal{B}_r(l)$. However, since this density is not the same for two different values of l, this detailed balance can and will break down. The MCMC is still uniformly ergodic, so a unique stationary distribution still exists; but without detailed balance, we lack the nice guarantees that Metropolis-Hastings would offer as to what that target distribution is. At present, then, my use of ρ_{local}, and all later versions of ρ that build on it, is based on empirical validation, as seen below in the simulation section, not rigorous theory.

1.4 Computational optimizations

1.4.1 ρ_{sampled}: a version of ρ_{local} which replaces numerator and denominator by unbiased estimators

Running the Finkelstein algorithm with acceptance probability ρ_{local} does not require exponential computation, but even polynomial amounts of computation can be daunting in practice. Recall that the of sums over all histories in ρ_{local} are proportional to the forward likelihood conditional on $\hat{\tau}^M$, the M-particle

approximation of the filtering distribution τ. At each step, we can save computation by, effectively, estimating τ with only an arbitrary fixed number $H << M$ particles; that is, by using only H history terms for these sums and using those to get unbiased Horvitz-Thompson estimators of the totals. This leads to ρ_{sampled}. Note that the specific H history particles used will change from step to step and locus to locus in the MCMC, thus taking advantage of the full M particles in equilibrium.

The idea of using unbiased estimators to calculate the Metropolis-Hastings acceptance ratio is not new, and, as Andrieu and Roberts 2009[4] show, this can be made to conserve the stationary distribution, provided that any randomness used in finding the estimator is maintained as part of an expanded Metropolis-Hastings parameter space. This could work for ρ_{full}. But now that we are working from ρ_{local}, this is impossible because the MCMC is already not true Metropolis-Hastings with a single common parameter space. As discussed above, the target distribution of the particle as a whole is different when changing values at different loci, although there's reason to hope that the difference for nearby loci is small. Thus, ρ_{sampled} will inevitably have a different stationary distribution from ρ_{local}. Nevertheless, in the algorithm below, in an attempt to ensure that the stationary distribution changes as little as possible, I expand the parameter space of ρ_{sampled} with a matrix $\boldsymbol{\eta}^s$. This ensures that the same H histories used when accepting a value at a locus are also used when deciding whether to change that value later.

In addition to showing that using unbiased estimators can conserve the target distribution when the parameter space is expanded, Andrieu and Roberts also discuss the case where the parameter space is not expanded. They show that this case still has a stationary distribution, which converges to the original target distribution as more samples are taken. With minor modifications, the same proof applies to the MCMC using ρ_{sampled}; the stationary distribution of ρ_{sampled} is not the same as that of ρ_{local}, but converges to it as $H \to \infty$.

Thus we revise the algorithm from section 2 as follows, expanding the parameter space for use with ρ_{sampled}:

1. As before, assume we have $\hat{\tau}^M$.

2. As before, for each particle $\boldsymbol{x}^i$, progress it to get a full particle $\tilde{\boldsymbol{z}}^i \sim \mathsf{P}\boldsymbol{x}^i$.

3. As before, find likelihood weights $w_l^i \equiv f(y_l|z_l^i)$ and forward densities $f_l^{j\to i} \equiv f_{\mathsf{P}}(z_l^i|\boldsymbol{x}^j)$ for all $i, j \in \{1, \ldots, M\}$ and $l \in \{1, \ldots, L\}$.

4. In parallel, for $k = 1, \ldots, M$, do the following:

   (a) As before, sample $\boldsymbol{\iota}^0 \in \{1..M\}^L$ to initialize the state of the new proposal particle.

   (b) In addition, initialize an $L \times H$ matrix $\boldsymbol{\eta}^0$ with entries in $\{1...M\}$, sampled iid with probabilities

   $$P(\eta_{l,h}^0 = i) = \frac{g(f_l^{i\to\iota_l^0})}{\Sigma_j g(f_l^{j\to\iota_l^0})},$$

   where g is an arbitrary monotonically increasing function.

   Each entry $\eta_{l,h}^0$ gives the index i of one history $\boldsymbol{x}^i$ which we will later use to estimate the denominator of ρ. The significance of g will be explained in more detail below.

   (c) As before, run a Metropolis-Hastings MCMC chain, for steps $s = 1, ..., S$, updating $\boldsymbol{\eta}$ and $\boldsymbol{\iota}$ at each step:

      i. As before, choose a spatial locus $\lambda(s)$ (aka λ) uniformly at random.

      ii. As before, sample a proposed replacement $\iota^*(s)$ (aka ι^*) for locus λ, with probability

      $$P(\iota^* = i) = w_\lambda^i / \sum_j w_\lambda^j.$$

      Also, define $\boldsymbol{\iota}^{**}$ as before.

iii. In addition, sample (iid) a set $(\eta_1^*, ..., \eta_H^*)$ of histories by which to judge this proposed replacement, where

$$P(\eta_h^* = i) = \frac{g(f_\lambda^{i\to\iota^*})}{\Sigma_j g(f_\lambda^{j\to\iota^*})}.$$

Define the matrix $\boldsymbol{\eta}^{**}$ by

$$\eta_{l,h}^{**} := \begin{cases} \eta_h^* & \text{if } l = \lambda \\ \eta_{l,h}^{s-1} & \text{if } l \neq \lambda \end{cases}$$

iv. Finally, define

$$\rho_{\text{sampled}} \equiv \frac{\Sigma_{h\in\{1..H\}} \frac{1}{g_\lambda(\eta_h^*, \iota^{**})} [\Pi_{l\in\mathcal{B}_r(\lambda)} f_l^{\eta_h^*\to\iota^{**}}]}{\Sigma_{h\in\{1..H\}} \frac{1}{g_\lambda(\eta_h^*, \iota^{s-1})} [\Pi_{l\in\mathcal{B}_r(\lambda)} f_l^{\eta_h^*\to\iota^{s-1}}]} \frac{\Sigma_{j=1}^M f_\lambda^{j\to\iota^{s-1}}}{\Sigma_{j=1}^M f_\lambda^{j\to\iota^{**}}}, \tag{1.9}$$

where

$$g_l(i,j) \equiv \frac{g(f_l^{i\to j})}{\Sigma_k g(f_l^{k\to j})}.$$

As usual, we accept the proposed replacement with M-H probability $1 \wedge \rho_{\text{sampled}}$. In case of acceptance, let $\boldsymbol{\iota}^s := \boldsymbol{\iota}^{**}$ and $\boldsymbol{\eta}^s := \boldsymbol{\eta}^{**}$. Otherwise, make no change, so that $\boldsymbol{\iota}^s := \boldsymbol{\iota}^{s-1}$ and $\boldsymbol{\eta}^s := \boldsymbol{\eta}^{s-1}$.

(d) As before, set $z_l^k = \tilde{z}_l^{\iota_l^S}$.

5. As before, the final set of particles forms

$$\hat{\pi} \equiv \frac{1}{M} \sum_{i=1}^M \delta[(z_1^i, ..., z_L^i)].$$

A few words on the function g which defines the probability of considering a given history i in determining the acceptance probability. First, note that the sampling probabilities for $\boldsymbol{\eta}^s$ are in principle arbitrary, so could be a function of

the full current vector $\boldsymbol{\iota}^s$. In the above discussion, however, they are a function of only $f_l^{i \to \iota^*}$; this simplifies both notation and computation. Second, g should be chosen to be some non-decreasing function of $f_\lambda^{i \to \iota^*}$, so that the variance in $\Pi_{l \in \mathcal{B}_r(\lambda)} f_l^{\eta_h^* \to \iota^{**}}$ is at least partially offset by that in g, increasing the efficiency of the estimation process. Another way of saying this is that we should make it more likely to sample plausible histories than implausible ones. Possible choices for g are discussed in Section 1.4.3 below. Whatever g is chosen, the denominator $\Sigma_j g(f_\lambda^{j \to \iota^*})$ can be precalculated for each possible choice of ι^*, meaning that this does not meaningfully increase computing requirements per MCMC step.

How much computation does ρ_{sampled} save? The ρ_{local} algorithm requires running M different MCMC chains, with each of L loci going through S steps, and at each step calculating a ρ using a sum over M histories of a product over the up to N locations in the relevant $\mathcal{B}_r(l)$. The total computation cost is at least $O(M^2LNS)$. This is better than $O(M^2L^2S)$ that ρ_{full} would have taken, but still somewhat burdensome. To get ρ_{sampled}, on the other hand, we only calculate the product of locus likelihoods for an arbitrary number H of histories rather than all M of them. Thus, the total computation cost falls to $O(MHLNS)$; since H and N are arbitrary constants that can be set independently from the full size M and L respectively, this is a substantial improvement.

1.4.2 Discussion of proposal weights

In the above discussion, the proposal weights v_λ^i, used by all versions of ρ, are arbitrary. That is, any proposals with nonzero weights could be used; the v_λ^i are accounted for out in the $\bar{F}$ term.

Ideally, these proposal weights should both be tuned for maximum efficiency of the MCMC; that is, insofar as it does not substantially increase the computational costs per step, to try to ensure that the variance of the acceptance probability is as low as possible (approaching the Gibbs sampling case where it's uniformly 1) while maintaining disperse (high-variance) proposal values for good ergodicity/mixing.

For v_λ^i, that is similar to the idea of an optimal proposal distribution, which is common in the particle filter literature.[50] The low-dimensional bootstrap particle filter uses $(z|x^{1..M})$ as a proposal, then reweights using $(y|z)$. In such a case, the idea of an ideal proposal distribution is that if you could propose from $(z|x^{1..M}, y)$, the reweighting step would not be necessary.

Applying a similar idea to v_λ^i, it becomes clear why I have set it equal to w_λ^i. Of course, including a factor of w_λ^i in v_λ^i helps these terms cancel and thus simplifies the calculation of ρ. But if simplicity of calculating ρ were the only consideration, I could have set v_λ^i to $w_\lambda^i \Sigma_{j=1}^M f_\lambda^{j\to i}$, so that the ratio $\frac{\Sigma_{j=1}^M f_\lambda^{j\to\iota(s-1)}}{\Sigma_{j=1}^M f_\lambda^{j\to\iota(s)*}}$ would cancel out too.

But setting $v_\lambda^i \equiv w_\lambda^i$ ensures that the proposal density for z_λ, conditional on $\boldsymbol{x}^{1..M}$ and $\boldsymbol{y}$, is $(z_\lambda|\boldsymbol{x}^{1..M}, y_\lambda)$ — not too far from the ideal $(z_\lambda|\boldsymbol{x}^{1..M}, \boldsymbol{y}, \boldsymbol{\zeta}_{\mathcal{B}_r(\lambda)\setminus\lambda}^{s-1})$ which would allow an acceptance probability of uniformly 1. That's because the density of $(z_\lambda|\boldsymbol{x}^{1..M})$ is included implicitly through the progression procedure, while that of $(z_\lambda|y_\lambda)$ is handled explicitly through w_λ^i.

1.4.3 Refining the history sampling weights

What about the history sampling weights $g_l^{i\to j}$? As above, these are arbitrary. Ideally, to minimize the variance of the acceptance probability, they would approximate:

$$\begin{aligned} g_l^{i\to j}(\boldsymbol{\zeta}_{\mathcal{B}_r(l)\setminus l}^{s-1}) &\propto f_{\mathrm{P}}(\tilde{z}_l^j|\boldsymbol{x}^i, \boldsymbol{\zeta}_{\mathcal{B}_r(l)}^{s-1}) \\ &= \Pi_{\kappa\in\mathcal{B}_r(l)} f_\kappa^{i\to j} \end{aligned} \tag{1.10}$$

... because this expression in the Horvitz-Thompson inverse sampling weight term would cancel exactly with its relative contribution to the estimated sum of products, so that the overall estimator would be governed solely by the sum of weights term in the denominator.

It is computationally infeasible to calculate these quantities exactly for each step of the MCMC, so I use $g_l^{i \to j}$. In the simulation below, I've tested two formulas for these weights:

1. $g_{\text{uniform}}(x) = 1$
2. $g_{\text{bentlog}}(x) = \frac{\log(f_\lambda^{i \to j}) - \log(\min(f_\lambda^{i \to j})}{\alpha} + max(0, \log(x) - \log(\max(x) + \beta)$, where $\min(x)$ and $\max(x)$ are the precalculated minimum and maximum values of $f_\lambda^{i \to j}$ and α, β are positive constants.

Both of these options are simply computationally-convenient first attempts; though simulations show g_{bentlog} is an improvement over g_{uniform}, it is surely not optimal in this regard. In further work, I will look into using proposal distributions that are conditional on the current values at other loci, not just on the observations at the current locus.

1.4.4 Theoretical limitations of the algorithms in this paper

Both the block particle filter and the Finkelstein particle filter proposed here are intended to deal with the curse of dimensionality. However, both may fail in cases where forward densities — that is, the relative probabilities of given states at time t conditional on the state at time $t-1$ — are concentrated around particular values, and thus insufficiently ergodic. Rebeschini and Van Handel's error bounds rely on a strong ergodicity assumption, bounding the forward density away from 0 in a way that they themselves acknowledge is unrealistic in real-world cases. In a separate paper, they explore further the kind of problems that can arise when this assumption does not apply, and the regime where that failure occurs in practice.[45]

For the Finkelstein particle filter, I am not giving any formal proofs of performance, but it is clear that if ergodicity is poor enough, my algorithm will also break down. For example, suppose that the forward density from history $\boldsymbol{x}^j$ to raw locus value z_l^i is less than ϵ if $i \neq j$, and otherwise greater than 100ϵ. In that case, the MCMC will strongly tend to get stuck in states where all locus

values come from the same history. Metaphorically, Sally Finkelstein would be too picky, never selecting nearby body parts that didn't match perfectly. The result would then reduce to the bootstrap particle filter, with more useless computational cost.

Do the real-world problems to which these algorithms are applied have enough ergodicity for them to function? For these algorithms to be appropriate, we'd need a situation with enough nonlinear effects that simple Kalman filters don't suffice; yet also one which still has plenty of new randomness at each time step, such that even if the forward density is not actually strongly ergodic, it is at least diffuse enough for these algorithms to work. In SLAM (simultaneous location and mapping) models for robotics applications, such situations arise. But in fluid dynamics models, chaotic dynamics are the rule. Such models can be deterministic or nearly so, with highly concentrated forward densities, yet still have interesting dynamics. Uncertainty in the initial conditions is amplified at each time step, so even in a deterministic model with new measurement tending to reduce the uncertainty at each time step, the uncertainty will rise again by the next time step, and so overall uncertainty can remain in equilibrium.

Due to the "picky Sally" problem explained just above, the Finkelstein algorithm as explained in current paper does not deal well with such deterministic or nearly-deterministic models. However, in a follow-up paper, I will offer a modification of this algorithm to address such models.

1.5 Numerical simulations

1.5.1 Setup

Filtering algorithms cannot be expected to precisely infer the underlying true state of the hidden Markov model. Instead, the goal is merely to infer its conditional distribution, and most of the interest of the problem lies in the fact that this distribution remains non-degenerate; as we acquire more information in order to narrow down the possible states, the state itself evolves, so we never

catch up.

Thus, we cannot simply follow the recipe of a more traditional simulation study of a technique for parameter inference. In traditional parameter inference, the object of interest is the true parameter value(s), which can be arbitrarily chosen when running a simulation. Although inference algorithms may yield an inferred distribution for the parameter(s), the interpretation of this distribution as a confidence distribution (for frequentist methods) or a credible distribution (for Bayesian methods) is in some sense not inherent to the problem; for example, in the case of Bayesian methods, a credible interval is only as valid as the priors that produce it. However, in this case, we are not making arbitrary assumptions in order to get the best or the most robust performance; the assumptions are given by the problem, and the aim is to calculate a true probability distribution. In order to efficiently measure an algorithm's performance, we'd like a setting where the correct value of the object of interest — not the point value, but the conditional distribution — is known.

So, in order to run a simulation study, I fall back on a linear Gaussian model, where the Kalman filter algorithm gives an analytically-correct filtering distribution. Of course, given that such an analytic solution does exist, one would never in practice use an inexact filtering algorithm such as those discussed by this paper. However, the ability of our more-general algorithm to roughly reproduce the results of a Kalman filter is, at the least, encouraging.

The particular linear Gaussian model I use is a model based on a progression matrix P, a novelty matrix N, and a measurement error covariance matrix E:

$$\begin{aligned} z &= Px + \delta \\ y &= z + \epsilon \\ \delta &\sim \mathcal{N}(0, N) \\ \epsilon &\sim \mathcal{N}(0, E) \end{aligned} \tag{1.11}$$

P is a tridiagonal matrix. N and E are diagonal matrices with periodic

structure, so that some loci are best learned about through their neighbors. Specifically, N's elements alternate between a higher and a lower variance, each value occurring at every 2nd locus, while E has a lower variance at every 5th locus.:

$$P \equiv \begin{bmatrix} b & c & & & 0 \\ a & b & c & & \\ & a & b & \ddots & \\ & & \ddots & \ddots & c \\ 0 & & & a & b \end{bmatrix} \tag{1.12}$$

$$a \equiv .4; b \equiv .35; c \equiv .05; a + b + c = .8 < 1$$

$$N \equiv \begin{bmatrix} 1 & & & & & & 0 \\ & q & & & & & \\ & & 1 & & & & \\ & & & q & & & \\ & & & & \ddots & & \\ 0 & & & & & & q \end{bmatrix}$$

$$q = .25$$

$$E \equiv \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ 4 \\ 5 \\ 6 \\ \vdots \\ d \end{array} \begin{array}{c} \begin{array}{ccccccccc} 1 & 2 & \dots & 4 & 5 & 6 & \dots & d \end{array} \\ \begin{bmatrix} e & & & & & & & 0 \\ & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & 1 & & & & \\ & & & & e & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ 0 & & & & & & & 1 \end{bmatrix} \end{array}$$

$$e = .16$$

The state was initialized at mean 0 and variance 5 independently at each locus. The model was run for 10 time steps, and for the particle filtering algorithms the outcome variables of 5 separate runs were averaged.

A number of parameters were tried for the algorithms, but a good set of numbers for comparing different models was: 400 particles for the Finkelstein variants, $400^2 = 160000$ particles for the bootstrap particle filter, and $400^2/5 = 32000$ particles for the block particle filter algorithm. These numbers were chosen so that each algorithm would take roughly comparable computing time; the only step that requires computing power that is quadratic in the number of particles is pre-calculating the forward densities $f_l^{j\to i}$ in the Finkelstein algorithm.

All results for Figures 1.5.2 and 1.5.3 are for a 30-dimensional model. Results for both Finkelstein and block particle filter algorithms remained materially similar as model dimension was varied from 30 to 90, demonstrating that the Finkelstein and Frankenstein algorithms' errors are roughly independent of dimension, as expected.

The Finkelstein algorithm was used with ρ_{sampled}, with 45 histories per location and two formulas for the history sampling probabilities g (g_{uniform}, and g_{bentlog} with $\alpha = \beta = 5$). The neighborhood width was $r \equiv 1$, which is to say that $\max |\mathcal{B}_r(l)| = 3$. Similarly, the zone size for the block particle filter algorithm was 3.

Note that I am not the first to simulate outcomes for the block particle filter. Although Rebeschini and van Handel's paper originally proposing it relied on proofs rather than simulations, more recent papers have implemented it and given results.[35] [15] [57] The results there are not directly comparable with those given here due to different models used.

1.5.2 Results

To get an intuition for this situation, I will begin by showing the evolution of a single run of the model. The top panel of figure 1.5.1 shows the evolution over

time of the true value ($\sum_{l=3}^{5} z_l$), the observed value ($\sum_{l=3}^{5} y_l$), and the filtering distribution as calculated by a Kalman filter ($\boldsymbol{E}_\pi(\sum_{l=3}^{5} z_l | \boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)$), for the sum of loci 3-5. The bottom panel shows the the mean of the estimated filtering distribution for each of the four algorithms I tested — Kalman filter (analytically correct), bootstrap particle filter, block particle filter, and Finkelstein particle filter.

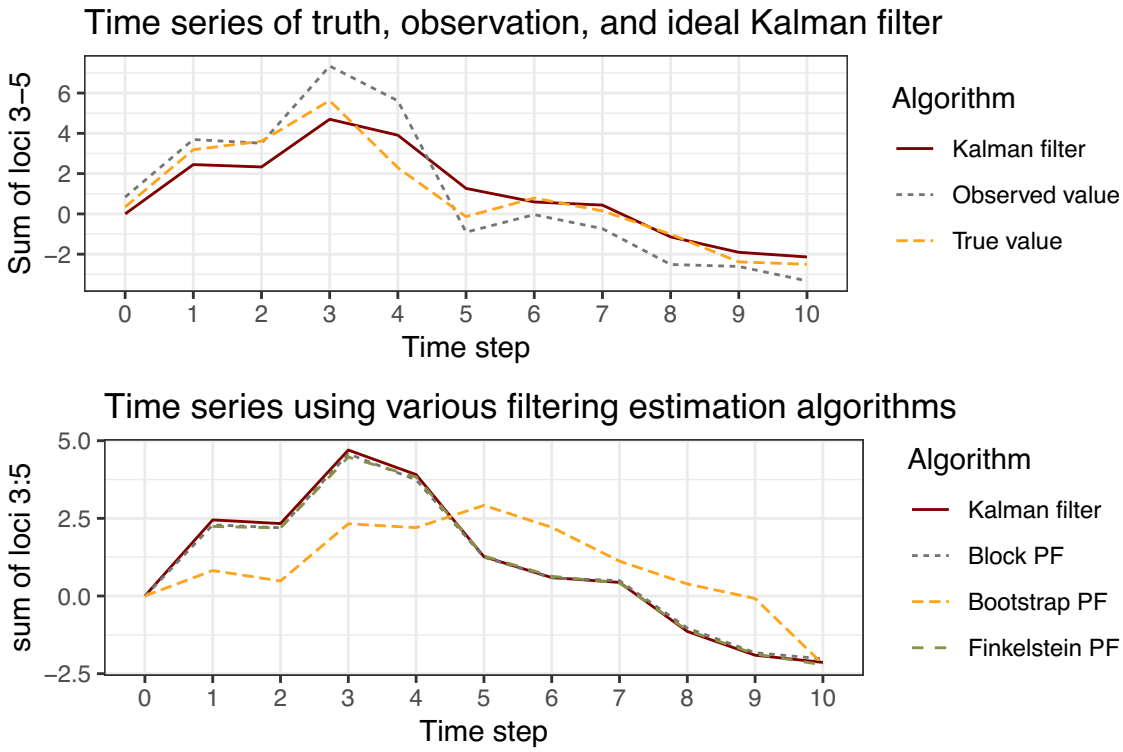


Figure 1.5.1: Time series from a single run of each algorithm for 10 time steps, in a model with 90 dimensions. Parameters for each algorithm are given in text.

In the upper panel of Figure 1.5.1, one can see that the observations vary relatively widely around the truth, while the Kalman filter mean follows those observations with more conservative moves, thus staying closer to the true value. In the lower panel, one can see that the bootstrap particle filter falls to the curse

of dimensionality; while the block and Finkelstein particle filters both manage to approximate the correct Kalman filter distribution relatively well.

To compare outcomes of the two working algorithms in greater depth, Figure 1.5.2 shows the average squared error per locus: that is, the squared difference between the estimated distribution mean and the correct mean as given by the Kalman filter, conditional on a single fixed series of observations $(y_1, ..., y_{10})$. Note that we are measuring error relative to the mean of the Kalman filter rather than to z_l; this is because the Kalman filter result is the ideal filtering distribution that we are trying to capture here. Though it's not visible in these graphs, both algorithms do a relatively good job of reproducing the variance of the filtering distribution; this is within 3% of the true value for both algorithms.

Figure 1.5.2 makes two things clear. First, there is a bias/variance tradeoff between the block particle filter and the Finkelstein particle filter; with comparable run times, the block filter has a higher bias but almost no variance in its distribution error. Second, as expected, the performance of the block particle filter differs for loci that are central to their neighborhood Z_j, as opposed to loci that are on the border of their neighborhood. (The small apparent difference in performance of the Finkelstein algorithm between the two kinds of loci is largely an artifact of the specific realization of $(y_1, ..., y_{10})$ that was used to generate this graph. The Finkelstein algorithm does not use fixed neighborhoods $\{Z_j\}$, so the distinction between central and peripheral is simply does not apply, except for the first and last loci overall.)

The error of the Finkelstein algorithm, like that of the block particle filter, appears to remain stable over time, as seen in Figure 1.5.3. This shows the time evolution of the KL divergence between the true filtering distribution, as calculated using the Kalman filter, and a Gaussian with mean vector and covariance matrix inferred from a Finkelstein particle filter. It appears that uniform sampling (g_{uniform}) can occasionally be unsuccessful in sampling good histories, as reflected by the spikes in that line; log sampling (g_{bentlog}) had

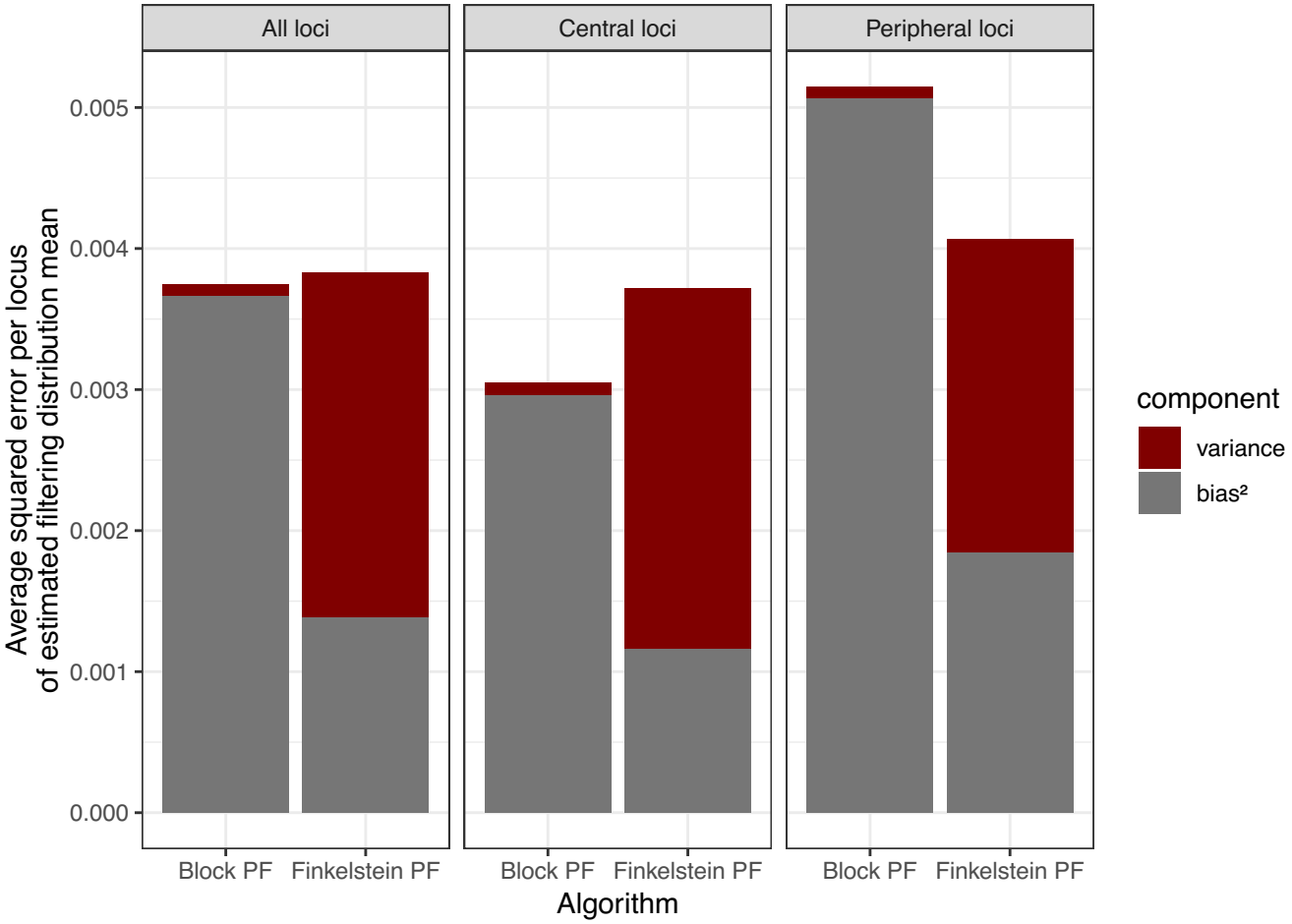


Figure 1.5.2: Breakdown of average squared error per locus.

superior stability.

1.6 Conclusion

I have introduced the novel Finkelstein particle filtering algorithm for estimating the filtering distribution of models with high dimensionality due to large spatial extent. In such models, the simple bootstrap particle filtering algorithm is unusable. But, as with the previously-proposed block particle filter, my algorithm

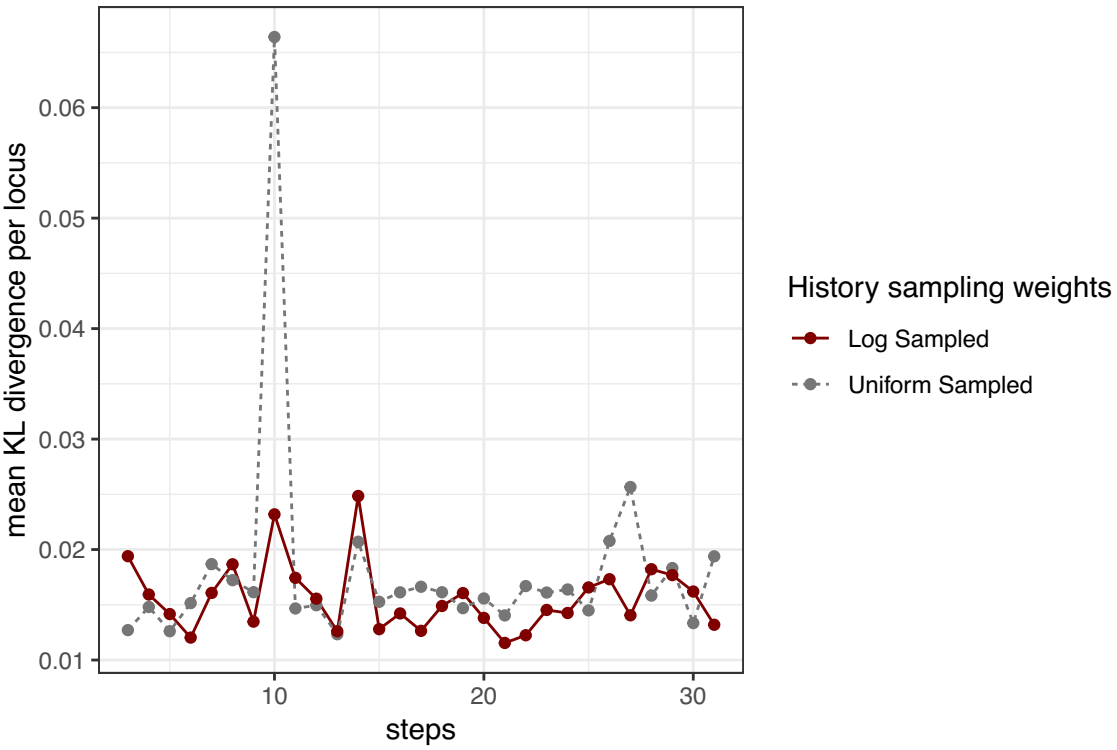


Figure 1.5.3: Stability of KL divergence across time steps.

relies on the locality of dynamics to resolve this problem, focusing on a small area at a time.

Using simulations, I have showed that the error of means of my algorithm has lower bias but higher variance than the block particle filter, given comparable parameters. All in all, the total squared error of means of the Finkelstein algorithm is more homogeneous across loci than that of the block algorithm; lower for loci peripheral to a neighborhood in the block particle filter, but higher for those which are central. I also give empirical evidence that the error of this algorithm is stable over time, making it a candidate for online data assimilation tasks.

It is commonplace to prefer variance over bias when such a tradeoff is possible, because this allows improving precision with additional computing power by

independent reruns of the algorithm. That improved precision would certainly be possible in this case with the Finkelstein algorithm. This picture is slightly complicated by the fact that such computing power might enable better results from the block particle filter by increasing the neighborhood size. But there are several problems with just increasing neighborhood size. Above all, computing power (that is, number of particles) needed could be up to exponential in neighborhood size, while it's just quadratic in number of Finkelstein particles or linear in independent Finkelstein runs. Second, unlike number of particles, neighborhood size comes in sizeable discrete intervals; it may not be possible to effectively use a small additional amount of computing power. And finally, to reduce the bias of the block particle filter, neighborhood size must be increased up-front, while the variance Finkelstein particle filter can in be reduced by independent runs (perhaps even by different scientists).

Thus, I believe that the Finkelstein particle filter algorithm offers meaningful advantages over prior proposals. In future work, I will extend this to cover chaotic dynamics in a deterministic or quasi-deterministic model.

2

Laplace Family Variational Inference for Independent Latent Variable Models

In this chapter, we provide an approach for approximate Bayesian inference in latent variable models: models where, in addition to a set of global parameters, there is a separate vector of latent parameters for each observation. The potentially large number of model parameters can cause difficulties with traditional inference techniques.

We approach the problem through variational inference (VI); that is, we aim to approximate the model posterior (in the sense of minimizing KL-divergence) with a parametric distribution from some variational family. Variational inference is an alternative to MCMC (Markov Chain Monte Carlo). While MCMC converges (under broad conditions) to give samples from the true distribution of interest, it can be impractical with high-dimensional models. Variational

inference, on the other hand, relies on approximations, but may be more practical and/or faster.[1]

Our key innovation is to define a new variational "guide" family of multivariate normal distributions, which we call the *Laplace family*. The Laplace family has a relatively small number of variational parameters, yet contains good approximations to the posterior (in the sense described below). Importantly, unlike the commonly used mean-field approximation, the Laplace family captures the fact that the model parameters are generally *not* independent. This is important to accommodate the fact that, even in cases where they were independent *a priori*, conditioning on observations induces dependence in the posterior.

After introducing this guide family, we show how two standard methods for speeding up variational inference — stochastic VI and amortization (variational autoencoding) — can be incorporated in this context. We also broaden the idea of amortization to include "analytic amortization", useful when the individual distributions that make up the model can be solved analytically for their maximum likelihood values.

2.1 Defining Notation: Latent Variable Models and Variational Inference

2.1.1 Latent Variable (LV) Models

For our purposes, a latent variable model consists of 3 core elements:

- a vector $\boldsymbol{\gamma}$ of global parameters, with $\boldsymbol{\gamma} \in \boldsymbol{\Gamma} \cong \mathbb{R}^g$, assumed distributed by the prior density $p(\boldsymbol{\gamma})$
- vectors $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_N$ of latent parameters, drawn independently from a parameter space $\boldsymbol{\Lambda} \cong \mathbb{R}^l$ where $\boldsymbol{\lambda}_i$ is distributed by the density

[1] A third alternative is SMC (Sequential Monte Carlo) methods. Though these techniques are promising in the context of high-dimensional models, they are not yet as mature and easy-to-apply as MCMC, and we will not discuss them further.

$p(\boldsymbol{\lambda}|\boldsymbol{\gamma}, \chi_i)$; here, χ_i represents known covariates for unit i, which will be implicit hereafter.

- observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, drawn independently from $p(\boldsymbol{x}|\boldsymbol{\lambda}, \boldsymbol{\gamma})$, with $\boldsymbol{x}_i \sim p(\boldsymbol{x}|\boldsymbol{\lambda}_i, \boldsymbol{\gamma})$.

In other words,

$$p\left(\boldsymbol{\gamma}, \boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_N, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\right) = p(\boldsymbol{\gamma}) \prod_{i=1}^{N} p(\boldsymbol{\lambda}_i|\boldsymbol{\gamma})\ p(\boldsymbol{x}_i|\boldsymbol{\lambda}_i, \boldsymbol{\gamma}) \tag{2.1}$$

which implies that

$$\left[(\boldsymbol{x}_i, \boldsymbol{\lambda}_i) \perp\!\!\!\perp (\boldsymbol{x}_j, \boldsymbol{\lambda}_j)\right] \Big| \boldsymbol{\gamma} \text{ for } i \neq j. \tag{2.2}$$

where $p(\boldsymbol{\gamma})$ represents the prior(s) on $\boldsymbol{\gamma}$.

In some latent variable models, the global parameters $\boldsymbol{\gamma}$ are the quantities of interest, while the latent parameters $\boldsymbol{\lambda}$ are merely nuisances; in others, it is the other way around. In either case, the motivation for using the full model is the hope that including both kinds of parameters will improve inference overall. That is to say, the goal of inference is to understand the full posterior distribution over all $d := g + Nl$ model parameters.

The techniques described in this paper require that each of the distributions $p(\boldsymbol{\gamma})$, $p(\boldsymbol{\lambda}|\boldsymbol{\gamma})$, and $p(\boldsymbol{x}|\boldsymbol{\lambda}, \boldsymbol{\gamma})$ have thrice differentiable density functions. Unlike in conjugacy-based variational algorithms, we do not require that they come from exponential families. However, for the purposes of analytic amortization (see section 2.3.5), we do prefer that, for given $\boldsymbol{\gamma}$ and i, there be an analytic solution for the conditional MAP (maximum *a posteori*)

$$\boldsymbol{\lambda}_i^* = \arg\max_{\boldsymbol{\lambda}} p(\boldsymbol{x}_i|\boldsymbol{\lambda}, \boldsymbol{\gamma}), \tag{2.3}$$

or at least an easily-computable approximation.

2.1.2 Variational Inference (VI)

Suppose we have a set of observations $\boldsymbol{x}$ and a model for these observations with parameters $\boldsymbol{\theta} \in \mathbb{R}^d$. In other words, we are given a prior distribution $p(\boldsymbol{\theta})$ and a likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$. We are interested in the posterior distribution

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int_{\boldsymbol{\theta'}} p(\boldsymbol{x}|\boldsymbol{\theta'})p(\boldsymbol{\theta'})d\boldsymbol{\theta'}}. \tag{2.4}$$

The problem is that, in general, the integral in the denominator is very difficult to compute or even to estimate — especially when d, the dimensionality of $\boldsymbol{\theta}$, is high.

The variational approach is to approximate the posterior distribution by a sampleable **guide distribution** $q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ belonging to some **guide family** $\mathcal{Q}_\Phi$, parametrized by the vector of **guide parameters** $\boldsymbol{\phi} \in \Phi$.[2] [59] To find the best approximation, we look for the value $\hat{\boldsymbol{\phi}} \in \Phi$ that minimizes the Kullback-Leibler (KL) divergence between $q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ and our target posterior distribution:

$$\hat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\phi}} \left[D_{\mathrm{KL}} \left(q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \,\middle|\middle|\, p(\boldsymbol{\theta}|\boldsymbol{x}) \right) \right]. \tag{2.5}$$

We can then estimate posterior quantities of interest by using samples from the fitted guide in place of samples from the posterior. Importance-weighting these samples by the ratio of the unnormalized posterior density to the guide density can give a further incremental improvement to the estimation.

Minimizing $D_{\mathrm{KL}} \left(q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \,\middle|\middle|\, p(\boldsymbol{\theta}|\boldsymbol{x}) \right)$ turns out to be equivalent to maximizing an expression known as the **ELBO** (or **variational free energy**) of

[2] In other literature, these are sometimes known as the *variational distribution*, *variational family*, and *variational parameters* respectively.

$p(\boldsymbol{\theta}, \boldsymbol{x})$ with respect to $q_{\phi}(\boldsymbol{\theta})$:

$$= \int [\log p(\boldsymbol{x}, \boldsymbol{\theta})] q_{\phi}(\boldsymbol{\theta}) d\boldsymbol{\theta} - \int [\log q_{\phi}(\boldsymbol{\theta})] q_{\phi}(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{2.6}$$

$$= \log \left(\int p(\boldsymbol{x}, \boldsymbol{\theta}) d\boldsymbol{\theta} \right) - D_{\mathrm{KL}} \left(q_{\phi}(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta}|\boldsymbol{x}) \right) \tag{2.7}$$

The name ELBO stands for **evidence lower bound**, because the first term in (2.7), the log of the normalizing constant of the posterior distribution, is sometimes called the **evidence**.

Note that the choice of minimizing $D_{\mathrm{KL}} \left(q_{\phi} \,||\, p \right)$ rather than $D_{\mathrm{KL}} \left(p \,||\, q_{\phi} \right)$ is motivated solely by computational tractability: since we know how to sample from q_{ϕ}, it is easy to estimate expectations over that distribution. If we could somehow integrate over the posterior $p(\boldsymbol{\theta}|\boldsymbol{x})$, we could compute the **evidence upper bound** or **EUBO**[3]:

$$\mathrm{EUBO}(\phi) := E_{p(\boldsymbol{\theta}|\boldsymbol{x})} \left[\log p(\boldsymbol{x}, \boldsymbol{\theta}) - \log q_{\phi}(\boldsymbol{\theta}) \right] \tag{2.8}$$

$$= \log \left(\int p(\boldsymbol{x}, \boldsymbol{\theta}) d\boldsymbol{\theta} \right) + D_{\mathrm{KL}} \left(p(\boldsymbol{\theta}|\boldsymbol{x}) \,||\, q_{\phi}(\boldsymbol{\theta}) \right) \tag{2.9}$$

The EUBO is actually a more useful quantity than the ELBO, as it can help bound the error when estimating posterior quantities based on importance-weighted samples from the fitted guide[12, 34]. Later in this paper, we will use the EUBO to evaluate the quality of our variational approximation to a posterior distribution for which we can get sufficient MCMC samples to treat as known. In general, however, the EUBO is almost never used, since in order to estimate it, we would need to sample from the very distribution that we are trying to approximate.

The first term of the ELBO, known as the **energy**, encourages a choice of q_{ϕ}

[3] Unlike "ELBO", the term "EUBO" is not standard.

whose regions of high probability coincide with regions where $p(\boldsymbol{\theta}|\boldsymbol{x}) \propto p(\boldsymbol{\theta}, \boldsymbol{x})$ is high. If we only maximized the energy, while allowing unrestricted distributions as our guide, the optimum would be a delta distribution with probability mass concentrated at the maximum of $p(\boldsymbol{\theta}|\boldsymbol{x})$,[4] which could be a problem if this local maximum accounted for only a small portion of the total probability mass of the posterior. This is why we also include the second term, known as the **entropy**, which encourages a choice of $\boldsymbol{\phi}$ that spreads out the probability mass of $q_{\boldsymbol{\phi}}$ as much as possible. Using the resulting fitted guide $q_{\hat{\boldsymbol{\phi}}}(\boldsymbol{\theta})$ as an approximation of $p(\boldsymbol{\theta}|\boldsymbol{x})$ is usually a better way to estimate quantities of interest than using the MAP (maximal *a posteriori*) point estimate.

Note that the ELBO is itself an intractable integral, except in a few special cases that allow closed-form solutions using conjugate distributions. When no closed-form solution exists, we estimate the integral by one or more samples from the guide. This is known as **Black-Box Variational Inference (BBVI)**[42]. The term "black-box" draws a contrast with older forms of VI that only worked for exponential family models with conjugate distributions. The number m of guide samples used for estimation is an arbitrary parameter to be set by the researcher.

2.2 Common Types of Guide Families

2.2.1 Mean-field and normal guide families

A key part of variational inference is choosing an appropriate guide family $\mathcal{Q}_\Phi$. The most common choice of guide family is some form of **mean-field family**: a product of independent exponential-family distributions for each model parameter θ_i[39]. There are, of course, some cases where this assumed posterior independence holds or nearly holds, but such cases are, if anything, the exception. Consider Berkson's paradox: two parameters that are *a priori* independent but both positively correlated with an observable will become

[4] Assuming, of course, that a delta distribution was in the guide family (or, if our definition of "optimum" includes limits, in the closure thereof).

negatively correlated with each other after conditioning on that observable. This sort of problem arises frequently in hierarchical models.

Our goal is to give a general method for constructing guide distributions that *do* capture the correlations among the model parameters. The natural choice is to pick a family of Gaussians: unlike many other distribution families, Gaussians are naturally multivariate, and make it easy to control correlation structure. Moreover, asymptotic theory shows that under loose regularity conditions, posterior distributions of continuous parameters tend towards normality. (Although these asymptotics only apply to the global parameters in our model, not to the latent variables.)

Of course, in restricting our attention to normal guide families, we may be foregoing the chance to choose conjugate distributions and/or ones that naturally have the correct support. Generally, we deal with issues of support by using transformed parameters whose support extends over the full real line. For example, if a parameter in the model is restricted to the positive real numbers, we can transform it to its logarithm, resulting in a guide that is effectively lognormal over the untransformed parameter. If these transformations are smooth, much of the asymptotic theory mentioned above still holds.

For multivariate normal guide families, the mean-field restriction of independence is equivalent to restricting the covariance matrix of the posterior on the parameters, which we'll call Σ, to be diagonal. Thus, a Gaussian mean-field family has $2D$ guide parameters; a mean and a variance for each of the d model parameters. This gives a poor fit if posterior correlations are significant. It is not hard to show that, if the true posterior approaches a multivariate Gaussian, the optimal mean-field approximation approaches the conditional variance of each component. Figure 2.2.1, reproduced from Figure 1 of [10], shows this problem graphically.

Even if the true posterior is not quite Gaussian, by the law of total variance ("Eve's law"), conditional variances are systematically (though not necessarily uniformly) lower than marginal ones. Thus, a mean-field guide, which assumes

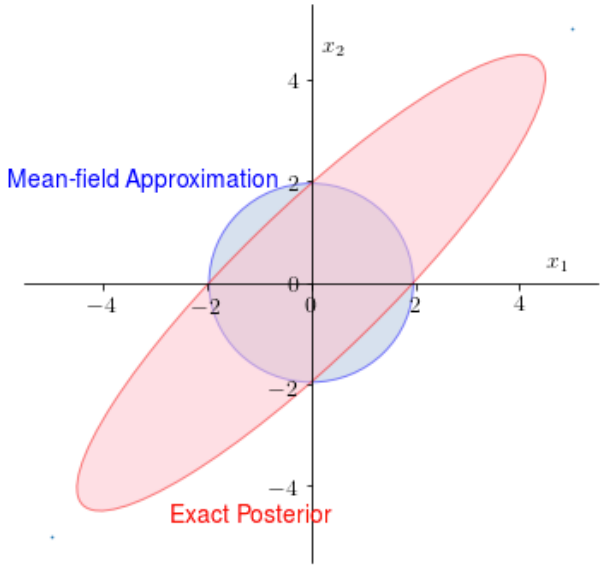


Figure 2.2.1: Stylized image of a credible set of a 2-dimensional correlated Gaussian posterior, and the optimal mean-field approximation thereof. (The ellipses are 2σ contours of the relevant Gaussian distributions, where the mean-field distribution has conditional variance of 1 in each dimension and correlation of .9. This figure is reproduced from Figure 1 of [10].)

no correlation between model parameters, will systematically underestimate their posterior marginal variance. Since estimating posterior marginal variances is often of primary interest in Bayesian analysis, this is a significant concern.

2.2.2 Non-mean-field guide families: prior work

One obvious way to address mean-field's problem with correlations would be to use an **unconstrained Gaussian** guide family; that is, the fitted posterior could be any multivariate Gaussian distribution on $\mathbb{R}^d$, with no artificial limitations on the form of its covariance matrix Σ. However, this means that the guide family will have $\mathcal{O}(d^2)$ guide parameters. Without further restrictions or assumptions, this is impractical for LV models.

Copula VI[21][53] allows arbitrary correlation structure in the guide, and unlike our proposal below, works even when the components of the guide are not

all Gaussian. However, like the unconstrained Gaussian approach described above, the dimension of the resulting guide family is $\mathcal{O}(d^2)$. Thus, it does not really resolve the issue of latent variable models requiring dimensionality that is quadratic in d, but rather that of non-normality. It may therefore be complementary with our approach below.

For time-series models, correlations across time are often of primary importance. A number of model-specific approaches to incorporating such correlations into the guide have been developed; for a survey, see [59], p. 12. Our approach is more general than these.

Hierarchical VI[43] uses a mean-field guide, but then places a prior on the guide and marginalizes the guide itself out; this allows dependencies among model parameters to be reflected in the inferred "guide hyperparameters". This is an interesting approach, but so far, we do not believe it has been applied in a black-box context. It may be difficult to extend this approach to cases where model distributions do not have known conjugate distributions.

An interesting compromise between full-rank correlations and mean-field is taken by Miller et al. as part of their Variational Boosting technique[33]. Though their primary focus is on using a series of mixture distributions as guide families, adding one mixture component at a time, they do allow an interesting correlation structure within each component, whose covariance matrix is constructed as the sum of a specific form low-rank matrix and a diagonal matrix. Though unlike our approach, this does not use the model itself to define the covariance, it does provide an interesting middle ground between purely diagonal mean-field approaches and full-rank approaches. They show encouraging results for this kind of compromise approach.

One way to allow a correlated posterior without explicitly specifying a correlation structure is by using normalizing flows to transform the guide from a simple mean-field structure to something more complex[46]. In particular, Hamiltonian flows, like our Laplace family proposal below, can use the model itself in defining the guide. In very simplified terms, Hamiltonian flows are similar to what one would get if by taking a sample from a simple parametric

guide such as a multivariate normal, then applying one or more steps of a deterministic Hamiltonian MCMC procedure, leaving an effective distribution for the final outcome which asymptotically approached the true posterior. Like our approach here, this effectively uses the model itself to structure the covariance of the fitted posterior estimate. The computational tradeoffs and necessary approximations involved are different, however; in the future, comparing these two approaches would be interesting.

Also worth mentioning are the prior researchers who have used Laplace approximations in the context of variational inference, but not as a full guide family. Wang and Blei[56] use a method they term Laplace Variational Inference, in which the Laplace approximation is used as a means of improving updates to non-Laplace guide families, in order to speed convergence of mean-field VI. This same approach is further pursued by others[60][38][32].

Finally, others have used Laplace approximations for sub-matrices of the guide-family, in a way similar to what we propose below[61][62][32]. Our work extends this idea to a more general context and uses it as a way to construct a full guide rather than just certain key subcomponents.

2.3 Variational Inference with a Laplace Guide Family

2.3.1 The Laplace family

In this section, we introduce a new type of normal guide family that allows us to capture posterior correlations between model parameters $\theta_1, \ldots, \theta_d$ with only $\mathcal{O}(d)$ guide parameters. (In Section 2.3.5, we will be able to reduce the number of parameters even further through amortization.)

The intuition behind our construction is as follows. Recall that for a probability distribution $p(\boldsymbol{\theta})$, the **observed information at $\boldsymbol{\theta}^*$** is the negative of the Hessian of $\log p(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$, evaluated at $\boldsymbol{\theta}^*$:

$$\mathscr{I}_p\left(\boldsymbol{\theta}^*\right) := -H\left[\log p(\boldsymbol{\theta})\right]\Big|_{\boldsymbol{\theta}^*}. \tag{2.10}$$

It is easy to check that if $\boldsymbol{\theta}^*$ is a mode of $p(\boldsymbol{\theta})$, then $\mathscr{I}_p(\boldsymbol{\theta}^*)$ is the precision matrix of the Laplace approximation for $p(\boldsymbol{\theta})$ at $\boldsymbol{\theta}^*$. Importantly, $\mathscr{I}_p(\boldsymbol{\theta}^*)$ can be evaluated even if we only have access to $p(\boldsymbol{\theta})$ up to normalizing constant.

Now suppose we could guarantee that, for our particular model posterior $p(\boldsymbol{\theta}|\boldsymbol{x})$, the matrix $\mathscr{I}_p(\boldsymbol{\theta}^*)$ was positive definite for all $\boldsymbol{\theta}^*$. Then there would be a very natural way to define a d-dimensional guide family for $p(\boldsymbol{\theta}|\boldsymbol{x})$: use the family of multivariate normal distributions $\{q_{\boldsymbol{\theta}^*} : \boldsymbol{\theta}^* \in \mathbb{R}^d\}$, where $q_{\boldsymbol{\theta}^*}$ has mean $\boldsymbol{\theta}^*$ and precision matrix $\mathscr{I}_p(\boldsymbol{\theta}^*)$. If the mode of $p(\boldsymbol{\theta}|\boldsymbol{x})$ (which, in this special case, would necessarily be unique) happened to maximize the ELBO, then the fitted guide $q_{\hat{\boldsymbol{\theta}^*}}$ would be the Laplace approximation of $p(\boldsymbol{\theta}|\boldsymbol{x})$ at this mode. More likely, because of asymmetries in the posterior, the ELBO will be higher if $\boldsymbol{\theta}^*$ is a slight perturbation of the mode of p; but in any case, we know the optimal guide is *at least as good* as the Laplace approximation.

Thus, for each mode $\boldsymbol{\theta}^*$ of $p(\boldsymbol{\theta})$, the Laplace family as defined below will contain distributions arbitrarily close to the Laplace approximation of $p(\boldsymbol{\theta})$ at $\boldsymbol{\theta}^*$. In many cases, we expect our final fitted guide $q_{\boldsymbol{\theta}^*,\psi}$ to be close to one of these distributions, but this will not necessarily be the case. For one thing, the Laplace approximation is the best-fit Gaussian only *locally* and is not necessarily the best approximation to $p(\boldsymbol{\theta})$ in the sense of KL-divergence. For instance, the mean $\boldsymbol{\theta}^*$ of our fitted guide might not be a mode of p at all. For instance, as we will see in the next section, a Gaussian centered at a saddle point between two modes of $p(\boldsymbol{\theta})$ may give a higher ELBO than the Laplace approximation at either mode. In other words, by using the Laplace guide family, we are essentially guaranteeing that an optimally-fitted guide will be *at least* as good as a Laplace approximation at one of the modes of p.[5] But in many cases, it will be even better.

In general, of course, $\mathscr{I}_p(\boldsymbol{\theta}^*)$ will not always be positive definite. But suppose we could find a function $f : S_d \to S_d$, acting on the set symmetric $d \times d$ matrices, such that

[5]It is important to note that maximizing the ELBO will lead us not to the highest mode of p, but to the mode that has the most total density around it. If we think of modes as "hills" in the distribution's density, a shorter but wider hill may have more volume than one that's tall but skinny.

- if $M \in S_d$ is positive definite with a determinant greater than some small constant, then $f(M) = M$;
- Otherwise $f(M)$ is a positive definite matrix close to M (in some norm).

We refer to such functions as **boosting functions**; these have been extensively studied in the context of non-linear optimization. In practice, computing $f(M)$ usually involves some modification of the Cholesky decomposition algorithm for positive definite matrices. See [14] for a survey of commonly used boosting methods.

Using a boosting function f, we could modify the above construction by letting $q_{\boldsymbol{\theta}^*}$ have precision matrix $f(\mathscr{I}_p(\boldsymbol{\theta}^*))$. When $\boldsymbol{\theta}^*$ is at or near a mode of p, $\mathscr{I}_p(\boldsymbol{\theta}^*)$ would already be positive definite and f would not cause any distortion; thus the Laplace approximation at every mode of p would still be available as a guide. And, once again, if we find that the ELBO is actually maximized at an entirely different value of $\boldsymbol{\theta}^*$, so much the better.

For additional flexibility, we will use not a single function f, but a parametrized family of such functions:

Definition: *Let S_d be the set of symmetric $d \times d$ matrices and let Ψ be a subset of $\mathbb{R}^d_+$ containing $\vec{\mathbf{0}}$ in its closure. A* **boosting family** *f_Ψ is a family of almost everywhere thrice-differentiable functions $f_{\boldsymbol{\psi}} : S_d \to S_d$, indexed by Ψ, such that:*

- *For any $\boldsymbol{\psi} \in \Psi$ and any $M \in S_d$, the matrix $f_{\boldsymbol{\psi}}(M)$ is positive definite.*
- *If M itself is positive definite then*

$$\lim_{\boldsymbol{\psi} \to \vec{\mathbf{0}}} f_{\boldsymbol{\psi}}(M) = M. \tag{2.11}$$

We are now ready to define the Laplace guide family for a probability distribution:

Definition: *Let $p(\boldsymbol{\theta})$ be a (possibly unnormalized) probability distribution on $\mathbb{R}^d$. Let $\Theta \subseteq \mathbb{R}^d$, $\Psi \subseteq \mathbb{R}^d_+$, and let f_Ψ be a boosting family as above.*

The **Laplace guide family** $\mathcal{L}_{\Theta\times\Psi}(p, f_\Psi)$ *is the set of* d*-dimensional normal distributions* $\{q_{\boldsymbol{\theta},\boldsymbol{\psi}} : \boldsymbol{\theta} \in \Theta,\ \boldsymbol{\psi} \in \Psi\}$*, where* $q_{\boldsymbol{\theta},\boldsymbol{\psi}}$ *has mean* $\boldsymbol{\theta}$ *and precision matrix* $f_{\boldsymbol{\psi}}\left(\mathscr{I}_p(\boldsymbol{\theta})\right)$*.*

Note that the space $\Theta \times \Psi$ of guide parameters can have up to $2D$ dimensions. However, as we will see below, when working with latent variable models, we often constrain Θ and Ψ to lower-dimensional subsets of $\mathbb{R}^d$ and $\mathbb{R}^d_+$ respectively.

While we have defined a Laplace guide family as a Gaussian over all model parameters, it is of course possible to keep certain parameters out of the multivariate normal and deal with them in other ways. For example, in this and the following chapter, there are two cases where we will take a model parameter ξ as having a delta distribution in the guide conditional on the guide parameters. In such cases, the value of ξ^* is still used to determine the point at which we take the Hessian, but the Hessian itself does not include ξ. The guide is then defined as the product of a multivariate normal over the other model parameters, with a delta distribution for $\xi = \xi^*$.

There are several reasons we might decide to do this:

- There may be some difficulty obtaining the Hessian with respect to ξ, for instance because some distribution involving ξ is not thrice-differentiable.
- It may be clear from the problem setup that the combined posterior including ξ will not be well-approximated by a multivariate normal. For instance, if ξ might be a scale parameter controlling the standard deviation of some other dimension(s)/parameter(s) in the posterior.
- ξ may simply be a nuisance parameter whose posterior variance is thought to be unlikely to contribute substantially to the posterior variance of other more-important parameters. In this case, using a delta distribution would merely be a time-saving trick to avoid the need to include ξ in the Hessian.

2.3.2 A toy example

Before proceeding to high-dimensional latent variable models, we illustrate the advantage of a Laplace family over a mean-field family using a simple, low-dimensional example.

Suppose we want to model an observable quantity x as the sum of two t-distributed random variables, plus a normally-distributed error term ϵ:

$$\begin{aligned} x &= T_1 + T_2 + \epsilon \\ T_i &\sim StudentT_\nu; i \in \{1, 2\} \\ \epsilon &\sim \mathcal{N}(0, \sigma^2) \end{aligned} \tag{2.12}$$

The quantities of interest are the of model parameters $\boldsymbol{\theta} := (T_1, T_2)$; or, to be precise, their posterior distribution conditional on the observed x and the known ν (degrees of freedom) and σ^2 (observation error variance). Note that, if x_{obs} is sufficiently far from 0 and ν is sufficiently low, then $p(\boldsymbol{\theta}|x_{\text{obs}})$ is bimodal: it is more likely that one of the T_i accounts for most of x_{obs} than that they each account for roughly half.

We approximate $p(\boldsymbol{\theta}|x_{\text{obs}})$ using variational inference with two different guide families:

- *The Laplace guide family* $\mathcal{L}_{\Theta\times\Psi}(p, f)$. Here $\Theta = \mathbb{R}^2$. For simplicity, we take Ψ to be one-dimensional:

$$\Psi := \left\{(\psi_1, \psi_2) \in \mathbb{R}^2 : \psi_1 = \psi_2\right\}. \tag{2.13}$$

  The guide $q_{\boldsymbol{\theta}^*,\boldsymbol{\psi}}(\boldsymbol{\theta})$ is normal with mean $\boldsymbol{\theta}^* = (T_1^*, T_2^*)$ and precision matrix $\mathcal{P}_{\boldsymbol{\theta}^*} := f_{\boldsymbol{\psi}}\left[\mathscr{I}_p\left(\boldsymbol{\theta}^*\right)\right]$.

- *The normal mean-field guide family* $\mathcal{F}_\Phi$. Here $\Phi = \mathbb{R}^2 \times \mathbb{R}_+^2$. With

$$\boldsymbol{\phi} = (T_1^*, T_2^*, \sigma_1, \sigma_2) \in \Phi, \tag{2.14}$$

the mean-field guide $q_{\phi}(\boldsymbol{\theta})$ is normal with mean (T_1^*, T_2^*) and covariance matrix $\text{Diag}(\sigma_1^2, \sigma_2^2)$.

For each family, we fit the guide to the model using variational inference as implemented in the Python package "Pyro"[8]. We then compare the fitted guides using two metrics:

- The ELBO with respect to the fitted guide: this is the quantity that variational inference is trying to maximize. Recall that *maximizing* the ELBO is equivalent to minimizing the KL-divergence *from the guide to the posterior*.
- The EUBO with respect to the fitted guide (see equation 2.8). Recall that *minimizing* the EUBO is equivalent to minimizing the KL-divergence *from the posterior to the guide*. (This makes the EUBO a more natural measure of fit than the ELBO.) In general, the integral in the EUBO is intractable, but in this simple example we can get good numerical estimates by taking a set of MCMC samples and assuming that they are a consistent approximation of samples from the true posterior.

Table 2.3.1 shows the results of the comparison for $\sigma = 0.4$ and several different values of x_{obs} and ν. In all cases, the Laplace family gives a better approximation than the mean-field family for both of our metrics.

The difference in the EUBO for $x_{\text{obs}} = 7$ and $\nu = 2$ is particularly striking. (As the table shows, such an observation is in the 2% upper tail; an outlier, but not an extreme one.) Figure 2.3.1 helps clarify the situation. When we plot credible regions for the true posterior (green), the Laplace fitted guide (red), and the mean-field fitted guide (blue), we see that the true posterior in this case is bimodal. The mean-field guide is centered at one of the modes and is thus essentially missing half the mass of the posterior distribution; this explains why $D_{\text{KL}}(p||q)$, and thus the EUBO, is so large. In contrast, the Laplace guide is centered at the saddle point between the two modes of the posterior. This is a

Table 2.3.1: A comparison of variational inference for the model in 2.12 using Laplace ("Lap") and mean-field ("MF") families. For ELBO columns, higher is better; for EUBO columns, lower is better.

x_{obs}	ν	σ	$P(\lvert x\rvert < x_{\text{obs}})$	ELBO		EUBO	
			a priori	Lap	M-F	Lap	M-F
0.0	30	0.4	0.00	-1.45	-2.29	.05	0.78
3.0	30	0.4	0.96	-3.43	-4.37	-1.41	-1.03
0.0	2	0.4	0.00	-1.65	-2.52	.08	.91
3.0	2	0.4	0.81	-3.10	-4.55	-0.84	3.26
7.0	2	0.4	0.96	-5.39	-7.23	-2.01	54.64

case where the the flexibility provided by the additional parameter ψ plays a significant role in improving the quality of the VI estimate.[6]

We can also use this example to illustrate the distinction between Laplace variational inference and a simple Laplace approximation to the posterior around its maximum density (MAP). Note that if σ were not taken as known in this model, but instead given a prior (such as a half-Cauchy distribution), the posterior density of $\sigma, \epsilon \to 0$ would be infinite. The ELBO, however, would not be, because as the energy term approaches infinity, so does the entropy term, leaving the overall ELBO with a lackluster finite value. This is an important way in which maximizing the ELBO results in better inference than maximizing the posterior density.

2.3.3 The Laplace guide family for a latent variable model

In this section, we show why Laplace guide families are particularly well-suited for working with latent variable models. In particular, we show that latent

[6]For this example, we use an alternate quasi-boosting family, based on using logsumexp as a softmax to ensure the matrix is SDD ([symmetric and] diagonally dominant). We use this rather than the boosting family based on GMW81 because the latter is not smooth with respect to ψ. If this model were more than a toy, we would have chosen one of the other, more-complex methods outlined in [14] rather than this SDD-based method.

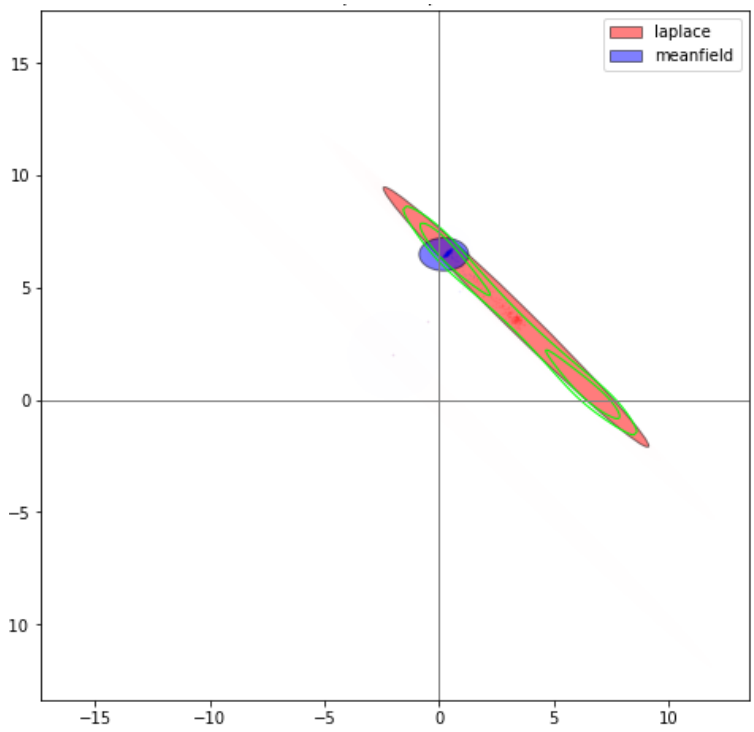


Figure 2.3.1: Posterior and VI-estimated posterior for $x_{\text{obs}} = 7, \nu = 2$. Green lines show (approximate) 95% and 50% true posterior credible sets; red and blue ellipses are estimated 95% credible sets for the fitted Laplace and mean-field guides respectively.

variable models have a block arrowhead structure on the Hessian, which allows substantially faster computation in optimizing to find the best-fit guide.

As usual, we assume the model has parameters $\boldsymbol{\theta} = \{\boldsymbol{\gamma}, \boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_N)$, where $\boldsymbol{\gamma} \in \mathbb{R}^g$ is the vector of global parameters and $\boldsymbol{\lambda}_i \in \mathbb{R}^l$ is the vector of latent parameters corresponding to the observation $\boldsymbol{x}_i$. The Laplace family has one guide parameter corresponding to each model parameter, and we split up the corresponding vector of guide parameters, $\boldsymbol{\theta}^*$, accordingly:

$$\boldsymbol{\theta}^* = (\boldsymbol{\gamma}^*, \boldsymbol{\lambda}_1^*, \ldots, \boldsymbol{\lambda}_N^*). \tag{2.15}$$

We refer to $\boldsymbol{\gamma}^*$ as the vector of *global guide parameters* and to $\boldsymbol{\lambda}_1^*, \ldots, \boldsymbol{\lambda}_N^*$ as the

latent guide parameters.

Because the $\boldsymbol{\lambda}_i$ are conditionally independent in the model, the matrix $\mathscr{I}_p(\boldsymbol{\theta}^*)$ has a block-arrowhead structure:

$$\mathscr{I}_p(\boldsymbol{\theta}^*) = \begin{pmatrix} G & C_1 & C_2 & \dots & C_N \\ C_1^T & U_1 & 0 & \dots & 0 \\ C_2^T & 0 & U_2 & \dots & 0 \\ \vdots & 0 & 0 & \ddots & 0 \\ C_N^T & 0 & 0 & \dots & U_N \end{pmatrix}, \tag{2.16}$$

This allows us to speed up the computation in several ways. First, for fixed g and l, computing $\mathscr{I}_p(\boldsymbol{\theta}^*)$ takes only $\mathcal{O}(N)$ time, rather than the default $\mathcal{O}(N^2)$. We can design a boosting family f_Ψ for block-arrowhead matrices such that $f_{\boldsymbol{\psi}}(\mathscr{I}_p(\boldsymbol{\theta}^*))$ is also block-arrowhead and computing it takes $O(N)$ time. As a result, sampling from $q_{\boldsymbol{\theta}^*,\boldsymbol{\psi}}$ is also only $\mathcal{O}(N)$ rather than $\mathcal{O}(N^3)$. See Appendix 2.1 for details and proofs.

We can further speed up the algorithm by setting the components of $\boldsymbol{\psi}$ corresponding to the different latent parameters to be equal. This reduces the dimensionality of the boosting parameter space Ψ from $g + lN$ to $g + l$. From now on, when working with latent variable models we will always assume that

$$\Psi = \{(\boldsymbol{\psi}_{\boldsymbol{\Gamma}}, \boldsymbol{\psi}_{\boldsymbol{\Lambda}}, \dots, \boldsymbol{\psi}_{\boldsymbol{\Lambda}}) : \boldsymbol{\psi}_{\boldsymbol{\Gamma}} \in \mathbb{R}_+^G,\ \boldsymbol{\psi}_{\boldsymbol{\Lambda}} \in \mathbb{R}_+^L\}, \tag{2.17}$$

and we will abbreviate $\boldsymbol{\psi} \in \Psi$ as $(\boldsymbol{\psi}_{\boldsymbol{\Gamma}}, \boldsymbol{\psi}_{\boldsymbol{\Lambda}})$.

2.3.4 Stochastic variational inference with a Laplace family

Stochastic variational inference (SVI) is a standard method for speeding up variational inference on high-dimensional latent variable models. It was first introduced in [22] for exponential family distributions, but is easily generalized to black-box VI.

In SVI, each iteration of the ELBO-optimization procedure uses only a

random (possibly weighted) subsample of the observations and their corresponding latent parameters. Because the $\boldsymbol{\lambda}_i$ are independent conditional on $\boldsymbol{\gamma}$, it is easy to get an unbiased estimate of the unnormalized posterior in this way, evaluating only the terms involving the subsampled units. We can also estimate the ELBO and its gradient, though these estimates will not necessarily be unbiased (see below). We then use a specially-adapted optimization algorithm, which takes the noise in the gradient estimate into account. A number of such stochastic optimization algorithms are available; *Pyro*, the probabilistic programming language that we use for our computations, uses the Adam algorithm [28] for this purpose.

Implementing SVI for the Laplace family is conceptually straightforward, but there are a few details that require attention. Recall that our vector of variational parameters is $(\boldsymbol{\theta}^*, \boldsymbol{\psi})$, and that $\boldsymbol{\theta}^*$ consists of the vector $\boldsymbol{\gamma}^*$ of global guide parameters and N vectors $\boldsymbol{\lambda}_1^*, \ldots, \boldsymbol{\lambda}_N^*$ of latent guide parameters. When N is large, we want to be able to estimate the ELBO using only $\boldsymbol{\gamma}^*$ and a subsample of the $\boldsymbol{\lambda}_i^*$ of size $n \ll N$.

Fix n, and let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N)$ be a vector of nonzero probabilities $(0 < \pi_i \leq 1)$ such that $\sum_{i=1}^N \pi_i = n$. Let $\mathcal{S} = \{i_1, \ldots, i_n\}$ be a sample from $\{1, \ldots, N\}$, drawn in such a way that $\text{Prob}(i \in \mathcal{S}) = \pi_i$. For instance, for a simple random sample (without replacement), we would have $\pi_i = n/N$ for all i.[7]

Given $\boldsymbol{\theta} = (\boldsymbol{\gamma}, \boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_N)$ and $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$, define

$$\begin{aligned} \boldsymbol{\theta}_{\mathcal{S}} &= (\boldsymbol{\gamma}, \boldsymbol{\lambda}_{i_1}, \ldots, \boldsymbol{\lambda}_{i_n}); \\ \boldsymbol{x}_{\mathcal{S}} &= (\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_n}); \\ p_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{S}}, \boldsymbol{x}_{\mathcal{S}}) &:= p(\boldsymbol{\gamma}) \prod_{i \in \mathcal{S}} \Big[p(\boldsymbol{\lambda}_i|\boldsymbol{\gamma})\ p(\boldsymbol{x}_i|\boldsymbol{\lambda}_i, \boldsymbol{\gamma}) \Big]^{1/\pi_i}. \end{aligned}$$

[7] Here we are assuming sampling without replacement and with a fixed sample-size n, which allows us to use the Horvitz-Thompson estimator. However, any other sampling scheme that has a corresponding unbiased estimator would work. The goal, as always, is to find an unbiased estimator with low variance. How best to do this depends on the situation.

Note that $\log p_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{S}}, \boldsymbol{x}_{\mathcal{S}})$ is a Horvitz-Thompson estimator for $\log p(\boldsymbol{\theta}, \boldsymbol{x})$.

Now let $\boldsymbol{\theta}^* \in \Theta$, and define $\boldsymbol{\theta}^*_{\mathcal{S}}$ analogously to $\boldsymbol{\theta}_{\mathcal{S}}$. The Laplace guide $q_{\boldsymbol{\theta}^*_{\mathcal{S}},\boldsymbol{\psi}}(\boldsymbol{\theta}_{\mathcal{S}})$ for $p_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{S}}|\boldsymbol{x}_{\mathcal{S}})$ is defined in the usual way: it a multivariate normal distribution with mean $\boldsymbol{\theta}^*_{\mathcal{S}}$ and precision matrix $f_{\boldsymbol{\psi}}\left(\mathscr{I}_{p_s}(\boldsymbol{\theta}^*_{\mathcal{S}})\right)$. We can do stochastic gradient ascent on the ELBO of $p_{\mathcal{S}}$ with respect to $q_{\boldsymbol{\theta}^*_{\mathcal{S}},\boldsymbol{\psi}}(\boldsymbol{\theta}_{\mathcal{S}})$, taking a new subsample $\mathcal{S}$ at each step of the gradient ascent. Once we obtain (or get close to) the optimal values $\boldsymbol{\theta}^*$ and $\boldsymbol{\psi}$ for this subsampling procedure, we recompute the full precision matrix (without subsampling) to obtain the final fitted guide $q_{\boldsymbol{\theta}^*,\boldsymbol{\psi}}$.

Unfortunately, the ELBO of $p_{\mathcal{S}}$ is *not* an unbiased estimate of the ELBO of p, so we do not expect the stochastic and non-stochastic versions of Laplace VI to converge to the same guide. However, the fitted guide $q_{\boldsymbol{\theta}^*,\boldsymbol{\psi}}$ obtained by SVI is, in expectation, a good approximation to $p_{\mathcal{S}}$, and thus should also be a good approximation to the true posterior as well.

2.3.5 Analytic amortization

In a model with a large number of latent variables $\boldsymbol{\lambda}_i$ — each of which requires a separate vector of guide parameters $\boldsymbol{\lambda}^*_i$ — the dimensionality of the guide family can become too high for even stochastic optimization. Using the algorithm described in the previous section, even after the global parameters $\boldsymbol{\gamma}^*$ converge to "good" values (i.e, ones that tend to roughly maximize the ELBO), we must still ensure that for each unit i, the corresponding $\boldsymbol{\lambda}^*_i \in \mathcal{S}$ for enough optimization-step-specific values of $\mathcal{S}$, to allow the optimized $\boldsymbol{\lambda}^*_i$ to converge to a "good" value as well. This process of "tying up loose ends" with the latent guide parameters could easily take more computing time than optimizing the globals.

One common approach is to constrain each $\boldsymbol{\lambda}^*_i$ to be a deterministic function $M_i(\boldsymbol{\gamma}^*)$. In many applications, the function M comes from a neural network and thus has its own free parameters (weights). The gradient of the ELBO is then computed with respect to these weights as well as $\boldsymbol{\gamma}^*$, and the weights are optimized as part of VI. This technique of reducing the number of guide

parameters to be optimized is known as **amortization**. In the context of neural networks, the function M is referred to as a **variational auto-encoder**; see [29] for details.

Since we are assuming that our model posterior has a relatively simple functional form, we take a different approach to amortization: we analytically derive (or approximate) the MAP of $\boldsymbol{\lambda}_i^*$ conditional on $\boldsymbol{\gamma}^*$ and $\boldsymbol{x}_i$, and then simply set $\boldsymbol{\lambda}_i^*$ to this value. In other words, we use a deterministic function

$$M_i(\boldsymbol{\gamma}^*) \approx \text{MAP}\left(\boldsymbol{\lambda}_i^* | \boldsymbol{\gamma}^*, \boldsymbol{x}_i\right) \tag{2.18}$$

that has no additional parameters to be optimized.[8]

Of course, this kind of **analytic amortization** is impossible when the relevant conditional distributions are a consequence of more complicated dynamics, such as neural nets or other forms of machine learning. But where possible, we believe that analytic amortization is both simpler and more efficient than the traditional kind. In this chapter, we use it with a Laplace guide family, but we see no reason why it could not be used with other guide families as well.

With analytic amortization, the Laplace guide family is defined as usual, but the guide parameter $\boldsymbol{\theta}^*$ is restricted to

$$\Theta_M = \left\{(\boldsymbol{\gamma}^*, \boldsymbol{\lambda}_1^*, \ldots, \boldsymbol{\lambda}_N^*) \in \Theta : \boldsymbol{\lambda}_i^* = M_i(\boldsymbol{\gamma}^*) \text{ for } i = 1, \ldots, N\right\}. \tag{2.19}$$

Thus the ELBO is now a function only of $\boldsymbol{\gamma}^*$ and $\boldsymbol{\psi}$.

It's worth noting one additional computational trick that should improve results in some cases. Sometimes, the analytic amortization is approximate; the exact conditional MAP of $\boldsymbol{\lambda}_i^*$ is not analytically tractable, so the best $M_i(\boldsymbol{\gamma}^*)$ practically available is only an approximation thereof. When this is true, a slight adjustment to the guide $q_{\boldsymbol{\theta}^*,\boldsymbol{\psi}}$ can improve its fit at almost no extra

[8] Note that using the conditional MAP as suggested here is generally appropriate when $\boldsymbol{\lambda}_i$ consists of solely location parameters. However, when $\boldsymbol{\lambda}_i$ includes a combination of location and scale parameters, the MAP is often 0 for the scale parameters regardless of the values of $\boldsymbol{\gamma}$; since this leads to an ELBO of $-\infty$ through the entropy term this is undesirable. Resolving this issue is beyond the scope of the current paper.

computational cost. Recall that $q_{\boldsymbol{\theta}^*,\psi}$ is normal with mean $\boldsymbol{\theta}^*$ and precision matrix $\mathcal{P} = f_\psi(\mathscr{I}_p(\boldsymbol{\theta}^*))$. Calculating $\mathcal{P}$ is the most computationally-intensive part of the algorithm. However, once we know $\mathcal{P}$, we have all the ingredients that we need in order to apply a step of Newton's method to our estimate of $\text{MAP}(\boldsymbol{\lambda}_i^*|\boldsymbol{\gamma}^*, \boldsymbol{x}_i)$. After all, $\mathcal{P}$ is the boosted negative Hessian of $\log p$ at $\boldsymbol{\theta}^*$, and in the course of computing $\mathcal{P}$, we had to compute $\nabla \log p$ as well. So, for each i, we can let

$$\tilde{\boldsymbol{\lambda}}_i^* := \boldsymbol{\lambda}_i^* + (\mathcal{P}_i)^{-1} \nabla \log p(\boldsymbol{\theta}^*, \boldsymbol{x})_i, \tag{2.20}$$

where $\mathcal{P}_i$ and $\nabla \log p(\boldsymbol{\theta}^*, \boldsymbol{x})_i$ denote the submatrix (respectively, subvector) corresponding to $\boldsymbol{\lambda}_i$. We can now let $\tilde{q}_{\boldsymbol{\theta}^*,\psi}$ be the normal distribution with precision $\mathcal{P}$ and mean $(\boldsymbol{\gamma}^*, \tilde{\boldsymbol{\lambda}}_1^*, \ldots, \tilde{\boldsymbol{\lambda}}_N^*)$. Note that $\tilde{q}_{\boldsymbol{\theta}^*,\psi}$ is no longer technically a Laplace guide, since its precision matrix is based on a Hessian taken at a different point than its mean. However, Newton's method can be expected to improve the mean of the guide, giving a direct (first-order) improvement in the energy term of the ELBO, while the slight error this creates in the Hessian should be a more indirect (second-order) effect.

2.3.6 Full algorithm for amortized Laplace SVI

Putting it all together, below is the full algorithm for amortized, subsampled Laplace variational inference in a latent variable model.

Given:

- a latent variable model, with notation as in Section 2.3.3;
- an integer $n < N$ (desired sample size in SVI) and a vector $\boldsymbol{\pi}$ of sampling probabilities[9], as in Section 2.3.4;
- a family f_ψ of boosting functions for arrowhead matrices, as in Appendix 2.1;

[9] In all the examples, we have used a simple random sample with equal probabilities. In some cases, this might be improved by tuning the probability weights π so as to minimize the variance of the ELBO estimate. Discussion of when and how to do this is beyond the scope of this chapter.

- a family of amortization functions $M_i : \mathbf{\Gamma} \to \mathbf{\Lambda}$ for $i \in \{1, \ldots, N\}$, as in Section 2.3.5;

Algorithm for fitting the guide:

Initialize $\boldsymbol{\gamma}^*$ to 0; $\boldsymbol{\psi}_{\mathbf{\Gamma}}$ and $\boldsymbol{\psi}_{\mathbf{\Lambda}}$ to vectors of small positive numbers (e.g. 0.01); and m to a positive integer;

while True **do**

  choose an independent sample $\mathcal{S} := \{i_1, \ldots, i_n\}$ from $\{1, \ldots, N\}$, with sampling probabilities given by $\boldsymbol{\pi}$;

  set $\boldsymbol{x}_{\mathcal{S}} := (\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_n})$;

  for $k \in 1, \ldots, n$ **do**

    set $\boldsymbol{\lambda}^*_{i_k} := M_{i_k}(\boldsymbol{\gamma}^*)$;

  set $\boldsymbol{\theta}^*_{\mathcal{S}} := \left(\boldsymbol{\gamma}^*, \boldsymbol{\lambda}^*_{i_1}, \ldots, \boldsymbol{\lambda}^*_{i_n}\right)$;

  compute $\mathscr{I}(\boldsymbol{\theta}^*_{\mathcal{S}})$: the negative Hessian[10] of the sampled model log density

$$\log p_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{S}}, \boldsymbol{x}_{\mathcal{S}}) := \log p(\boldsymbol{\gamma}) + \sum_{k=1}^{n} \frac{1}{\pi_{i_k}} \log p(\boldsymbol{\lambda}_{i_k}, x_{i_k} | \boldsymbol{\gamma}) \quad (2.21)$$

  with respect to the variables $\boldsymbol{\theta}_{\mathcal{S}} := (\boldsymbol{\gamma}, \boldsymbol{\lambda}_{i_1}, \ldots, \boldsymbol{\lambda}_{i_n})$, evaluated at $(\boldsymbol{\theta}^*_{\mathcal{S}}, \boldsymbol{x}_{\mathcal{S}})$;

  compute $\mathcal{P}_{\mathcal{S}} := f_{\psi,\pi}(\mathscr{I}(\boldsymbol{\theta}^*_{\mathcal{S}}))$;

  (OPTIONAL) set $\boldsymbol{\theta}^*_{\mathcal{S}} := \left(\boldsymbol{\gamma}^*, \tilde{\boldsymbol{\lambda}}^*_{i_1}, \ldots, \tilde{\boldsymbol{\lambda}}^*_{i_n}\right)$, where

$$\tilde{\boldsymbol{\lambda}}^*_i := \boldsymbol{\lambda}^*_i + (\mathcal{P}_i)^{-1} \nabla \log p(\boldsymbol{\theta}^*, \boldsymbol{x})_i, \quad (2.22)$$

[10] When using automatic differentiation, maintaining the distinction between $\boldsymbol{\theta}_{\mathcal{S}}$ and $\boldsymbol{\theta}^*_{\mathcal{S}}$ requires some care. In $\boldsymbol{\theta}_{\mathcal{S}}$, the $\boldsymbol{\lambda}$s are not related to $\boldsymbol{\gamma}$ via the amortization function M, so the gradient of M must be excluded from the calculation of the Hessian; yet when we estimate the gradient of the ELBO with respect to $\boldsymbol{\gamma}^*$, we do want to include M in calculating the gradient of the point at which the Hessian was taken. To accomplish this programmatically in pytorch, we make a copy of $\boldsymbol{\gamma}^*$ and use the `detach` command to sever its connection to the $\boldsymbol{\lambda}$s via M. We use the detached copy when computing the Hessian and add its gradient onto that of the original variable $\boldsymbol{\gamma}^*$ just before ADAM optimization.

as in Section 2.3.5;

11: draw m i.i.d samples $\boldsymbol{\theta}_{\mathcal{S}}^1, \ldots, \boldsymbol{\theta}_{\mathcal{S}}^m$ from the normal distribution with mean $\boldsymbol{\theta}_{\mathcal{S}}^*$ and precision matrix $\mathcal{P}_{\mathcal{S}}$;[11]

12: set

$$\text{ELBO}_{\text{est}} := \frac{1}{m} \sum_{j=1}^{m} \left[\log p_{\mathcal{S}}(\boldsymbol{\theta}_{\mathcal{S}}^j, \boldsymbol{x}_S) - (\boldsymbol{\theta}_{\mathcal{S}}^j - \boldsymbol{\theta}_{\mathcal{S}}^*)^T \mathcal{P}_{\mathcal{S}} (\boldsymbol{\theta}_{\mathcal{S}}^j - \boldsymbol{\theta}_{\mathcal{S}}^*) \right]; \tag{2.23}$$

13: use backwards-mode automatic differentiation to find the gradient of ELBO_{est};[12]

14: update $\boldsymbol{\gamma}^*$ and $\boldsymbol{\psi}$ accordingly, using the Adam stochastic optimization algorithm;

15: if a stopping condition has been met, **break** out of the loop;[13]

16: redo steps 4-10 with a full sample (all units); that is, $\mathcal{S} = \{1, \ldots, N\}$ and $\pi = (1, 1, \ldots, 1)$, omitting the subscript $\mathcal{S}$ from all quantities.

17: **return** fitted guide $q_{\boldsymbol{\theta}^*, \boldsymbol{\psi}}$: a normal distribution with mean $\boldsymbol{\theta}^*$ and precision matrix $\mathcal{P}$.

2.4 A simple application

We compare the performance of three variational inference methods — mean-field VI, Laplace VI, and amortized Laplace VI — on a relatively simple latent variable model, using both real and simulated data.

[11] Note that we can perform the sampling without fully inverting $\mathcal{P}_{\mathcal{S}}$; see Appendix 2.1.

[12] This is automated by pyro, applying state-of-the-art variance-reducing tricks such as [47].

[13] The specific stopping rule for the optimization loop is not our focus here. In practice, we use an exponential moving average of ELBO_{est}, with a decay time of 100 epochs; and we stop when that average is not lower than its value 500 epochs ago.

2.4.1 The model

Consider a multi-site experimental study with randomly-assigned treatment and control groups at each site. Let x_i be the difference in means between the treatment and control groups at site i. We can think of x_i as an estimate of the true treatment effect τ_i for site i. For moderately sized groups, we can take the variance of x_i to be s_i, the estimated standard error of $E[(x_i - \tau_i)^2]$ at each site (a known quantity). If we then assume that the true treatment effects are distributed according to a scaled, shifted Student t-distribution with unknown mean μ, scale σ, and degrees of freedom ν, we obtain the following model:

$$\begin{aligned} T_i/\sigma &\sim StudentT_\nu; \quad i \in \{1, ..., N\} \\ \tau_i &= \mu + T_i \\ x_i &\sim \mathcal{N}(\tau_i, s_i^2) \end{aligned} \tag{2.24}$$

The site-level quantities of interest are the true treatment effects τ_i. The scale parameter σ captures the scale of the variability of τ_i, while a low ν indicates that outliers are relatively prevalent.

The Student t-distribution may not be the true distribution of site-level effects. We use it here because its degree-of-freedom parameter ν allows us to explicitly model the prevalence of outliers. Clearly, we will want the expected posterior variance of the τ_i's to (nearly) match the sample variance of the x_i's adjusted by s_i. But this can be accomplished either by setting both ν and σ to be high (corresponding to a broad cluster of x_i's with few outliers) or by setting both to be low (corresponding to a smaller cluster, but with outliers). Lower values of σ and ν would lead to higher estimates of the percentage of sites where the treatment effect would (or did) fall above some nontrivial threshold. Thus, the extra flexibility from adding ν to the model can potentially help us answer this scientifically-meaningful question.

We use the following priors for our model parameters (transforming where

necessary to obtain parameters with unconstrained support):

$$\begin{aligned} d &:= \log(\nu - \nu_{\text{min}}) \sim \mathcal{N}(1, 1.5^2) \\ \varsigma &:= \log(\sigma - \sigma_{\text{min}}) \sim \mathcal{N}(0, 2^2) \\ \mu &\sim \mathcal{N}(0, 20) \end{aligned} \tag{2.25}$$

$$\nu_{\text{min}} = 2.5,\ \sigma_{\text{min}} = \max(s_i) * 1.9$$

The constant $\nu_{min} = 2.5$ is chosen arbitrarily to ensure well-behaved overall variance, while the constraint $\sigma_{min} = max(s_i) * 1.9$ is chosen to make that conditional MAP function used in amortization tractable. (See Appendix 2.2 for details. Note that this minimum on the cross-site variation is problematic, especially if it conflicts with our prior beliefs about the relative scale of within-site and cross-site variation. It is required to make our MLE function for analytic amortization numerically well-behaved, but it could be removed if we were willing to use an approximation to the MLE for cases where the conditional likelihood is multimodal.)

2.4.2 The three VI algorithms

We construct three different guide families for the multi-site model with N sites:

- The **mean-field family** has $6 + 2N$ guide parameters (6 local and $2N$ latent), corresponding to the mean and standard deviation for each of μ, ς, d, and $T_1, \ldots, T_N$.
- The **Laplace family** has $7 + N$ guide parameters: 3 global parameters μ^*, ς^*, d^*; N latent parameter $T_1^*, \ldots, T_N^*$; and 4 boosting parameters $\psi_\mu, \psi_\varsigma, \psi_d, \psi_T$.
- The **amortized Laplace family** has only 7 optimizable guide parameters.

We amortize the N latent parameters by setting

$$T_i^* = M_i(\mu^*, \varsigma^*, d^*) := \arg\max_{T_i} p(T_i|\mu^*, \varsigma^*, d^*). \tag{2.26}$$

The formula and derivation for the function M_i can be found in Appendix 2.2.

We use block-arrowhead SDD quasi-boosting (Appendix 2.1) for unamortized Laplace VI.

For all three VI methods:

- We use stochastic variational inference with sample size $n = 100$ and equal sampling probabilities.
- We perform ELBO maximization using stochastic gradient ascent within the "pyro" python package. We use the ADAM stochastic optimization algorithm, with standard parameters (including a learning rate of 0.005 and $(\beta_1, \beta_2) = (0.8, 0.9)$.)
- For estimating the ELBO, we use either $m = 1$ or $m = 3$ samples from the guide.

Once we obtain the final fitted values of all the guide parameters, we go back and calculate the Hessian one final time over *all* the model parameters, using a numerical approximation for the second derivative of the posterior with respect to d. The final fitted guide is thus a multivariate normal, as in regular Laplace VI.

2.4.3 Testing the algorithms on simulated data

We generated two datasets from the multi-site model, as follows:

- For both datasets, we set $N = 400, \mu = 1.0$ and $\sigma = 2.0$.
- For Dataset #1, we set $\nu = 3.0$; for Dataset #2, we set $\nu = 30.0$;

- We used the same vector $\boldsymbol{s} = (s_1 \ldots, s_{400})$ for both datasets; the s_i were sampled independently from Gamma$(4, 8)$, with all values above 1 set to 1;
- For each dataset, we sampled T_i and x_i for $i \in 1, \ldots, 400$ from the model, conditional on μ, σ, ν, and $\boldsymbol{s}$.
- The final dataset in each case consisted of x_i and s_i for $1 \leq i \leq 400$.

Each of the three VI algorithm was run on each dataset with $m = 3$. Repeated runs, as well as runs with $m = 1$, converged to similar values; that is, the stochastic optimization seems robust. For each dataset, we also carried out Hamiltonian MCMC (using the Stan package with NUTS sampler, for 4 chains of 1000 warm-up and 1000 samples each) and used the distribution of MCMC samples as a stand-in for the true posterior.

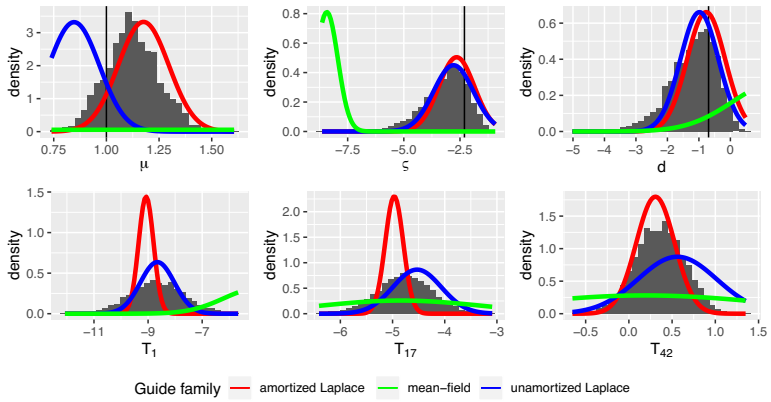
For each algorithm, dataset, and value of m, Table 2.4.1 reports the following metrics:

- The average ELBO and EUBO with respect to the fitted guide;
- The average coverage of T_i for $1 \leq i \leq 400$; that is, the proportion of the MCMC samples that fall inside the symmetric 95% credible interval of the fitted guide. Note that this not the sense of "coverage" usually used in simulation studies; instead of being the percentage of separately-estimated intervals containing the truth, it is the percentage of the "true" (MCMC-based) posterior contained in a single estimated posterior.
- The average coverage of μ, and σ, and ν in the same sense.

The marginal distributions of six model parameters (μ, ς, d, and three arbitrarily chosen latents) for the three fitted VI guides and MCMC are shown in Figure 2.4.2 for Dataset #1 ($\nu = 3$) and in Figure 2.4.1 for Dataset #2 ($\nu = 30$).

Table 2.4.1: Measures of fitted variational inference outcome quality for two simulated scenarios, three guide families, and two numbers of guide samples m. For both scenarios, data were generated with $\mu = 1.0, \sigma = 2.0$.

Dataset	ν	m	Family	Amortized?	EUBO	ELBO	Coverage of 95% interval			
							μ	σ	ν	T_i
1	3	3	Laplace	Y	-357	-1126	0.922	0.813	0.835	0.882
1	3	3	Laplace	N	-224	-1130	0.352	0.928	0.914	0.834
1	3	3	Mean-field	N	75	-3913	1.000	0.124	0.538	0.934
2	30	3	Laplace	Y	-125.97	897	0.912	0.9602	0.593	0.864
2	30	3	Laplace	N	-101.87	953	0.389	0.947	0.722	0.825
2	30	3	Mean-field	N	-183.98	885	1.000	0.000	0.981	.999



In both the $\nu = 3$ (high-outlier) and the $\nu = 30$ (nearly-Gaussian) cases, amortized Laplace is superior to unamortized Laplace, which is (substantially) superior to mean-field. This is visible in terms of lower EUBO, higher ELBO, and better (closer-to-nominal) coverage. Nonetheless, even for amortized Laplace, coverage leaves room for improvement.

2.4.4 Application to ECHS data

We will apply the above model to a multi-site evaluation study of the Early College High School (ECHS) program. Funded by the Bill and Melinda Gates Foundation, this is a program in which high-school students earn an associate degree or up to two years of college credit along with their high-school diploma.

Figure 2.4.1: Marginal distributions of MCMC values and variational fits to Dataset #2, generated using $\mu = 1$, $\sigma = 2$, and $\nu = 30$. In all cases, $m = 3$ guide samples per step were used in estimating the ELBO.

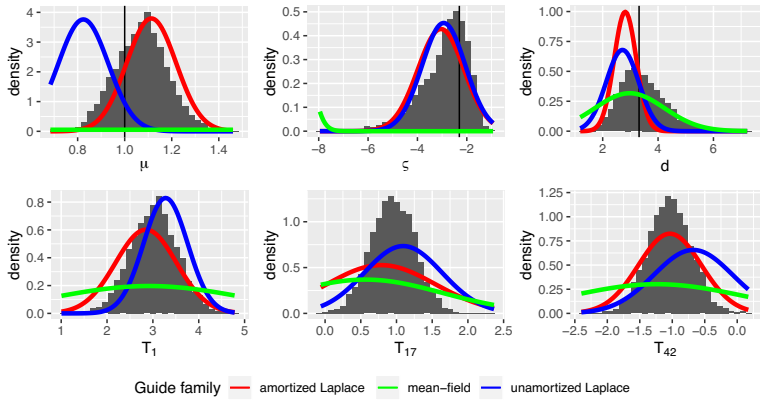


Figure 2.4.2: Marginal distributions of MCMC values and variational fits to Dataset #1, generated using $\mu = 1$, $\sigma = 2$, and $\nu = 3$. In all cases, $m = 3$ guide samples per step were used in estimating the ELBO.

Our data set, from [13], is based on 4,004 North Carolina students who entered one of 44 lotteries and either did or did not qualify for ECHS "treatment." The outcome of interest is a binary indicator of whether a student is "on track" to complete North Carolina's Future-Ready Core Graduation Requirements by the end of ninth grade. In particular, we'd like to understand the distribution of lottery-specific treatment effects.

We follow Yuan, Feller, and Miratrix[58] in terms of data cleaning decisions. In particular, this means that we only consider those students who could be linked to the North Carolina Department of Instruction (NCDPI) databank; whose ninth grade school was within 20 miles of their eighth grade school; and for whom full covariate data is available (race, gender, free or reduced-price lunch eligibility, first generation college student status, and eighth grade math and reading scores). This reduces the sample to 3,477 students across 38 lotteries; 2,021 treated and 1,456 untreated.

Using the multi-site model presented above, we fit the data using both MCMC

and variational inference. Since there were only 44 sites, we did not subsample. The marginal results for the global parameters and three arbitrary site-level (latent) parameters are shown in Figure 2.4.3:

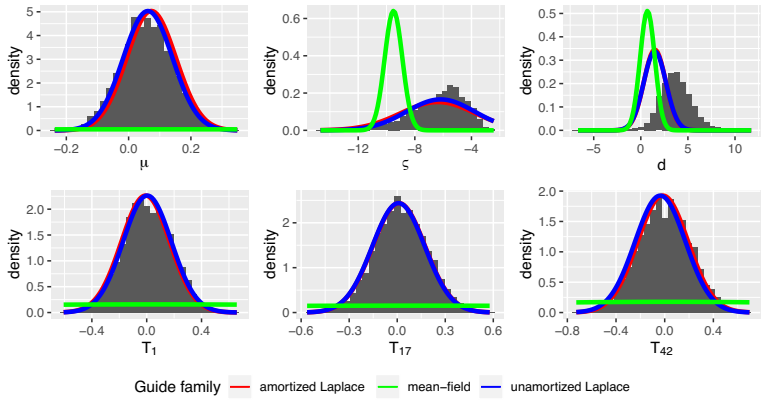


Figure 2.4.3: MCMC values and variational fits to ECHS data.

The (unamortized) mean-field approach seems to have failed badly in this case. As for the (largely similar; and, for d, actually indistinguishable) results of amortized and unamortized laplace, in scientific terms, they seems to suggest that a Gaussian model, without t-distributions to accommodate outliers, would have been sufficient. That is to say, the log-degrees-of-freedom parameter d seems to have its posterior density mode somewhere in the range 2-4, indicating degrees of freedom ν in the range of 10-60 — a distribution of treatment effects that approaches normality.

2.5 Conclusion

In this chapter, we have shown that Laplace families can be a powerful tool for approximate inference. They can capture important dependencies in the posterior better than mean-field guides, without the dimensional overhead of unconstrained Gaussian families. We have given a practical algorithm for using Laplace families with latent variable models, a domain in which existing

techniques (either variational or MCMC) can struggle due to the high dimensionality of the posterior.

Our results on two simple models show the promise of this method, and justify our plan to use this method for the much more comprehensive ecological inference model in Chapter 3.

In order to broaden the practical applications of Laplace variational inference, several directions would be interesting:

- Improve practitioner accessibility. For instance, optimize the code and make a pyro "autoguide" that can create a Laplace family guide automatically.

- Investigate the conditions for efficient convergence of stochastic variational inference to the true optimum member of the guide family. This would include looking for tricks to reduce the variance of the ELBO gradient estimate; finding ways to tune the optimization algorithm, including using the already-calculated Hessian explicitly in the optimizer; and empirically exploring the variability of the results.

- Combine the Laplace family approach with other techniques. Potentially promising combinations include the copula approach of [21][53], to allow the fitted posterior more flexibility than a Gaussian family; non-analytic amortization like the variational autoencoder, to make the process of amortization more of an automated turn-key procedure for the researcher; and perhaps some of the other techniques mentioned in Section 2.2.2 as "prior work".

- Experiment with using multivariate t-distributions rather than multivariate normals as guide families, so that fatter tails might tighten the ELBO-EUBO gap.

3

Ecological Inference

Ecological inference is the notoriously tricky problem of inferring individual behavior from group-level data. Although it comes up in many different domains, the main setting in which ecological inference has been studied during the past few decades is the US Voting Rights Act [1], as interpreted by the Supreme Court in *Thornburg v Gingles* (1986). [2] [19] This is the context that motivates our work in the current paper and from which we draw all our examples.

In *Thornburg v Gingles*, the Court established a set of criteria for determining whether the electoral system in a given jurisdiction violates the voting rights of a racial minority. Some of the criteria depend on the extent to which voting patterns in the jurisdiction are correlated with "race". This raises an ecological inference problem: how can we determine whether voters of different "races" tend to vote differently? Election data tell us how many people in each precinct u voted for each candidate c. However, since the ballot is secret, we have no direct

data on how many of these voters were of "race" r. Vote counts (if we had them) would form an $R \times C$ matrix Y_u (where R is the number of "races" and C is the number of candidates), but, as we'll see in more detail below, the only data we have access to are the row and column sums of Y_u for each precinct u. Our task then is to infer credible values for the elements of these matrices.

The inferential strategies to use in such cases were widely disputed until King's book *A Solution to the Ecological Inference Problem* [26] gave a coherent hierarchical approach for the simple 2×2 case. This was later generalized to a hierarchical Bayesian model for "the $R \times C$ case" (that is, for R and/or C greater than 2) by Rosen, Jiang, King, and Tanner (hereafter RJKT)[48]. King called his approach to the ecological inference problem "EI". Since then, EI has become a broad umbrella term for all approaches that follow in King's footsteps.

The general outline of the EI paradigm is as follows:

1. Give a coherent *a priori* model for voting behavior. The model should include some cross-precinct variability, but also explicitly favor cases where voters of the same group have similar voting patterns across all precincts.

2. Condition this model on the observed data, and estimate and/or draw samples from the Bayesian posterior. Each posterior sample will include both global parameters (such as the fraction of voters of "race" r who would be expected to vote for candidate c) and precinct-level parameters (such as the fraction of the voters of "race" r in precinct u who are inferred to have actually voted for candidate c).

3. Report posterior credible intervals for aggregates of the precinct-level parameters, not for the global parameters. This is because we expect the model to be somewhat wrong; we trust the precinct-level parameters more because they are conditioned on the true data more directly than are the global parameters.

It is this last step which most distinguishes King's EI from the approaches that

preceded it. Note that it is somewhat counter-intuitive. For instance, say we are interested in the percent of African-Americans who voted for candidate c. In the EI approach, even though we have a single parameter which could be interpreted as a prediction of what that number would be if we re-ran the election, we do not simply report a credible interval for that parameter. Instead, for each posterior sample, we add up the number of African-Americans in each precinct who are inferred to have actually voted for candidate c, and report a credible interval for that sum. Though more complex, this approach uses the data we have more fully and efficiently.

RJKT apply this general approach to a specific model of an election with R "races" and C candidates. They derive a fast, moment-based approximation of their model posterior, which can give answers more quickly, though less accurately, than full MCMC. Unfortunately, this derivation depends on the specifics of their model, and thus does not easily generalize to situations where a more complex model might be required, such as when the researcher wishes to use data from multiple elections.

Since RJKT, others have continued to explore different models, mostly following the basic EI paradigm outlined above. Notable examples include many of the articles compiled in [27], [24], [20], [25], and [30].

This chapter has three main goals:

1. Give an easily-extensible framework for building election models that can be used in the EI paradigm.

2. Describe a computationally-tractable approach to approximately sample from such models (variational inference).

3. Demonstrate the feasibility of this approach by fitting a simple $R \times C$ single-election version of the model on simulated data based on the 2016 Presidential election in North Carolina.

In other words, this chapter will not attempt to break new ground in terms of results. Though the model we fit *can* easily be extended in a variety of ways, we do

not do so. We merely show that its performance on existing "solved" problems is comparable to that of existing tools. The task of actually extending this model to new domains is left for future work.

In Section 3.1, we construct a Bayesian model for the most basic version of $R \times C$ ecological inference, which can serve as the basis for more general models. In Section 3.2, we briefly review how variational inference works in the general context of hierarchical Bayesian modeling (including the idea of a **Laplace family guide**, introduced in Chapter 2 of this thesis). In Section 3.3, we describe how to apply these techniques to our basic EI model. In section 3.4, we apply this model to simulated data representing the 2016 presidential election in North Carolina, and give some results comparing this with the RJKT approach. Finally, in Section 3.5, we show how our basic model can be extended in several possible directions.

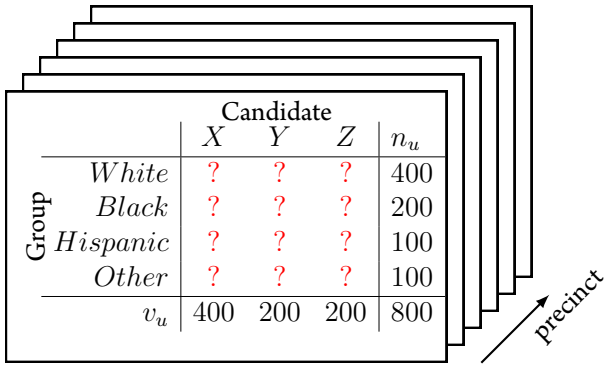
3.1 Basic model for ecological inference

In this section, we describe a hierarchical Bayesian model for the most basic version of the ecological inference problem. The setting is a plurality election in a jurisdiction where the voters belong to R different racial (or other) groups. There are C candidates (possibly including a "did not vote" option). The jurisdiction contains U electoral units (precincts). For each precinct u, we know the following information:

- $n_{u,r} =$ the number of voters of "race" r in precinct u;
- $v_{u,c} =$ the number of voters in precinct u who voted for candidate c.

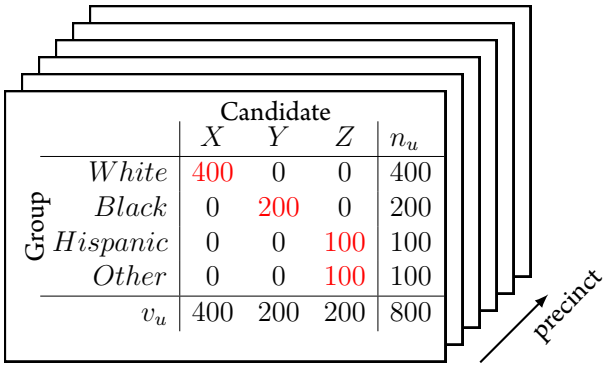
We will denote the combined data for precinct u as $\boldsymbol{x}_u$. Technically, one might consider the $\boldsymbol{v}$ as observations and the $\boldsymbol{n}$ as givens, but for our purposes, their meaning is nearly symmetric: they can be seen as R row sums and C column sums for each of P different matrices. We wish to infer how the votes for each candidate were distributed among the different racial groups; this could be seen as the element-wise sum of all the matrices.

Figure 3.1.1 shows a simple example of a single hypothetical precinct worth of observations, along with two possible underlying voting patterns consistent with those observations. Note that, ignoring integer constraints, any possible set of observations is consistent with a unique possibility in which a randomly-chosen voter's "race" is independent of their candidate support, as in case B of the figure; we will use this fact later.

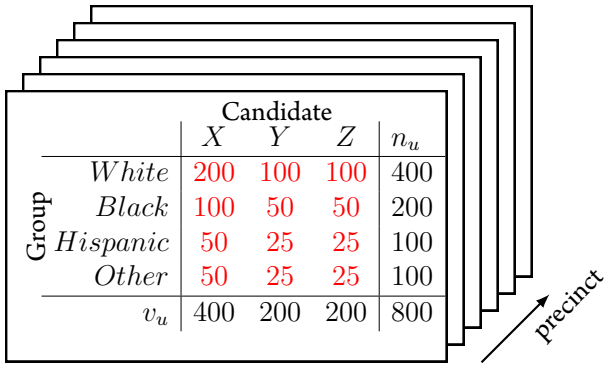


Observed data for one hypothetical precinct

.



Possible underlying data A: assuming largest groups vote for most-popular candidates



Possible underlying data B: assuming candidate's percent support is independent of racial/ethnic group

Figure 3.1.1: Observations for one hypothetical precinct, and two possible underlying vote patterns consistent with those observations.

From a Bayesian point of view, the model's function is to make reasonable

assumptions about patterns of voting — for instance, that across precincts, a given racial group tends to vote in similar proportions for a given candidate — in order to allow us to infer from data which voting totals are more credible.

For ease of explanation, we begin with a relatively-simple basic model, shown graphically below in Figure 3.1.2. In Section 3.5, we will discuss possible extensions of this model, such as analyzing data for multiple elections and/or incorporating additional information such as the racial category or party of each candidate.

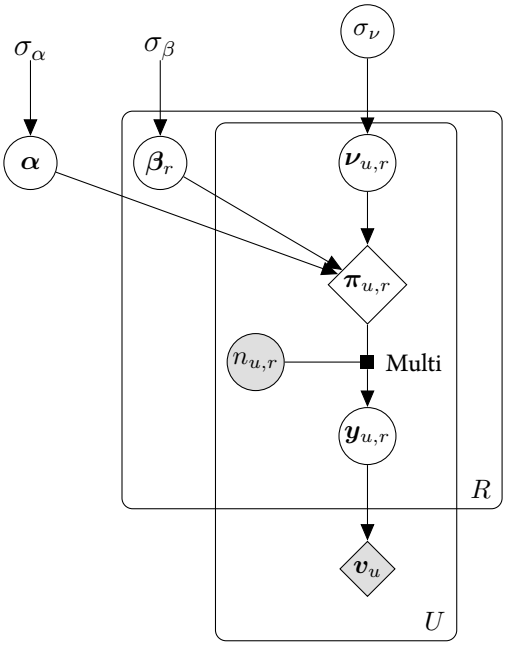


Figure 3.1.2: The basic model for ecological inference.
(*Gray nodes represent observed quantities. Diamonds represent computed quantities.*)

As usual, gray and white nodes in Figure 3.1.2 represent observed and unobserved quantities respectively. Circles represent random variables, while diamonds are computed deterministically. Beginning from the bottom of the diagram, we have:

- **Quantities considered known *a priori***: Vectors $\boldsymbol{n}_u \in \mathbb{R}^R$ for each

$u \in \{1, \ldots, U\}$, with entries $n_{u,r}$; the number of (potential) voters for each "race".

- **Observed variables**: Vectors $\boldsymbol{v}_u \in \mathbb{R}^C$ for each $u \in \{1, \ldots, U\}$, with entries $v_{u,c}$ as above; the total number of voters choosing each option.

- **Latent variables:** Vectors $\boldsymbol{y}_{u,r} \in \mathbb{R}^C$ for each $u \in \{1, \ldots, U\}$ and $r \in \{1, \ldots, R\}$. The entry $y_{u,r,c}$ represents the number of voters from group r in precinct u voting for candidate c. For each precinct u, the vectors $\boldsymbol{y}_{u,1}, \ldots, \boldsymbol{y}_{u,R}$ can be stacked to form the rows of the $R \times C$ precinct vote matrix Y_u — the matrix depicted in Figure 3.1.1.

  Although the vectors $\boldsymbol{y}_{u,r}$ are unknown to us, they are in principle observable and independent of any particular model. For this reason, we refer to them as "latent variables" rather than "model parameters". Note, however, that in a Bayesian setting, there is no fundamental distinction between parameters and latent variables. In particular, in the context of variational Bayesian inference as described in Section 3.2, the $\boldsymbol{y}_{u,r}$ will fall under the category of "parameters", since they are unobserved and need to be inferred.

  For each u and r, the vector $\boldsymbol{y}_{u,r}$ has distribution

$$\boldsymbol{y}_{u,r} \sim \text{Multinomial}\left(n_{u,r}, \boldsymbol{\pi}_{u,r}\right), \tag{3.1}$$

  where the probability vector $\boldsymbol{\pi}_{u,r}$ is computed from the model parameters, as described below.

- **Model parameters:** These are quantities that contribute to $\pi_{u,r,c}$, the probability of an individual voter of "race" r in precinct u voting for candidate c. We distinguish between two types of model parameters:

  - *Global parameters* depend on characteristics of the voter or candidate that apply across the entire jurisdiction, independent of the precinct

u. In our simple version of the model, the only such characteristic is the voter's "race" r. Our global parameters consist of:

- A vector $\boldsymbol{\alpha} \in \mathbb{R}^C$, restricted to the $C-1$-dimensional subspace of mean-0 vectors, whose probability density is proportional to that of $\boldsymbol{\alpha} \sim \mathcal{N}(0, {\sigma_\alpha}^2 I)$.[1] The entry α_c corresponds to the overall support for candidate c, across all racial groups and precincts.
- A matrix $\boldsymbol{\beta}$, whose rows are mean-0 vectors $\boldsymbol{\beta}_r \in \mathbb{R}^C$ for each $r \in \{1, \ldots, R\}$, and whose columns are also restricted to have mean 0; with density proportional to $\beta_{r,c} \sim \mathcal{N}(0, \sigma_\beta^2 I)$. Thus, $\boldsymbol{\beta} \in \mathbb{R}^{R \times C}$, but it is restricted to a subspace of dimension $(R-1)(C-1)$.[2] The entry $\beta_{r,c}$ corresponds to an additional preference for (or against) candidate c specific to racial group r, across all precincts.

  In more complex models, we may have additional global parameters corresponding to the "race" or party of the candidate, the year or type of election, etc.

- *Nuisance parameters* depend on the precinct u and correspond to random variation in voting patterns between precincts.

  In our basic model, the nuisance parameters consist of matrices $\boldsymbol{\nu}_u \in \mathbb{R}^C$, with rows for each $r \in \{1, \ldots, R\}$ and $u \in \{1, \ldots, U\}$, iid with distribution $\nu_{u,r,c} \sim \mathcal{N}(0, \sigma_\nu^2 I)$. The entry $\nu_{u,r,c}$ correspond to additional preference for (or against) candidate c among voters of "race" r specifically in precinct u.

  In more complex models we may have additional precinct-level parameters. For instance, if the data included elections for different offices and in different years, we might have parameters for party by

[1] Expressing the prior in this form may seem strange, but it is symmetric over "races" and computationally-convenient.

[2] Note that the $R(C-1)$ total dimensions of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the right number to allow any average pattern of voting preferences for each "race".

precinct as well as party by precinct by year by office. The choice of exactly which combinations of factors to include at the precinct level would depend on the scientific questions the model was intended to address, but these nuisance parameters generally should be kept to only one or two if possible.

Because nuisance parameters are much more numerous and of less intrinsic interest than the global parameters, we will deal with them differently when constructing the guide family to perform variational inference.

Given the model parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\{\boldsymbol{\nu}_u\}$, the probability that a voter of "race" r in precinct u votes for candidate c is

$$\pi_{u,r,c} = \frac{\exp(\alpha_c + \beta_{r,c} + \nu_{u,r,c})}{\sum_{\tilde{c}} \exp(\alpha_{\tilde{c}} + \beta_{r,\tilde{c}} + \nu_{u,r,\tilde{c}})}. \tag{3.2}$$

In other words, according to our model, a voter's propensity to vote for candidate c is proportional to a product of lognormally-distributed parameters that correspond to the overall strength of the candidate (α_c), the global preferences of the voter's racial group ($\beta_{r,c}$), and precinct-level effects by "race" ($\nu_{u,r,c}$). We assume that, conditional on these parameters, each individual voter's decision is independent.

- **Model hyperparameters**: In our simple model, these are just the three quantities σ_α, σ_β, and σ_ν, which control the distributions of the parameters α_c, $\beta_{r,c}$, and $\nu_{u,r,c}$ respectively.

  In practice, we usually set $\sigma_\alpha = \sigma_\beta = 2$, which is large enough to allow the model plenty of flexibility to fit cases where candidates differ in popularity by factors of 100 or more. (These values are constants because these values are only very weakly constrained by the data, and as long as they are not too low, the posterior estimates of α, β should be good). On the other hand, since our prior belief is that random variation between

precincts is, on average, relatively small, we let $\log(\sigma_\nu) \sim \mathcal{N}(-2.5, 1.2)$, corresponding to a 90% credible interval of roughly $(0.01, 0.6)$ for σ_ν. That is, roughly speaking, the odds of a person of a given race voting for a given candidate may typically vary by as little as a factor of 1.01, or as much as a factor of 1.6, across precincts.

In more complex problems, we may have additional hyperparameters corresponding to the correlation between different model parameters; see Section 3.5, Example 2 for details.

Generally, in a hierarchical Bayesian model, we are given a distribution of the observed variables conditional on the latent variables and parameters. But in ecological inference, the interaction between the data and the model is somewhat different. The observed variables $\boldsymbol{n}_u$ and $\boldsymbol{v}_u$ impose *deterministic constraints* on the row and column sums of the matrix Y_u of latent variables:

$$\begin{aligned} \text{Row sums} \quad &: \quad \sum_{c=1}^{C} y_{u,r,c} \;=\; n_{u,r} \quad \text{for each } r; \\ \text{Column sums} \quad &: \quad \sum_{r=1}^{R} y_{u,r,c} \;=\; v_{u,c} \quad \text{for each } c. \end{aligned}$$

In other words, for each precinct u, the data vectors $\boldsymbol{n}_u$ and $\boldsymbol{v}_u$ define an $(R-1) \times (C-1)$-dimensional polytope $\bar{\mathcal{Y}}_u$ in the space of $R \times C$ matrices, and the likelihood $P\left(\boldsymbol{n}_u, \boldsymbol{v}_u \mid Y_u\right)$ is simply the indicator function of this polytope.[3]

To illustrate this idea, consider Figure 3.1.3, showing a precinct u in an election with 3 candidates and 2 "races" (unlabeled). In this 3×2 case, the polytope is 2-dimensional; the figure shows what it would look like projected into the space of $y_{u11} \times y_{u12}$.

[3] Actually, the likelihood is the indicator function of the polytope $\bar{\mathcal{Y}}_u$ only for integer values of all elements, and is 0 for any non-integer values. We address this issue in Section 3.3.2.

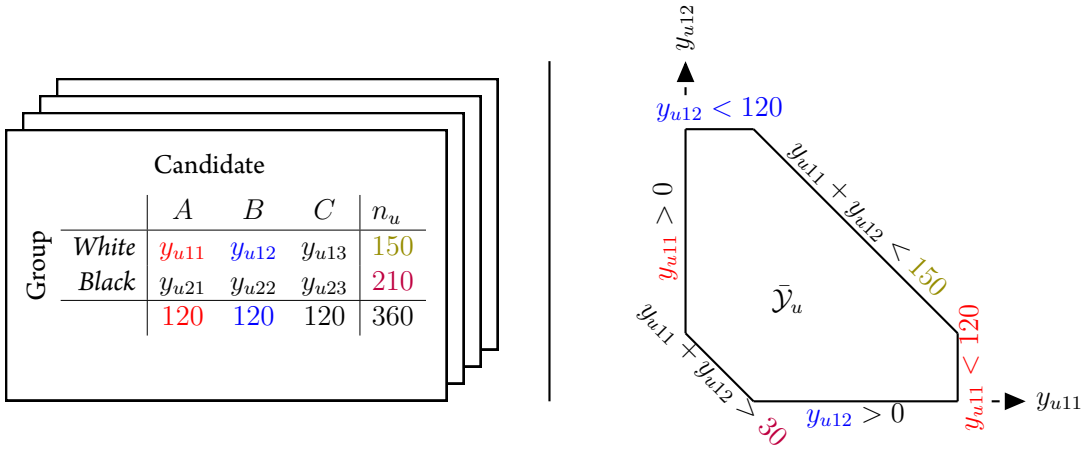


Figure 3.1.3: Example of precinct-level observations and the resulting $\bar{\mathcal{Y}}_u$.

For ease of notation, we combine our parameters into sets as follows:

- For each precinct u, the known quantities $\boldsymbol{n}_u$ and the observed variables $\boldsymbol{v}_u$ into a single vector
$$\boldsymbol{x}_u \in \mathbb{R}^{C+R}; \tag{3.3}$$
- the global parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_R$ into a single vector
$$\boldsymbol{\gamma} \in \boldsymbol{\Gamma} \simeq \mathbb{R}^{C+RC}; \tag{3.4}$$
- the precinct-level nuisance parameters $\boldsymbol{\nu}_{u,r}$ (for all u and r) into a single vector
$$\boldsymbol{\nu} \in \mathbb{R}^{URC}; \tag{3.5}$$
- the matrices Y_u (for all u) into a single vector of matrices
$$\boldsymbol{Y} \in \mathbb{M}_{R\times C}^{U} \simeq \mathbb{R}^{URC}. \tag{3.6}$$

The posterior distribution of all model parameters and latent variables is just the prior distribution with Y_u restricted to $\bar{\mathcal{Y}}_u$, renormalized accordingly. The

hard part, as usual, is computing the normalization constant. As usual in the EI paradigm, we are most interested in the posterior distribution of the latent variables $\boldsymbol{y}_{u,r}$ rather than the model parameters $\boldsymbol{\beta}_r$.

3.2 Variational inference and Laplace families

To obtain the posterior on y_u, we will be using variational inference, with a transformed, amortized Laplace guide family, as described in Chapter 2. In this section, we remind the reader of the basic concepts and notation. (Readers who have Chapter 2 fresh in their minds can safely skip this section.)

Suppose we have a set of observations $\boldsymbol{x}$ and a model for these observations with parameters $\boldsymbol{\theta} \in \mathbb{R}^D$. In other words, we are given a prior distribution $p(\boldsymbol{\theta})$ and a likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$. We are interested in the posterior distribution

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int_{\boldsymbol{\theta}} p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}. \tag{3.7}$$

The variational approach is to approximate the posterior distribution by a sampleable **guide** distribution $q_\phi(\boldsymbol{\theta})$ belonging to some **guide family** $\mathcal{Q}_\Phi$ parametrized by $\phi \in \Phi$. We wish to find the value of ϕ that minimizes the Kullback-Leibler divergence between the guide and the posterior. This turns out to be equivalent to maximizing (usually using some form of gradient ascent) an expression known as the **ELBO**:

$$\begin{aligned}\text{ELBO}(\boldsymbol{\phi}) &:= E_{q_{\boldsymbol{\phi}}}\left[\log p(\boldsymbol{x}, \boldsymbol{\theta}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})\right] \\ &= \int [\log p(\boldsymbol{x}, \boldsymbol{\theta})] q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) d\boldsymbol{\theta} - \int [\log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})] q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) d\boldsymbol{\theta}.\end{aligned}$$

So the fitted guide is $q_{\hat{\boldsymbol{\phi}}}(\boldsymbol{\theta})$, where

$$\hat{\boldsymbol{\phi}} = \arg\max_{\boldsymbol{\phi}} \left(\text{ELBO}(\boldsymbol{\phi})\right). \tag{3.8}$$

Of course, how well $q_{\hat{\phi}}(\boldsymbol{\theta})$ actually approximates $p(\boldsymbol{\theta}|\boldsymbol{x})$ depends on the choice of the variational family $\mathcal{Q}_\Phi$. The most common choice, known as **mean-field family**, is a family of Gaussians with the covariance matrix constrained to be diagonal (in order to limit the number of variational parameters to be learned).

However, the mean-field family is clearly inappropriate for EI, since the key model parameters $y_{u,r,c}$ are clearly highly correlated for each u. Instead, we use the **Laplace guide family** $\mathcal{L}_\Phi$ introduced in Chapter 2, which has roughly the same number of parameters as the mean-field family, yet is also able to accurately capture the correlation structure of the posterior. In particular, the Laplace family contains (a slight distortion of) the Laplace approximation to $p(\boldsymbol{\theta}|\boldsymbol{x})$ at every mode of p. We can therefore expect the fitted guide $q_{\hat{\phi}}(\boldsymbol{\theta})$ to be at least as good at approximating p as the Laplace approximation at the dominant mode — and possibly better.

A Laplace family guide $q_\phi \in \mathcal{L}_\Phi$ is a multivariate normal distribution, with the primary component of the parameter ϕ encoding the mean $\boldsymbol{\theta}^*$, while the covariance matrix is obtained from the inverse of the "observed" information

$$\mathscr{I}_{p(\boldsymbol{\theta},\boldsymbol{x})}\left(\boldsymbol{\theta}^*\right) := -H\left[\log p(\boldsymbol{\theta},\boldsymbol{x})\right]\Big|_{\boldsymbol{\theta}^*}. \tag{3.9}$$

Here, H is the Hessian of $\log p(\boldsymbol{\theta},\boldsymbol{x})$ with respect to $\boldsymbol{\theta}$, evaluated at $\boldsymbol{\theta} = \boldsymbol{\theta}^*$. If $\boldsymbol{\theta}^*$ is a local maximum of $p(\boldsymbol{\theta},\boldsymbol{x})$, then $q_\phi(\boldsymbol{\theta})$ is the Laplace approximation of $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ at $\boldsymbol{\theta}^*$.

Although the matrix $\mathscr{I}\left(\boldsymbol{\theta}^*\right)$ itself may not be positive definite, we can adjust it to be so using a parametrized family of "boosting" functions $f_{\boldsymbol{\psi}}$. The adjustment is controlled by a second set of guide parameters $\boldsymbol{\psi}$. When $\mathscr{I}\left(\boldsymbol{\theta}^*\right)$ is already positive definite (and therefore needs no adjustment), we ensure that $f_{\boldsymbol{\psi}}(\mathscr{I}\left(\boldsymbol{\theta}^*\right)) \to \mathscr{I}\left(\boldsymbol{\theta}^*\right)$ as $\boldsymbol{\psi} \to \mathbf{0}$. (For details on boosting functions, please refer to Chapter /refch2.)

Due to the presence of precinct-level latent variables, the number of parameters in an EI model is likely to be very large. To limit the number of guide

parameters, we make use of a standard method for variational inference in latent variable models, **amortization**. We split the model parameters $\boldsymbol{\theta}$ into global parameters $\boldsymbol{\gamma}$ (i.e. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}_r$ for each r) and latent variables/parameters $\boldsymbol{\lambda}_u$ for each precinct u (i.e. $\boldsymbol{\nu}_{u,r}$ and Y_u for each u and r). The guide parameters $\boldsymbol{\theta}^*$ are similarly split into $\boldsymbol{\gamma}^*$ and $\boldsymbol{\lambda^*}_u$, with $\boldsymbol{\lambda^*}_u$ constrained to be a function of $\boldsymbol{\gamma}^*$ and the precinct-u observations $\boldsymbol{x}_u$. The resulting amortized Laplace family allows inference that is both faster and slightly more robust than a pure Laplace approach. Please refer to Chapter 2 for details.

3.3 Variational inference for EI

Our goal is to approximate the posterior $p\left(\sigma_\nu, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{Y} \mid \boldsymbol{x}\right)$ using variational inference with a Laplace family of normal distributions. However, there are a few issues that need to be addressed before we can proceed:

1. Our model contains a discrete (multinomial) component, whereas variational inference methods generally rely on the posterior being differentiable (and the Laplace approach in particular requires the posterior to be thrice-differentiable almost everywhere);

2. As explained in Section 3.1, the support of $p\left(\boldsymbol{Y} \mid \boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{\nu}\right)$ is $\Pi_{u=1}^{U} \bar{\mathcal{Y}}_u$, where each $\bar{\mathcal{Y}}_u$ is a closed $(R-1)(C-1)$-dimensional polytope in $\mathbb{M}_{R\times C}$. Thus, because of its highly constrained support, $p\left(\sigma_\nu, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{Y} \mid \boldsymbol{x}\right)$ does not lend itself well to a normal approximation.

3. The global parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}_r$ are naturally expressed as dimension R and $R \times C$ respectively, but they are restricted to subspaces of dimension $R-1$ and $(R-1)\times(C-1)$, because changes in these parameters that are perpendicular to the product of those subspaces do not affect the likelihood.[4]

[4] The analogous redundancy in $\boldsymbol{\nu}^*$ is not an issue, since we intend to amortize all the parameters in $\boldsymbol{\nu}^*$ rather optimizing them individually.

3.3.1 Handling discrete parameters

The first problem is relatively easy to deal with: we can define a **"continuous multinomial"** distribution,[5] CMult$(n, \boldsymbol{\pi})$ with unnormalized density

$$f(y_1, \ldots, y_C) = \begin{cases} \Pi_{c=1}^{C} \frac{\pi_c^{y_c}}{\Gamma(y_c+1)} & \text{if } \sum_{c=1}^{C} y_c = n \text{ and } y_c \geq 0 \text{ for all } c; \\ \\ 0 & \text{otherwise.} \end{cases} \tag{3.10}$$

We can now change our model to have $(\boldsymbol{y}_{u,r}|\boldsymbol{\gamma}, \boldsymbol{\nu}) \sim \text{CMult}(n_{u,r}, \boldsymbol{\pi}_{u,r})$, where $\boldsymbol{\pi}_{u,r}$ is computed from $\boldsymbol{\gamma}$ and $\boldsymbol{\nu}$ as in Section 3.1. Note that we do not need to compute the normalizing constant for CMult, since the ELBO requires only an unnormalized density.

Note that CMult is a good approximation of the multinomial distribution for values far away from the boundary, but is less so near the boundary. This is because the total probability mass over an interval which is away from the boundary is likely to be roughly the same for the two distributions, but this is not so at the boundary; CMult, unlike multinomial, almost never exactly takes extreme values 0 or n. In practice, we deal with this by adding pseudo-voters, as described in the following section, and then subtracting those pseudo-voters out of our estimates at the end. This effectively extends the boundary of CMult slightly beyond that of multinomial, allowing impossible negative voter estimates in rare cases but hopefully reducing the bias overall.

3.3.2 Handling polytope support

We address the second problem by reparametrizing the polytopes $\bar{\mathcal{Y}}_u$. For each $u \in \{1, \ldots, U\}$, we define a bijective, almost-everywhere smooth, mapping

$$m_u : \mathbb{R}^{(R-1)(C-1)} \to \mathcal{Y}_u, \tag{3.11}$$

[5] When presenting this material, we have been asked if the continuous multinomial corresponds to any reasonable data-generating process. We believe that the answer is no; we are using it merely as a differentiable approximation to the multinomial.

where $\mathcal{Y}_u$ is the data-specific open polytope (not including the boundary). Via this reparametrization, any continuous probability distribution on the open set $\mathcal{Y}_u$ corresponds to a distribution on $\mathbb{R}^{(R-1)(C-1)}$ with unconstrained support. One possible construction for m_u is given in Appendix 3.3.

Note that this reparametrization switched from the closed polytope to the open one, which would mean that a posterior sample can never estimate exactly zero voters in a given y_{urc}. This is a similar issue to that of the difference between CMult and multinomial near the boundary, mentioned above, and so we hope that our step of adding pseudo-voters while fitting but subtracting them from the final estimates will address both of these issues simultaneously. In effect, this allows estimates of y_{urc}, after the pseudo-voters are subracted out, to be slightly negative; we believe that in this way, the total posterior probability mass near zero will roughly approximate the probability mass that would be exactly at zero if we were able to use a discrete multinomial in our model.

The product of the maps $m_1, \ldots, m_U$ gives us a global reparametrization

$$m_{\boldsymbol{x}} : \mathbb{R}^{U(R-1)(C-1)} \to \Pi_{u=1}^{U} \mathcal{Y}_u. \tag{3.12}$$

For any $\boldsymbol{W} = (W_1, \ldots, W_U) \in \mathbb{R}^{U(R-1)(C-1)}$, we can now define an a.e. smooth probability density function based on the density function over $\mathcal{Y}_u$

$$\tilde{p}_{\boldsymbol{x}}(\sigma_\nu, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{W}) := p\left(\sigma_\nu, \boldsymbol{\gamma}, \boldsymbol{\nu}, m_{\boldsymbol{x}}(\boldsymbol{W}) \mid \boldsymbol{x}\right) \cdot |J_{m_{\boldsymbol{x}}}(\boldsymbol{W})|, \tag{3.13}$$

where $J_{m_{\boldsymbol{x}}}$ is the Jacobian of $m_{\boldsymbol{x}}$.

3.3.3 Subspace for global parameters

To address the third problem, we define reduced-dimension guide parameters $\boldsymbol{\alpha}^* \in \mathbb{R}^{C-1}$ and $\boldsymbol{\beta}^* \in \mathbb{R}^{R(C-1)}$. That is, we leave the redundant model parameters $\boldsymbol{\alpha}_C$, $\boldsymbol{\beta}_{r,C}$, and $\boldsymbol{\beta}_{R,c}$ out of the guide's parameter space, and generate them from $\boldsymbol{\alpha}^*$ or $\boldsymbol{\beta}^*$ as needed.

The graphical representation of $\tilde{p}_{\boldsymbol{x}}$ is shown in Figure 3.3.1, with differences

between $\tilde{p}_{\boldsymbol{x}}$ and $p\left(\sigma_\nu, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{Y} \mid \boldsymbol{x}\right)$ highlighted in red.

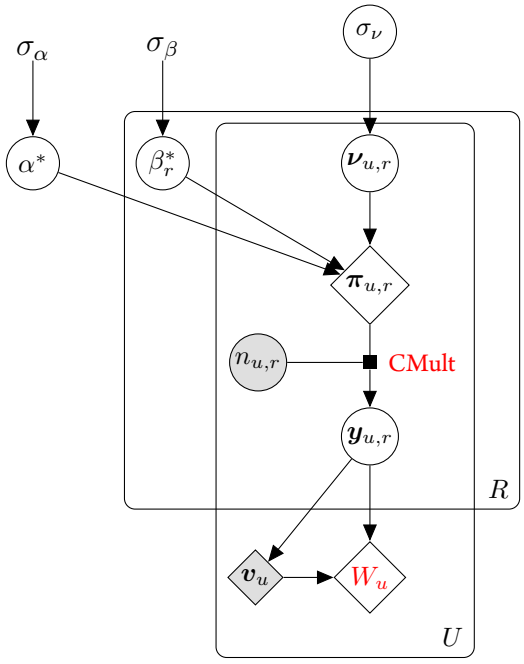


Figure 3.3.1: Model for $\tilde{p}_{\boldsymbol{x}}(\sigma_\nu, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{W})$.
(*Gray nodes represent observed quantities. Diamonds represent computed quantities.)*

In the next section, we will show how to use variational inference to find an approximation $q_{\hat{\boldsymbol{\phi}}}(\sigma_\nu, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{W})$ for $\tilde{p}_{\boldsymbol{x}}$. We can then use the map $m_{\boldsymbol{x}}^{-1}$ to transform $q_{\hat{\boldsymbol{\phi}}}$ back to an approximation of our original posterior distribution $p(\sigma_\nu, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{Y} \mid \boldsymbol{x})$, as desired.

3.3.4 Defining the guide family

To approximate $\tilde{p}_{\boldsymbol{x}}(\sigma_\nu, \boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{W})$, we use (a slight modification of) the amortized Laplace family as described in chapter 2.

We define

$$\begin{aligned}
\boldsymbol{\gamma}^* &:= (\boldsymbol{\alpha}^-, \boldsymbol{\beta}^-) \in \mathbb{R}^{R(C-1)}; \qquad (3.14)\\
\sigma_{\boldsymbol{\nu}} &\in \mathbb{R}^1_+;\\
{\boldsymbol{\lambda}^*}_u &:= (\boldsymbol{\nu}^*_u, \ldots, \boldsymbol{\nu}^*_u, W^*_u) \in \mathbb{R}^{RC+(R-1)(C-1)} \text{ for each } u;\\
\Theta_M &:= \left\{(\boldsymbol{\gamma}^*, \boldsymbol{\lambda}^*_1, \ldots, \boldsymbol{\lambda}^*_U) \in \Theta : \boldsymbol{\lambda}^*_u = M_u(\boldsymbol{\gamma}^*) \text{ for } u = 1, \ldots, U\right\},
\end{aligned}$$

where

$$M_u : \mathbb{R}^{R(C-1)} \to \mathbb{R}^{1+RC+(R-1)(C-1)} \qquad (3.15)$$

is a differentiable function that approximates the MAP of $(\boldsymbol{\nu}_u, W_u)$ conditional on $\boldsymbol{\gamma}^*$ (and implicitly, via the polytope mapping, on $\boldsymbol{x}_u$):

$$(\sigma^*_\nu, \boldsymbol{\nu}^*, W^*) := M(\boldsymbol{\gamma}^*) \approx \arg\max_{\sigma_\nu, \boldsymbol{\nu}, W)}(\tilde{p}_{\boldsymbol{x}}(\boldsymbol{\gamma}^*, \boldsymbol{\nu}, \boldsymbol{W}). \qquad (3.16)$$

The derivation for the function $M_u(\boldsymbol{\gamma}^*)$ can be found in Appendix 3.4.1.

Because both our amortization function M and our transformation m are poor when any elements of $\boldsymbol{Y}$ are less than 1, we add $k = 1$ "pseudo-voter" to each such element immediately after finding $\boldsymbol{Y}^*$, as well as adding corresponding amounts to the observations: C to each n_u, and R to each v_u. We have no principled reason to choose $k = 1$ as the optimal correction factor here, but have reason to believe that the optimal k is greater than 0, both in order to reduce the bias caused by the difference between Stirling's approximation and the Gamma function when $y_{u,r,c}$ is less than 1, and because subtracting these pseudo-voters back out at the end allows the distribution of $m_u(\boldsymbol{W}_u)$ to include the boundaries of the true, closed polytope (and a bit beyond), rather than being artificially restricted to the open polytope by the function m_u.

We define the guide $q_\phi\,(\boldsymbol{\gamma}, \sigma_\nu, \boldsymbol{\nu}, \boldsymbol{W})$ to be a multivariate normal with mean $(\boldsymbol{\gamma}^*, \sigma^*_\nu, \boldsymbol{\nu}^*, \boldsymbol{W}^*)$. That is to say, to sample from the guide, we'd draw from a Gaussian over latent W space, then transform that to Y space including pseudo-voters, then subtract out pseudo-voters. The precision (inverse

covariance) matrix of this multivariate normal is "Laplacian": the observed information matrix of the model with respect to all arguments at the mean of the guide, adjusted to be positive definite: $f_{\psi}\left[\mathscr{I}_{\tilde{p}_{\boldsymbol{x}}}\left(\boldsymbol{\gamma}^{*}, \sigma_{\nu}^{*}, \boldsymbol{\nu}^{*}, \boldsymbol{W}^{*}\right)\right]$.

As explained in chapter 2, we can take advantage of the block arrowhead structure of this Hessian for several optimizations, including when drawing samples from the guide.

A graphical representation of the guide $q_{\phi}(\boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{W})$ is shown in Figure 3.3.2. The optimization of the ELBO will be performed through Pyro, a Python package built for stochastic variational inference.

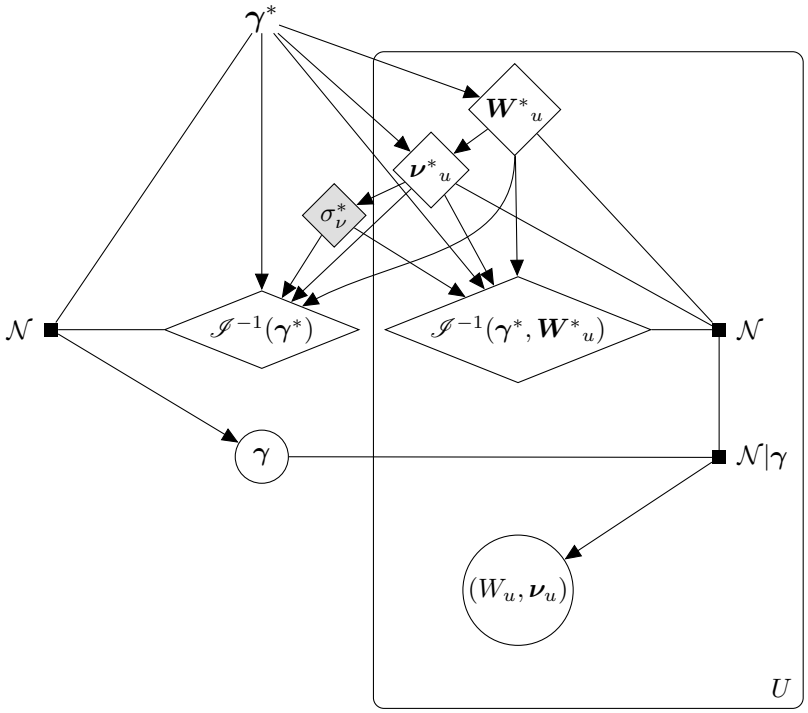


Figure 3.3.2: Graphical model for the guide $q_{\phi}(\boldsymbol{\gamma}, \sigma_{\nu}, \boldsymbol{\nu}, \boldsymbol{W})$. As usual, diamonds represent computed quantities. The grey diamond for σ_{ν}^{*} indicates that it is treated as a constant for purposes of computing the ELBO gradient.

3.3.5 Algorithm for amortizing Laplace variational EI

Our algorithm for fitting the above ecological inference model follows the general Laplace guide algorithm as described in section 2.3.6 of chapter 2. To do so, we must translate generalities to specifics in several regards; most importantly, in terms of our algorithm for approximating
$M(\boldsymbol{\gamma}^*) = (\sigma_\nu^*, \boldsymbol{\nu}^*, \boldsymbol{W}^*) \approx \arg\max_{\sigma_\nu, \boldsymbol{\nu}, \boldsymbol{W}} \tilde{p}(\boldsymbol{\gamma}^*, \sigma_\nu, \boldsymbol{\nu}, \boldsymbol{W})$.

The basic structure of this approximation process is sketched out below, and fully explained in appendix 3.4.2. It involves a series of rough first-order approximations, with one step of Newton's Method at the end to reduce both the bias and the variance of the approximations. We understand that some of the decisions in designing this procedure were arbitrary and could probably be improved upon, but the overall algorithm seems to work.

The basic steps and formulas are as follows. For an explanation of the logic behind each formula, see Appendix 3.4.2.

1. Find $\boldsymbol{Y}^* \approx \arg\max_{\boldsymbol{Y}} \tilde{p}(\boldsymbol{\gamma}^*, \sigma_\nu = 0, \boldsymbol{\nu} = \vec{\boldsymbol{0}}, \boldsymbol{Y})$. This process is described in Appendix 3.4.1. Use m^{-1} to turn this into $\boldsymbol{W}^*$.[6]
2. Estimate $\hat{\boldsymbol{\nu}} := \log\left(\frac{y^*_{u,r,c}}{n_{u,r}\pi^*_{r,c}}\right)$, an estimator of $\boldsymbol{\nu}$.
3. Estimate $\hat{\sigma}^2_{u,r,c} := \frac{n_{u,r} - y^*_{u,r,c}}{n_{u,r} y^*_{u,r,c}}$, an estimator of $\sigma^2_{u,r,c}$, the part of the variance of $\hat{\boldsymbol{\nu}}_{u,r,c}$ that is attributable to sampling variance in $\boldsymbol{y}_{u,r,c}$.
4. Take $max(0, \hat{\boldsymbol{\nu}}^2_{b,r,c} - \hat{\sigma}^2_{u,r,c})$, for each u, r, c, as estimates of σ_ν; average these estimates to get σ^*_ν.
5. Take ${\boldsymbol{\nu}^*}_{u,r,c} := \frac{\hat{\boldsymbol{\nu}}_{u,r,c}/\hat{\sigma}^2_{u,r,c}}{1/\hat{\sigma}^2_{u,r,c} + 1/(\sigma^*_\nu)^2}$, combining the "likelihood" distribution with approximate mean and variance $[\hat{\boldsymbol{\nu}}_{u,r,c}, \hat{\sigma}^2_{u,r,c}]$ and the "prior" distribution with mean and variance $[0, \sigma^*_\nu]$, using a precision-weighted average.

[6] We are aware that, because of the Jacobian of m, this introduces bias; $\arg\max_{\boldsymbol{W}} \tilde{p}(m(\boldsymbol{W}), ...) \neq m^{-1}[\arg\max_{\boldsymbol{Y}} \tilde{p}(\boldsymbol{Y}, ...)]$. Currently, we are hoping that the one step of Newton's method at the end addresses this issue sufficiently. We plan to develop a better fix for this issue in future versions of the algorithm.

6. After taking the Hessian, use one step of Newton's method on each precinct u's latent parameters to redefine $(\boldsymbol{W}^*{}_u, \boldsymbol{\nu}^*{}_u)$, as explained in section 2.3.5 of chapter 2.

3.4 Results

We tested our algorithm on simulated voting data for 2774 precincts, with precinct racial composition based on the demographics of actual registered voters in North Carolina in 2016.[7] Simulated election results for three "candidates" (Democrat, Republican, Other/not voting) were generated from the basic EI model in Section 3.1, with model parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ set to mirror the statewide turnout and exit polls in North Carolina for the 2016 Presidential election (Tables 3.4.1 and 3.4.2).[8]

Table 3.4.1: 2016 Presidential election in North Carolina: turnout and exit polls

		Of those who voted, percent voting for:		
"Race"	Voter Turnout	Clinton (D)	Trump (R)	Other
Black	64%	89%	8%	3%
White	71%	32%	63%	5%
Other	59%	56%	40%	4%

Using the above values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}_r$, we created four datasets with low precinct-to-precinct variability ($\sigma_\nu = 0.02$) and four with high variability ($\sigma_\nu = 0.3$). We chose $\sigma_\nu = 0.02$ as our low value because, for most precincts in

[7] Precinct-level voter registration data by "race" were obtained from the North Carolina State Board of Elections. [37] All "races" other than white and black were combined into a single category "other". One voter per racial category was added to each precinct to avoid zeros; this was taken to be ground truth.

[8] Sources: https://www.cnn.com/election/2016/results/exit-polls/north-carolina/president; https://www.wfae.org/post/numbers-are-breaking-down-ncs-2016-voter-demographics#stream/0

Table 3.4.2: 2016 Presidential election in North Carolina: model parameters

"Race"	Clinton (D)	Trump (R)	Other/not voting
Black	$\beta = 0.49$	$\beta = -1.23$	$\beta = 0.01$
White	$\beta = -0.43$	$\beta = 0.94$	$\beta = -0.15$
Other	$\beta = -0.06$	$\beta = 0.30$	$\beta = 0.14$
Total	$\alpha = 0.20$	$\alpha = -0.48$	$\alpha = 0.28$

our data, this setting makes the variability in y_{urc} due to differences in ν_{urc} only slightly higher than the variability in y_{urc} due to multinomial sampling.

For each dataset, we compare the performance of our algorithm with that of RJKT, which uses MCMC to sample from a hierarchical Dirichlet-multinomial model[48]. Our algorithm was implemented in python using the pyro probabilistic programming package[9]. For the RJKT algorithm, we used the eiPack package in R[31].

As usual in EI, we focus our attention on the posterior distribution of the $R \times C$ matrix Q, where $Q_{r,c}$ is the overall fraction of voters of "race" r who voted for candidate c:

$$Q_{r,c} = \frac{\sum_{u=1}^{U} Y_{u,r,c}}{\sum_{u=1}^{U} n_{u,r}}. \tag{3.17}$$

Let Q_i be the value of Q in Dataset i. For each algorithm $\mathcal{A} \in \{\text{RJKT, Laplace}\}$, let $\hat{Q}^{(\mathcal{A},i)}$ and $s_Q^{(\mathcal{A},i)}$ be estimates of the posterior mean (weighted by n_u) and sample standard deviation of Q across precincts (ie, the SD of the percent value in each cell, across samples from the posterior), based on 1000 samples from the estimated posterior distribution obtained by running algorithm $\mathcal{A}$ on Dataset i.

Let $\overline{Q}_\sigma$, $\overline{Q}^{(\mathcal{A},\sigma)}$, and $s_Q^{(\mathcal{A},\sigma)}$ be the averages of Q_i, $\hat{Q}^{(\mathcal{A},i)}$, and $s_Q^{(\mathcal{A},i)}$ respectively for the datasets i where $\sigma_\nu = \sigma$. Tables 3.4.3 and 3.4.4 show the individual entries of these matrices for $\sigma_\nu = 0.02$ and $\sigma_\nu = 0.3$ respectively.

These results show that, in terms of point estimates, our model is performing comparably to RJKT for the low-variance scenario, and slightly worse for the high-variance scenario. In the latter case, our model's results are notably biased

Table 3.4.3: Results for $\sigma_\nu = 0.02$

		Other/nonvoting		Clinton (D)		Trump (R)	
	$\mathcal{A}$	$\overline{Q}$	s_Q	$\overline{Q}$	s_Q	$\overline{Q}$	s_Q
	Truth	32.4%		22.6%		45.0%	
White	RJKT	32.3%	0.086%	22.5%	0.065%	44.9%	0.070%
	Laplace	32.4%	0.015%	22.8%	0.014%	44.9%	0.016%
	Truth	38.0%		56.7%		5.28%	
Black	RJKT	38.6%	0.26%	56.1%	0.17%	4.81%	0.23%
	Laplace	37.9%	0.038%	56.1%	0.032%	5.99%	0.026%
	Truth	43.3%		32.7%		24.0%	
Other	RJKT	40.9%	1.0%	33.3%	0.37%	24.4%	0.99%
	Laplace	43.8%	0.13%	32.6%	0.11%	23.5%	0.15%

Table 3.4.4: Results for $\sigma_\nu = 0.3$

		Other/nonvoting		Clinton (D)		Trump (R)	
	$\mathcal{A}$	$\overline{Q}$	s_Q	$\overline{Q}$	s_Q	$\overline{Q}$	s_Q
	Truth	32.3%		22.7%		45.0%	
White	RJKT	32.2%	0.069%	23.0%	0.13%	44.7%	0.092%
	Laplace	32.6%	0.018%	23.5%	0.016%	43.9%	0.021%
	Truth	38.9%		55.6%		5.48%	
Black	RJKT	41.0%	0.50%	53.2%	0.53%	5.29%	0.13%
	Laplace	41.0%	0.040%	53.0%	0.031%	5.99%	0.032%
	Truth	43.0%		32.7%		24.3%	
Other	RJKT	37.3%	0.95%	35.3%	0.55%	26.0%	0.65%
	Laplace	35.5%	0.15%	33.3%	0.12%	31.2%	0.17%

towards even splits for each "race" (i.e. towards $Q_{rc} = \frac{1}{C}$ for all r).

We believe that we understand, and have a possible fix for, the reason for this bias. While RJKT's model views precinct-level residuals in terms of percentages, ours naturally does so in terms of odds. For example, consider a precinct where black and white voters will vote for Trump with probabilities 8% and 38% respectively. Our model considers it to be equally "hard" to double the odds of voting for Trump for either "race", whether it means changing the probability for white voters from 38% to 55% or changing the probability for black voters from 8% to 15%. Meanwhile, the RJKT model considers that 7% of black voters in a given precinct switching candidates is much more probable than 17% of white voters doing so. Thus, when it detects high inter-precinct variability, our model is biased towards ascribing it to larger groups. We believe that scaling the precision of $\nu_{u,r,c}$ by the expectation of $y_{u,r,c}$, conditional on observations and global parameters, may remove this bias.

A more serious issue is that our estimate of s_Q^σ is consistently lower than that of RJKT. Moreover, while for RJKT, we consistently have

$$\left|\overline{Q}_\sigma - \overline{Q}^{(\mathcal{A},\sigma)}\right| < s_Q^{(\mathcal{A},\sigma)}, \tag{3.18}$$

this is not always the case for Laplace, even when $\sigma_\nu = 0.02$. We are looking into this problem.

3.5 Possible extensions of the model

It is easy to extend the basic model to allow for any additional structure and/or extra data we are interested in. We give just three examples here.

Example 1: An important critique of ecological inference methods was expressed by Freedman et al.[16], who gave examples where KRT's original Bayesian ecological inference could give answers at odds with known underlying truth, and yet the diagnostics of the model did not raise any alarm bells. These problematic examples hinge on situations in which the voting behavior of a

majority ethnicity in a given precinct is systematically correlated with the percentage of one or more given minority ethnicities in that precinct. Such a model can have a likelihood that is very similar to that of a model without such correlations, where these cross-precinct differences in majority voting behavior are incorrectly attributed to the minority voters.

This possibility, where one behavior pattern masquerades as another because both lead to the same or similar patterns of observable outcomes, may be seen as problematic for ecological inference on two different levels, practical and philosophical. On the practical level, it can cause problems with empirical coverage of practical EI methods such as the ones proposed in this paper. On a philosophical level, it raises questions of identifiability and consistency that go to the heart of the very feasibility of any form of ecological inference.

One could begin to address this by including terms in the model to model a potentially systematic dependence of one group's behavior on the local percentage of another group. (Note that adding this to the model does not imply any stance on its causal status.) Similar to one of RKJT's modifications of the RKT model, we can add hyperparameters

$$\rho_{r,r',c} \sim \mathcal{N}(0, \sigma_\rho) \tag{3.19}$$

for each ordered triplet (r, r', c) of two racial groups and one candidate, and change the distribution of $\boldsymbol{\nu}$ as follows:

$$\nu_{u,r,c} \sim \mathcal{N}\left(\sum_{r'} \rho_{r,r',c} \log \frac{(R-1)(n_{u,r'}+1)}{R-1+\sum_{r''\neq r'} n_{u,r''}}, \sigma_\nu^2\right), \tag{3.20}$$

where $n_u = \sum_r n_{u,r}$. In other words, a [positive/negative] $\rho_{r,r',c}$ represents a tendency for racial group r to vote [more/less] for candidate c when there are more members of racial group r' present in the same precinct. For simplicity, we could limit these coefficients to 0 when $r' \neq r$, so that each racial group only cares about its own prevalence in the precinct, not the ratio between other groups.

In practice, including such terms in the model will tend to increase the posterior variance of $\boldsymbol{Y}$ (or, equivalently, $\boldsymbol{W}$). To understand why, consider that the model is now able to account for a positive correlation between the percentage of a given demographic group r's prevalence in a precinct and that precinct's observed vote totals for candidate c in two different ways: by inferring that group-r voters tend to vote for c, or by inferring that non-r voters tend to vote more for c when they live in precincts with more group-r neighbors. Although the likelihood for these two possibilities across multiple precincts will typically differ slightly, those differences will usually be small unless the number of precincts is very large. The increased posterior variance should improve the empirical coverage properties of the model, and so we recommend this sort of correlation structure be included. In fact, prior knowledge about possible values of ρ could be included in this kind of model, improving it yet further.

On a more philosophical level, we believe that such expanded models, and any resulting improved empirical coverage in cases where this can be checked, can increase our confidence that ecological inference is feasible. While the above modification to the model only allows simple linear and homoskedastic dependence of one group's behavioral odds on the prevalence of another group, the flexibility of this methodology would allow more complicated modifications that include nonlinearity and/or heteroskedasticity to be tried as well. Though dealing with this matter formally is beyond the scope of this paper, we believe due to the phenomenon known as the Bayesian Occam's Razor, that whenever the observed data is consistent with multiple underlying explanations, the fitted posterior will tend to discount complicated possibilities of cross-group behavioral dependencies in favor of simpler explanations, where variations that correlate with a group's prevalence are explained by that group's own behavior. Of course, if the data is more consistent with cross-group dependencies, a flexible model should reflect that fact in the posterior.

In any case, sensitivity analyses using model expansions such as this will be able to increase our confidence in the applicability of EI methods and the accuracy of their results.

Example 2: Suppose we are interested in whether voters from two different groups vote in similar ways.[9] We can explicitly incorporate such correlation into our model as follows:

- Let σ_β, σ_ν, and $\boldsymbol{\alpha}$ be as before.
- Add a hyperparameter $\tilde{\boldsymbol{\Sigma}}_c$ for each candidate c: an $R \times R$ matrix to control the correlation across racial groups of support for c. Give it an appropriate prior (such as an inverse Wishart distribution).
- Combine the parameter vectors $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_R$ into a single $R \times C$ matrix B, where column c has distribution $\mathcal{N}(0, \sigma_\beta^2 \tilde{\boldsymbol{\Sigma}}_c)$.
- For each u, combine the parameter vectors $\boldsymbol{\nu}_{u,1}, \ldots, \boldsymbol{\nu}_{u,R}$ into a single $R \times C$ matrix N_u, where column c has distribution $\mathcal{N}(0, \sigma_\nu^2 \tilde{\boldsymbol{\Sigma}}_c)$.
- Let $\boldsymbol{\pi}_{u,r}$ and Y_u be as before.

We can then carry out variational inference on this model exactly as in Section 3.3.

Here, once again, our primary object of analysis would not be the parameter matrix B that gives correlation propensity, but the actual votes $y_{u,r,c}$ as constrained by the observed data. However, including B as a parameter allows us to get a more accurate estimate of the latent values $y_{u,r,c}$ and of their variance, especially if there is, in fact, nontrivial correlation between the different racial groups.

Example 3: Suppose we are interested in modeling multiple elections. To do this, we add an index $e \in \{1, 2, \ldots, E\}$ to all variables and parameters. We can also define $t(e)$ for the year of election e, $\mathcal{C}(e)$ for the set of candidates running in election e, and $\pi(c)$ for the party of candidate c. Our data would consist of $\boldsymbol{n}_{e,u}$

[9] In several recent VRA cases, plaintiffs have claimed that the Gingles criteria (*Thornburg v Gingles*, 1986) can be applied not only to an individual racial minority group, but also to a coalition of such groups. For such cases, in order to establish whether the coalition groups vote as a bloc, it will be important to compare the voting behavior of each individual group in the coalition with the voting behavior of the coalition as a whole.

and $\boldsymbol{v}_{e,u}$ for each e and u, and the latent variable matrix $Y_{e,u}$ would be constrained to lie in the polytope $\boldsymbol{Y}_{e,u}^{-}$ defined as before.

There are then two different ways we can extend the basic model, each of which has a separate role to play:

- We can add more global parameters to our model, representing various quantities of interest. For instance, we can have a parameter $\gamma_{\pi(c),t(e)}$, indexed by party and year of election, to capture changes in partisanship over time, irrespective of "race" and type of election. We could also have a parameter $\eta_{\pi(c),r}$, indexed by party and "race", to capture partisan tendencies of racial groups that endure over time. As usual, the results of our analyses would be based on estimates of the latent variables $Y_{e,u}$ rather than of the model parameters γ or η. However, including these parameters in the model will help sharpen the estimates of $Y_{e,u}$.

- We can ensure that the nuisance parameters ν are indexed by election and party — that is, $\nu_{u,r,\pi(c),e}$ — and then add covariance hyperparameters to capture the fact that precinct-level variation in partisanship is likely to be somewhat stable across elections. For instance, if we have data for two consecutive elections e_1 and e_2, we could add hyperparameters $\rho_{r,\pi(c)}$ for each r and $\pi(c)$, with a uniform prior over $[-1, 1]$, and let

$$\left(\nu_{u,r,\pi(c),e_1},\ \nu_{u,r,\pi(c),e_2}\right) \sim \mathcal{N}\left((0,0),\ {\sigma_\nu}^2 \begin{bmatrix} 1 & \rho_{r,\pi(c)} \\ \rho_{r,\pi(c)} & 1 \end{bmatrix}\right) \tag{3.21}$$

  If our assumption that precinct idiosyncracies are relatively stable is correct, them this extended model will be able to better estimate such idiosyncracies, thus sharpening/improving our estimates of other quantities (most importantly $Y_{u,e}$). The improvement here is roughly analagous to a shift from an unpaired t-test to a paired one.

The examples above demonstrate the flexibility of this basic model format. Because the model assumes that the mid-level parameters like $\boldsymbol{\alpha}$ or $\boldsymbol{\nu}_{u,r}$ are

mutually distributed as a multivariate Gaussian, it is easy to add internal correlation structures, adjustments based on covariates, prior information, individual data, or other factors.

3.6 Conclusion

We have described a highly-extensible model for ecological inference, and given a means of fitting it using variational inference. We've shown that the simplest version of our model and methodology give results almost as good as existing widely-used methods. We have suggested a simple change that we hope will make those results comparable to that method. This model opens the door to several extensions that would not be possible using traditional methods, such as jointly modeling multiple elections, including hierarchical structure such as variance by county as well as by precinct, etc.

Appendices

2.1 Block-arrowhead precision matrices

Recall that for a latent variable model with global parameters $\boldsymbol{\gamma} \in \mathbb{R}^g$ and latents $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_N \in \mathbb{R}^l$, the matrix $\mathscr{I}_p(\boldsymbol{\theta}^*)$ is a block-arrowhead matrix. In this appendix, we collect some useful results about matrices of this form that are relevant to Laplace variational inference. Further formulas relating to block arrowhead matrices may be found in [23].

Theorem 1 *Let*

$$A = \begin{pmatrix} G & C_1 & C_2 & \ldots & C_N \\ C_1^T & U_1 & 0 & \ldots & 0 \\ C_2^T & 0 & U_2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_N^T & 0 & 0 & \ldots & U_N \end{pmatrix} \tag{2.22}$$

be a block-arrowhead matrix, with $G \in S_g, U_i \in S_l$, and $C_i \in \mathbb{M}_{g \times l}$ for $1 \leq i \leq N$. A is positive definite if and only if $U_1, \ldots, U_N$ and $\tilde{G} = G - \sum_{i=1}^N C_i U_i^{-1} C_i^T$ are all positive definite.

Proof: Let

$$B = \begin{pmatrix} U_1 & 0 & \ldots & 0 & C_1^T \\ 0 & U_2 & \ldots & 0 & C_2^T \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & U_N & C_N^T \\ C_1 & C_2 & \ldots & C_N & G \end{pmatrix}. \tag{2.23}$$

Since B is just a symmetric permutation of A, it has the same signature as A.

Form the LDL^T decomposition of B, $\tilde{G}$ and U_i for $i = 1, \ldots, N$:

$$\begin{aligned} B &= L_B D_B L_B^T; \\ \tilde{G} &= L_{\tilde{G}} D_{\tilde{G}} L_{\tilde{G}}^T; \\ U_i &= L_i D_i L_i^T. \end{aligned} \tag{2.24}$$

Elementary linear algebra shows that

$$L_B = \begin{pmatrix} L_1 & 0 & \dots & 0 & 0 \\ 0 & L_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & L_N & 0 \\ C_1 U_1^{-1} L_1 & C_2 U_2^{-1} L_2 & \dots & C_N U_N^{-1} L_N & L_{\tilde{G}} \end{pmatrix} \tag{2.25}$$

and

$$D_B = \begin{pmatrix} D_1 & 0 & \dots & 0 & 0 \\ 0 & D_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & D_N & 0 \\ 0 & 0 & \dots & 0 & D_{\tilde{G}} \end{pmatrix}. \tag{2.26}$$

The theorem follows immediately from the fact that a symmetric matrix is positive definite if and only if the matrix D in its LDL^T decomposition has all positive entries on the diagonal.

Theorem 1 suggests a way of constructing boosting families for block-arrowhead matrices. Let $\boldsymbol{\psi} = (\boldsymbol{\psi_\Gamma}, \boldsymbol{\psi_\Lambda})$ be as in Section 2.3.3, and suppose $f_{\boldsymbol{\psi_\Gamma}}$ and $f_{\boldsymbol{\psi_\Lambda}}$ are boosting families defined on S_g and S_l respectively. Let A be a block-arrowhead matrix as in Theorem 1. Define

$$f_{\boldsymbol{\psi}}(A) = \begin{pmatrix} G^+ & C_1 & \dots & C^N \\ C_1^T & U_1^+ & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_N^T & 0 & \dots & U_N^+ \end{pmatrix}, \tag{2.27}$$

where

$$U_i^+ = f_{\psi_\Lambda}(U_i) \quad \text{for } 1 \le i \le N, \tag{2.28}$$

$$G^+ = f_{\psi_\Gamma}\left(G - \sum_{i=1}^N C_i (U_i^+)^{-1} C_i^T\right) + \sum_{i=1}^N C_i (U_i^+)^{-1} C_i^T. \tag{2.29}$$

Then, by Theorem 1, $f_\psi(A)$ is positive definite. Moreover, if A is itself positive definite, then

$$\lim_{\psi \to 0} f_\psi(A) = A. \tag{2.30}$$

Thus f_ψ satisfies the conditions of a boosting family for block-arrowhead matrices. Note that the boosted matrix $f_\psi(A)$ is itself block-arrowhead.

Theorem 2 *Suppose*

$$\boldsymbol{\theta} = (\boldsymbol{\gamma}, \boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_N) \sim \mathcal{N}(\boldsymbol{\theta}^*, \Sigma), \tag{2.31}$$

where Σ^{-1} is a block-arrowhead matrix as in Theorem 1. Then the marginal covariance of $\boldsymbol{\gamma}$ is

$$\Sigma_{\boldsymbol{\gamma}} := \left(G - \sum_{i=1}^N C_i U_i^{-1} C_i^T\right)^{-1} \tag{2.32}$$

Proof: This follows directly from inverting the LDL^T decomposition of Σ^{-1} in the proof of Theorem 1.

For fixed g and l, we thus have the following $O(N)$ procedure for sampling from $\mathcal{N}(\boldsymbol{\theta}^*, \Sigma)$:

- Sample $\boldsymbol{\gamma} \sim \mathcal{N}\left(\boldsymbol{\gamma}^*, \Sigma_{\boldsymbol{\gamma}}\right)$;
- For each i from 1 to N, sample from the conditional normal distribution $\boldsymbol{\lambda}_i | \boldsymbol{\gamma}$:

$$\boldsymbol{\lambda}_i \sim \mathcal{N}(\boldsymbol{\lambda}_i^* + U_i^{-1} C_i^T (\boldsymbol{\gamma} - \boldsymbol{\gamma}^*), U_i^{-1}). \tag{2.33}$$

If ν and σ are held constant, then up to an additive constant, the log posterior of the multi-site model is:

$$\log[p(\boldsymbol{T} = \boldsymbol{t}/\sigma, \boldsymbol{x} = \boldsymbol{x}] = -\sum_i \left[\frac{\nu+1}{2}\log\left(1+\frac{t_i^2}{\sigma^2 d}\right) + \frac{(x_i - t_i)^2}{2s_i^2}\right] \tag{2.34}$$

We would like to maximize this in order to find the conditional MLE. Note that this could also be called the conditional MAP; because the conditioning typically screens out any prior distributions, the two are generally equivalent in this case.

Focusing on one site i at a time, the derivative of the above — that is, the score function — with respect to t_i is:

$$\frac{x_i - t_i}{s_i^2} - \frac{(\nu+1)t_i}{\sigma^2\nu + t_i^2} \tag{2.35}$$

The modes of the likelihood are the roots of the cubic equation

$$f_i(t) = t^3 - x_i t^2 + [\sigma^2\nu + s_i^2(\nu+1)]t - \sigma^2\nu x_i. \tag{2.36}$$

These roots can be found using the cubic formula. For simplicity, we restrict our model to the case where the cubic has only one real root (i.e. the likelihood is unimodal). We can do this by setting a lower bound for σ in terms of the s_i:

Claim: If $\sigma \geq 1.9 s_i$ then $f_i(t)$ has exactly one real root.

Proof: Let $c_i = s_i^2(\nu+1)$ and let $y = \sigma^2\nu$. We can rewrite $f_i(t)$ as

$$f_i(t) = t^3 - x_i t^2 + (y + c_i)t - x_i y. \tag{2.37}$$

It is well know that $f_i(t)$ has exactly one real root if and only if its discriminant,

$$-4(y+c_i)^3 - 4x_i^2 y - x_i^2\,(8y^2 + 20yc_i + c_i^2), \tag{2.38}$$

is negative. Since $y, z > 0$, the first two terms above are automatically negative. Thus, to make the entire discriminant negative, it is sufficient (though not necessary) to ensure that

$$8y^2 - 20yc_i - c_i^2 > 0. \tag{2.39}$$

Solving the quadratic for y, we get

$$y > \frac{5 + 3\sqrt{3}}{4}\, c_i. \tag{2.40}$$

Switching back to σ and s_i, we finally obtain the sufficient condition

$$\sigma \;>\; s_i \sqrt{\frac{5 + 3\sqrt{3}}{4} \cdot \frac{\nu + 1}{\nu}} \tag{2.41}$$

We have already constrained ν to be at least 2.5, so the coefficient of s_i in 2.41 is at most

$$\sqrt{\frac{5 + 3\sqrt{3}}{4} \cdot \frac{3.5}{2.5}} \;\approx\; 1.889. \tag{2.42}$$

Rounding up a little, we set the constant $\sigma_{min} = 1.9 \max(s_i)$. This guarantees that each of the cubics $f_i(t)$ will have a unique real root.

If we wished to remove this lower bound on σ for greater realism, we could develop a smooth formula that gave some well-defined approximation of the MLE even in the bimodal case. This would, of necessity, be inexact in some cases, because the true MLE in some cases changes discontinuous as other parameters vary, when one of the two modes passes the other; but a smooth approximation might still work well, especially when the true data-generating parameter values are sufficiently far from such discontinuities. Developing such a function, and investigating the performance of the resulting overall algorithm, is beyond the scope of this paper.

3.3 Reparametrizing the polytope $\mathcal{Y}_u$

In this appendix, we give one possible construction for the a.e.[10] smooth bijective map $m_u : \mathbb{R}^{(R-1)(C-1)} \to \mathcal{Y}_u$ mentioned in Section 3.3.2.

For notational convenience, we index the coordinates of a vector $\boldsymbol{w} \in \mathbb{R}^{(R-1)(C-1)}$ using pairs of numbers (r, c) with $1 \leq r \leq R-1$ and $1 \leq c \leq C-1$. In other words,

$$\boldsymbol{w} = (w_{1,1}, w_{1,2}, \ldots, w_{1,C-1}, w_{2,1}, \ldots, w_{R-1,C-1}).$$

Let $\dot{Y} \in \bar{\mathcal{Y}}_u$ be the matrix with coordinates

$$\dot{Y}_{rc} = \frac{n_{ur} \cdot v_{uc}}{\sum_{\tilde{r}=1}^{R} n_{u\tilde{r}}}. \tag{3.43}$$

Figure 3.3.1 shows $\dot{Y}$ for the precinct-level data from Figure 3.1.3. We think of $\dot{Y}$ as a kind of "center" for $\bar{\mathcal{Y}}_u$; its entries corresponds to the most likely voting outcomes if each voter's probability of voting for each candidate were independent of race.

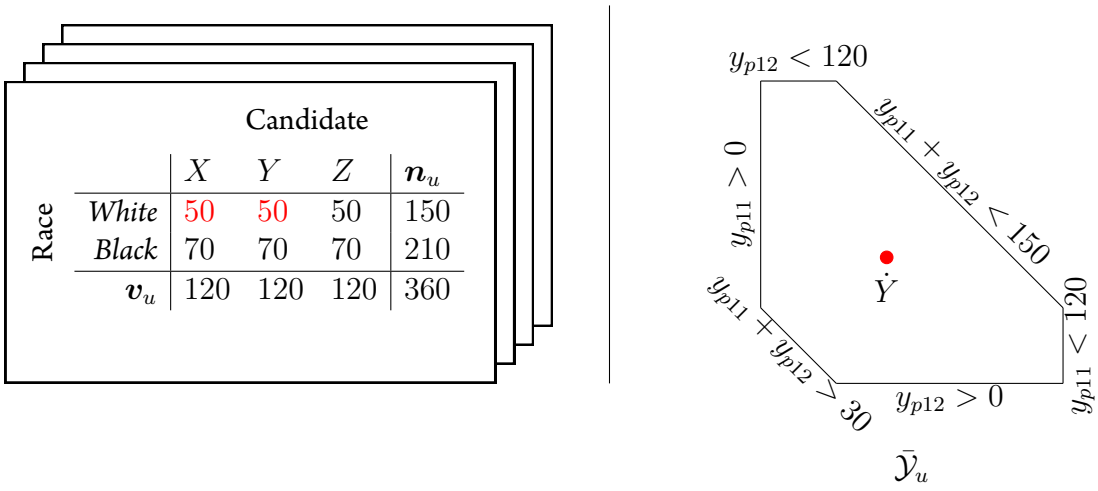
Let P_u be the $(R-1)(C-1)$-dimensional hyperplane in $\mathbb{M}_{R\times C}$ containing the polytope $\bar{\mathcal{Y}}_u$. We construct m_u in stages:

- Define a bijective affine map $a : \mathbb{R}^{(R-1)(C-1)} \to P_u$, such that $a(\mathbf{0}) = \dot{Y}$.
- Define a retraction $g : P_u \to \mathcal{Y}_u$, continuous everywhere and smooth almost everywhere, such that $g(\dot{Y}) = \dot{Y}$.
- For each $\boldsymbol{w} \in \mathbb{R}^{(R-1)(C-1)}$, let $m_u(\boldsymbol{w}) := g(a(\boldsymbol{w}))$. Note that $m_u(\mathbf{0}) = \dot{Y}$.

The maps a and g are defined as follows:

[10] It is also possible to construct a function that is smooth everywhere, but the construction is unwieldy and slows down inference unnecessarily.

Figure 3.3.1: Example of precinct-level observations and the corresponding $\dot{Y} \in \bar{\mathcal{Y}}_u$.



Race \ Candidate

	X	Y	Z	$\boldsymbol{n}_u$
White	50	50	50	150
Black	70	70	70	210
$\boldsymbol{v}_u$	120	120	120	360

- For $1 \leq r \leq R-1$ and $1 \leq c \leq C-1$, the entries of the matrix $a(\boldsymbol{w})$ are given by

$$a(\boldsymbol{w})_{rc} = w_{rc} + \dot{Y}_{rc}. \tag{3.44}$$

  The remaining entries of $a(\boldsymbol{w})$ can be filled in based on the row and column constraints that define P_u:

$$\begin{aligned} \textstyle\sum_{r=1}^{R} a(\boldsymbol{w})_{rc} &= v_{uc} \quad \text{for each } c; \\ \textstyle\sum_{c=1}^{C} a(\boldsymbol{w})_{rc} &= n_{uc} \quad \text{for each } r. \end{aligned} \tag{3.45}$$

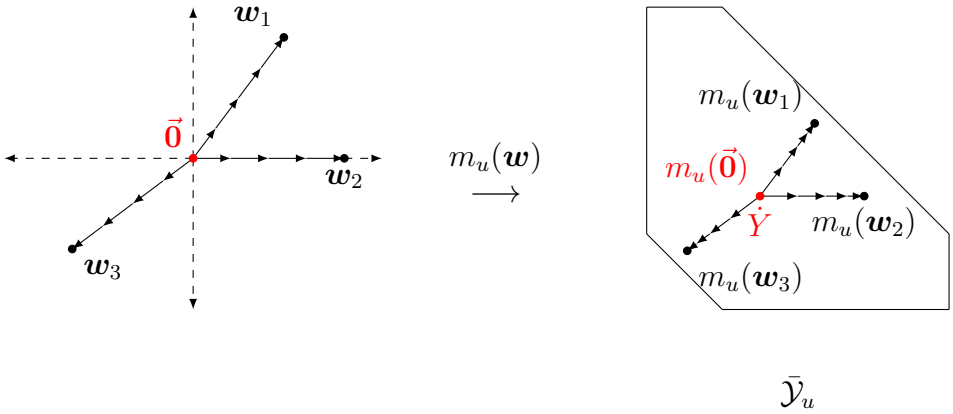
- For each matrix $M \neq \dot{Y} \in P_u$, let $b(M)$ be the intersection of the ray $\overrightarrow{\dot{Y}M}$ with the boundary of $\bar{\mathcal{Y}}_u$. Note that the function b is smooth on P_u away from a finite union of codimension-1 hyperplanes. Now let $\|\cdot\|$ be

the Euclidean norm on $\mathbb{M}_{R\times C} \simeq \mathbb{R}^{RC}$, and define

$$g(M) := \begin{cases} \dot{Y} & \text{if } M = \dot{Y}, \\ \dot{Y} + \exp\left(-\frac{\|b(M)-\dot{Y}\|}{\|M-\dot{Y}\|}\right) \cdot (b(M) - \dot{Y}) & \text{otherwise.} \end{cases} \tag{3.46}$$

Figure 3.3.2 illustrates the resulting map m_u.

Figure 3.3.2: Visualization of m_u for four input values: $\vec{\mathbf{0}}$, $\boldsymbol{w}_1$, $\boldsymbol{w}_2$, and $\boldsymbol{w}_3$.



We also need to compute the Jacobian determinant of m_u. To do this, define RC linear functions on $\boldsymbol{w}$ (corresponding to the RC possible facets of the

polytope $\bar{\mathcal{Y}}_u$) as follows:

$$\begin{aligned}
L_{rc}(\boldsymbol{w}) &= -\tfrac{w_{rc}}{\bar{Y}_{rc}} && \text{for each } 1 \le r \le R-1,\ 1 \le c \le C-1;\\
L_{rC}(\boldsymbol{w}) &= \tfrac{1}{\bar{Y}_{rC}} \textstyle\sum_{c=1}^{C-1} w_{rc} && \text{for each } 1 \le r \le R-1;\\
L_{Rc}(\boldsymbol{w}) &= \tfrac{1}{\bar{Y}_{Rc}} \textstyle\sum_{r=1}^{R-1} w_{rc} && \text{for each } 1 \le c \le C-1;\\
L_{RC}(\boldsymbol{w}) &= -\tfrac{1}{\bar{Y}_{RC}} \textstyle\sum_{r=1}^{R-1} \sum_{c=1}^{C-1} w_{rc}
\end{aligned} \tag{3.47}$$

Let $s = \max_{r,c} L_{rc}(\boldsymbol{w})$. Then it is not hard to check that

$$|J_{m_u}(\boldsymbol{w})| = \frac{e^{-(R-1)(C-1)/s}}{s^{(R-1)(C-1)+1}}. \tag{3.48}$$

3.4 Amortization

3.4.1 Amortizing $\boldsymbol{Y}^*$

Problem statement

Given:

- an $R \times C$ matrix Π with strictly positive entries whose rows sum to 1. In other words, for all $r \in \{1, \ldots, R\}$ and $c \in \{1, \ldots, C\}$,

$$\pi_{rc} > 0 \quad \text{and} \quad \sum_{c=1}^{C} \pi_{rc} = 1.$$

  We denote the r-th row of Π by $\boldsymbol{\pi}_r$.

- a positive integer n.

- a vector $\tilde{\boldsymbol{d}} = (\tilde{d}_1, \ldots, \tilde{d}_R) \in \mathbb{N}_+^R$ such that $\sum_r \tilde{d}_r = n$.

- a vector $\tilde{\boldsymbol{v}} = (\tilde{v_1}, \ldots, \tilde{v_C}) \in \mathbb{N}_+^C$ such that $\sum_c \tilde{v_c} = n$.

We wish to find an $R \times C$ matrix $\tilde{Y}$ (with rows $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r \in \mathbb{N}^C$) that maximizes the product of multinomial probabilities

$$\prod_r p(\boldsymbol{y}_r \mid \tilde{d}_r, \boldsymbol{\pi}_r),$$

subject to the constraints

$$\sum_c y_{rc} = \tilde{d}_r \text{ for each } r.$$

$$\sum_r y_{rc} = \tilde{v_c} \text{ for each } c.$$

Problem restatement using Stirling's approximation

We fix the following notation:

$$\boldsymbol{v} = \frac{\tilde{\boldsymbol{v}}}{n}, \quad \boldsymbol{d} = \frac{\tilde{\boldsymbol{d}}}{n}, \quad \tilde{Q} = \tilde{Y}/n.$$

Thus our constraints are:

$$\sum_c v_c = 1, \quad \sum_r d_r = 1,$$

$$\sum_c q_{rc} = d_r \text{ for each } r,$$

$$\sum_r q_{rc} = v_c \text{ for each } c.$$

The first-order Stirling's approximation to $k!$ is

$$\log k! = k(\log k - 1) + O(\log k),$$

where log is the natural logarithm. Ignoring the final $O(\log k)$ term, the

logarithm of the multinomial probability $p(\boldsymbol{y}_r \mid \tilde{d}_r, \boldsymbol{\pi}_r)$ can be approximated as follows:

$$\begin{aligned}
\log p(\boldsymbol{y}_r \mid \tilde{d}_r, \boldsymbol{\pi}_r) &= \log\left(\tilde{d}_r! \prod_c \frac{\pi_{rc}^{y_{rc}}}{y_{rc}!}\right) \\
&= \log(nd_r)! + \sum_c \Big(nq_{rc}\log\pi_{rc} - \log(nq_{rc})!\Big) \\
&\approx nd_r\Big(\log(nd_r) - 1\Big) + \sum_c nq_{rc}\log\pi_{rc} - \sum_c nq_{rc}\Big(\log(nq_{rc}) - 1\Big) \\
&= \sum_c nq_{rc}\log\pi_{rc} - \sum_c nq_{rc}\log\frac{q_{rc}}{d_r} \\
&= n\sum_c q_{rc}\log\left(\frac{d_r\pi_{rc}}{q_{rc}}\right),
\end{aligned}$$

where, if some $q_{rc} = 0$, we take $q_{rc}\log q_{rc}$ to equal 0.

Note that this is a (negative) multiple of $D_{\mathrm{KL}}\left(\frac{\boldsymbol{q_r}}{d_r}||\boldsymbol{\pi}_r\right)$, where $\frac{\boldsymbol{q_r}}{d_r}$ and $\boldsymbol{\pi}_r$ are seen as discrete probability distributions over {1,...,C}.

Using this approximation, we modify our goal as follows:

Restated problem: Given

- a matrix $\Pi \in M_{R\times C}$ as above;
- a vector $\boldsymbol{d} = (d_1, \ldots, d_R) \in \mathbb{R}_+^R$ such that $\sum_r d_r = 1$;
- a vector $\boldsymbol{v} = (v_1, \ldots, v_C) \in \mathbb{R}_+^C$ such that $\sum_c v_c = 1$.

For each $r \in \{1, \ldots, R\}$ and $c \in \{1, \ldots, C\}$, let

$$f_{rc}(x) = \begin{cases} x\log\left(\frac{x}{d_r\pi_{rc}}\right) & \text{if } x > 0, \\ 0 & \text{if } x = 0. \end{cases}$$

We wish to find an $R \times C$ matrix $Q = \big(q_{rc}\big)$ with nonnegative entries that

minimizes the objective function

$$f(Q) := \sum_{r,c} f_{rc}(q_{rc}),$$

subject to the linear constraints

$$g_r(Q) := \sum_c q_{rc} - d_r = 0 \quad \text{for each } r \in \{1, \ldots, R\},$$
$$h_c(Q) := \sum_r q_{rc} - v_c = 0 \quad \text{for each } c \in \{1, \ldots, C\}.$$

Analytic solution to restated problem

First we introduce some notation. For any vector $\boldsymbol{v} \in \mathbb{R}^k$, let $D_{\boldsymbol{v}}$ denote the $k \times k$ diagonal matrix with the entries of $\boldsymbol{v}$ on the diagonal.

Let $\mathcal{P}$ the set of all $R \times C$ matrices with non-negative coefficients satisfying the constraints g_r and h_c for all r and c. $\mathcal{P}$ is the $(R-1)(C-1)$-dimensional polytope in $M_{R\times C}$ on which we are trying to minimize f. In other words, we are looking for a minimum of $f_{\mathcal{P}}$, the restriction of f to $\mathcal{P}$.

Claim: The function $f_{\mathcal{P}}$ has a single global minimum: it is the unique matrix $\tilde{Q} \in \mathcal{P}$ such that

$$\tilde{Q} = D_{\boldsymbol{\alpha}} \Pi D_{\boldsymbol{\beta}}$$

for some vectors $\boldsymbol{\alpha} \in \mathbb{R}^R_+$ and $\boldsymbol{\beta} \in \mathbb{R}^C_+$. In other words, for all r and c,

$$q_{rc} = \alpha_r \pi_{rc} \beta_c.$$

(Note that while $\tilde{Q}$ is unique, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are only unique up to a multiplicative constant.)

Proof: By the method of Largrange multipliers, a point Q in the interior of $\mathcal{P}$ is an interior critical point of $f_{\mathcal{P}}$ iff

$$\nabla f(Q) = \sum_r a_r \, \nabla g_r(Q) + \sum_c b_c \, \nabla h_c(Q),$$

for some scalars $a_1, \ldots, a_R, b_1, \ldots, b_C \in \mathbb{R}$. In other words, for each r and c, we must have

$$\frac{\partial f}{\partial q_{rc}} = \frac{df_{rc}}{dq_{rc}} = \log\left(\frac{q_{rc}}{d_r \pi_{rc}}\right) + 1 = a_r + b_c.$$

Exponentiating, we obtain

$$\frac{q_{rc}}{\pi_{rc}} = \alpha_r \beta_c,$$

where $\alpha_r = d_r e^{a_r - 1}$ and $\beta_c = e^{b_c}$.

Thus $Q \in \mathcal{P}$ is a critical point of $f_{\mathcal{P}}$ iff there exist vectors $\boldsymbol{\alpha} \in \mathbb{R}_+^R$ and $\boldsymbol{\beta} \in \mathbb{R}_+^C$ such that $q_{rc} = \pi_{rc} \alpha_r \beta_c$.

Since every summand of f is strictly convex, so is f itself, and hence so is $f_{\mathcal{P}}$. Thus $f_{\mathcal{P}}$ has at most one critical point, and if this critical point exists, it is the global minimum of $f_{\mathcal{P}}$. To complete the proof of the claim, all that remains to show is $f_{\mathcal{P}}$ does indeed have a critical point in the interior of $\mathcal{P}$.

Suppose, by way of contradiction, that this is not the case. Then $f_{\mathcal{P}}$ must achieve its minimum at the boundary of $\mathcal{P}$, i.e. at some matrix Q with one or more entries equal to 0. Consider the matrix

$$Q'_\varepsilon = \varepsilon Q + (1 - \varepsilon)\boldsymbol{v}\boldsymbol{d}^T.$$

Since $\boldsymbol{v}\boldsymbol{d}^T$ is an $R \times C$ matrix with positive entries that satisfies all the constraints $g_1, \ldots, g_R, h_1 \ldots, h_C$, it is in the interior of $\mathcal{P}$. If $0 < \varepsilon < 1$, then $Q'(\varepsilon)$ must be in the interior of $\mathcal{P}$ as well. To derive a contradiction, we will now show that $f(Q'_\varepsilon) < f(Q)$ for sufficiently small ε.

Let $U = \boldsymbol{v}\boldsymbol{d}^T - Q$. Then

$$\begin{aligned}\lim_{\varepsilon\to 0}\frac{f(Q'_\varepsilon)-f(Q)}{\varepsilon} &= \lim_{\varepsilon\to 0}\frac{f(Q+\varepsilon U)-f(Q)}{\varepsilon}\\ &= \sum_{r,c}\left[\lim_{\varepsilon\to 0}\frac{f_{rc}\left(q_{rc}+\varepsilon u_{rc}\right)-f_{rc}(q_{rc})}{\varepsilon}\right]\\ &= \sum_{r,c\,:\,q_{rc}>0} u_{rc}f'_{rc}(q_{rc}) + \sum_{r,c\,:\,q_{rc}=0}\left[\lim_{\varepsilon\to 0}\frac{f_{rc}(\varepsilon u_{rc})}{\varepsilon}\right]\end{aligned}$$

Note that, if $q_{rc}=0$, then $u_{rc}=v_c d_r$, so

$$\lim_{\varepsilon\to 0}\frac{f_{rc}(\varepsilon u_{rc})}{\varepsilon} = \lim_{\varepsilon\to 0} v_c d_r \log\left(\frac{\varepsilon v_c}{\pi_{rc}}\right) = -\infty.$$

Thus we have shown that

$$\lim_{\varepsilon\to 0}\frac{f(Q'_\varepsilon)-f(Q)}{\varepsilon} = \sum_{r,c\,:\,q_{rc}>0} u_{rc}f'_{rc}(q_{rc}) + \sum_{r,c\,:\,q_{rc}=0}[-\infty] = -\infty.$$

This means that $\frac{f(Q'_\varepsilon)-f(Q)}{\varepsilon} < 0$ for small enough ε, so $f(Q'_\varepsilon) < f(Q)$, as desired. Thus a point Q on the boundary of $\mathcal{P}$ cannot be a minimum of $f_{\mathcal{P}}$.

To conclude, we have shown that $f_{\mathcal{P}}$ has a unique critical point $\tilde{Q}$ in the interior of $\mathcal{P}$, which is its global minimum on $\mathcal{P}$. Moreover, $\tilde{Q}$ is the unique point in $\mathcal{P}$ satisfying $\tilde{Q} = D_{\boldsymbol{\alpha}}\Pi D_{\boldsymbol{\beta}}$ for some $\boldsymbol{\alpha}\in\mathbb{R}^R_+$ and $\boldsymbol{\beta}\in\mathbb{R}^C_+$. Setting $\tilde{Y} = n\tilde{Q}$, we obtain an approximate solution to our original problem.

Iterative algorithm for computing $\tilde{Q}$

Given any two vectors $\boldsymbol{\alpha}\in\mathbb{R}^R_+$ and $\boldsymbol{\beta}\in\mathbb{R}^C_+$, let

$$M_{\boldsymbol{\alpha}} := \begin{pmatrix} D_{\Pi^T\boldsymbol{\alpha}} \\ \\ D_{\boldsymbol{\alpha}}\Pi \end{pmatrix} \in M_{(R+C)\times C},$$

$$M_{\boldsymbol{\beta}} := \begin{pmatrix} D_{\boldsymbol{\beta}}\Pi^T \\ \\ D_{\Pi\boldsymbol{\beta}} \end{pmatrix} \in M_{(R+C)\times R},$$

$$\boldsymbol{b} := \begin{pmatrix} v_1 \\ \vdots \\ v_C \\ d_1 \\ \vdots \\ d_R \end{pmatrix} \in \mathbb{R}_+^{R+C}.$$

Then $D_{\boldsymbol{\alpha}}\Pi D_{\boldsymbol{\beta}} \in \mathcal{P}$ if and only if $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ satisfy

$$M_{\boldsymbol{\beta}} \cdot \boldsymbol{\alpha} = M_{\boldsymbol{\alpha}} \cdot \boldsymbol{\beta} = \boldsymbol{b}.$$

All we have done here is rewritten the constraints $g_1, \ldots, g_R$ and $h_1, \ldots, h_C$ in matrix form and in terms of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

The constraints are redundant, since each of the sets $\{g_1, \ldots, g_R\}$ and $\{h_1, \ldots, h_C\}$ on its own ensures that the entries of Q sum to 1. Thus we can omit (say) the last row from $M_{\boldsymbol{\alpha}}$, $M_{\boldsymbol{\beta}}$, and $\boldsymbol{b}$ to obtain

$$M_{\boldsymbol{\alpha}} \in M_{(R+C-1)\times C},$$

$$M_{\boldsymbol{\beta}} \in M_{(R+C-1)\times R},$$

$$\tilde{\boldsymbol{b}} \in \mathbb{R}^{R+C-1},$$

such that $D_{\boldsymbol{\alpha}}\Pi D_{\boldsymbol{\beta}} \in \mathcal{P}$ if and only if $M_{\boldsymbol{\beta}} \cdot \boldsymbol{\alpha} = M_{\boldsymbol{\alpha}} \cdot \boldsymbol{\beta} = \tilde{\boldsymbol{b}}$. Note that both $M_{\boldsymbol{\alpha}}$ and $M_{\boldsymbol{\beta}}$ are now full rank.

The linear systems $M_{\boldsymbol{\beta}} \cdot \boldsymbol{\alpha} = \tilde{\boldsymbol{b}}$ and $M_{\boldsymbol{\alpha}} \cdot \boldsymbol{\beta} = \tilde{\boldsymbol{b}}$ are overconstrained and thus, in general, have no solutions. However, given $\boldsymbol{\alpha}_0$, we can find the least

squares solutions,

$$\boldsymbol{\beta}_0 = (M_{\boldsymbol{\alpha}_0}^T M_{\boldsymbol{\alpha}_0})^{-1} M_{\boldsymbol{\alpha}_0}^T \tilde{\boldsymbol{b}},$$

and then iterate for $k \geq 1$:

$$\begin{aligned} \boldsymbol{\alpha}_k &= (M_{\boldsymbol{\beta}_{k-1}}^T M_{\boldsymbol{\beta}_{k-1}})^{-1} M_{\boldsymbol{\beta}_{k-1}}^T \tilde{\boldsymbol{b}}, \\ \boldsymbol{\beta}_k &= (M_{\boldsymbol{\alpha}_k}^T M_{\boldsymbol{\alpha}_k})^{-1} M_{\boldsymbol{\alpha}_k}^T \tilde{\boldsymbol{b}}. \end{aligned}$$

This gives us a sequence of matrices

$$Q_k = D_{\boldsymbol{\alpha}_k} \Pi D_{\boldsymbol{\beta}_k},$$

with Q_k closer to $\mathcal{P}$ than Q_{k-1} (in L_2-norm). As usual with this kind of iterated projections, convergence to the fixed point $\tilde{Q}$ is guaranteed.

In practice, of course, we need to stop after a finite number of iterations. Once Q^k is close enough to $\mathcal{P}$ (i.e. within the tolerance we have chosen), we take the projection of Q^k onto $\mathcal{P}$ as the final output of the algorithm. Although the output is only an approximation to the local minimum of $f_{\mathcal{P}}$, it does satisfy the constraints $g_1, \ldots, g_R, h_1, \ldots, h_C$ exactly, as required.

3.4.2 Amortizing $\sigma_\nu^*, \boldsymbol{\nu}^*$

Recall the procedure for estimating σ_ν^* and $\boldsymbol{\nu}^*$ as given above:

1. Find $\boldsymbol{Y}^* \approx \arg\max_{\boldsymbol{Y}} \tilde{p}(\boldsymbol{\gamma}^*, \sigma_\nu = 0, \boldsymbol{\nu} = \vec{\boldsymbol{0}}, \boldsymbol{Y})$. This process is described in Appendix 3.4.1.
2. Estimate $\hat{\boldsymbol{\nu}} := \log\left(\frac{y^*_{u,r,c}}{n_{u,r}\pi^*_{r,c}}\right)$, an estimator of $\boldsymbol{\nu}$.
3. Estimate $\hat{\sigma}^2_{u,r,c} := \frac{n_{u,r} - y^*_{u,r,c}}{n_{u,r} y^*_{u,r,c}}$, an estimator of $\sigma^2_{u,r,c}$, the part of the variance of $\hat{\boldsymbol{\nu}}_{u,r,c}$ that is attributable to sampling variance in $\boldsymbol{y}_{u,r,c}$.
4. Take $max(0, \hat{\boldsymbol{\nu}}^2_{b,r,c} - \hat{\sigma}^2_{u,r,c})$, for each u, r, c, as estimates of σ_ν; average these estimates to get σ_ν^*.

5. Take ${\boldsymbol{\nu}^*}_{u,r,c} := \frac{\hat{\boldsymbol{\nu}}_{u,r,c}/\hat{\sigma}^2_{u,r,c}}{1/\hat{\sigma}^2_{u,r,c}+1/(\sigma^*_\nu)^2}$, combining the "likelihood" distribution with approximate mean and variance $[\hat{\boldsymbol{\nu}}_{u,r,c}, \hat{\sigma}^2_{u,r,c}]$ and the "prior" distribution with mean and variance $[0, \sigma^*_\nu]$, using a precision-weighted average.

6. After taking the Hessian, use one step of Newton's method on each precinct u's latent parameters to redefine $({\boldsymbol{W}^*}_u, {\boldsymbol{\nu}^*}_u)$, as explained in section 2.3.5 of chapter 2.

Here are the assumptions and approximations behind these formulas:

For step 2: we first define $\pi^*_{rc} := \frac{e^{\boldsymbol{\alpha}^*_c+\boldsymbol{\beta}^*_{r,c}}}{\sum_{c'=1}^{C} e^{\boldsymbol{\alpha}^*_{c'}+\boldsymbol{\beta}^*_{r,c'}}}$, then assume that for each u, r, the vector $(\nu_{u,r,1}, ..., \nu_{u,r,C})$ satisfies $\sum_{c=1}^{C} \pi^*_{r,c} e^{\nu_{u,r,c}} = 1$. (This assumption can be ensured, without changing the distribution of $\boldsymbol{Y}|\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, \boldsymbol{\nu}$, by shifting each $\nu_{u,r,c}$ by a constant $k_{u,r}$.)[11]

Then, let $\pi^{(\nu)}_{u,r,c} := \pi^*_{r,c} e^{\nu_{u,r,c}}$. Note that $\boldsymbol{\pi}^{(\nu)}_{u,r}$ is a probability vector. Consider $y^*_{u,r,c}$ as if it had been sampled using this vector: $\boldsymbol{y}^*_{u,r} \overset{let}{\sim} Binom(n_{u,r}, \boldsymbol{\pi}^{(\nu)}_{u,r})$. Now, under that consideration, $\hat{\nu}_{u,r,c} := \log\left(\frac{y^*_{u,r,c}}{n_{u,r}\pi^*_{r,c}}\right)$ is an estimate of $\nu_{u,r,c}$, as $E(e^{\hat{\nu}_{u,r,c}}) = e^{\nu_{u,r,c}}$.

For step 3: we use a first-order Taylor expansion of the equation of our estimator in the previous step to estimate:

$$\begin{aligned} Var_{multinomial}(\hat{\nu}_{u,r,c}|\nu_{u,r,c}) \approx & \quad Var_{multinomial}(y^*_{u,r,c}) \left(\frac{d\hat{\nu}_{u,r,c}}{dy^*_{u,r,c}}\right)^2 \\ = & \quad \pi^*_{u,r,c}(1-\pi^*_{u,r,c}) n_{u,r} \left(\frac{1}{y^*_{u,r,c}}\right)^2 \\ \approx & \quad \frac{n_{u,r} - y^*_{u,r,c}}{n_{u,r} y^*_{u,r,c}} \end{aligned} \tag{3.49}$$

For step 4, we no longer treat $\nu_{u,r,c}$ as a given, but recall that it has a distribution. Thus, since under the prior, $E(\nu_{u,r,c}) = 0$, we have

[11] Perhaps it would seem more natural to use those $k_{u,r}$ to ensure that each $\boldsymbol{\nu}_{u,r}$ is mean-0; but like that assumption, this one ensures that if any of the elements of $\boldsymbol{\nu}_{u,r}$ is nonzero, then there must be a miz of positive and negative elements.

$E(\nu^2_{u,r,c}) = Var(\hat{\nu}_{u,r,c}) = Var(\nu_{u,r,c}) + Var(\hat{\nu}_{u,r,c}|\nu_{u,r,c}) \approx \sigma^2_\nu + \hat{\sigma}^2_{u,r,c}$. So for each u, r, c, we have an estimator of σ^2_ν:

$$\sigma^2_\nu \approx \nu^2_{u,r,c} - \hat{\sigma}^2_{u,r,c}$$

Since a variance can not be negative, we set
$\sigma^*_\nu := \frac{1}{URC} \sum_{u,r,c} max(0, \nu^2_{u,r,c} - \hat{\sigma}^2_{u,r,c})$. (In practice, we replace $max(0, x)$ with the smooth function $\frac{1}{k} \log(1 + e^{kx})$, using pytorch's numerically-stabilized version of logsumexp, with $k = 100$.)

Step 5 is self-explanatory as-is, and step 6 is explained in section 2.3.5 of chapter 2.

References

[1] Voting Rights Act of 1965, 1965. 52 U.S.C § 10101 (1965).

[2] Thornburg v. Gingles, 1986. 478 U.S. 30 (1986).

[3] S. Agapiou, O. Papaspiliopoulos, D. Sanz-Alonso, and A. M. Stuart. Importance Sampling: Intrinsic Dimension and Computational Cost. *Statistical Science*, 32(3):405–431, August 2017. ISSN 0883-4237, 2168-8745. doi: 10.1214/17-STS611.

[4] Christophe Andrieu and Gareth O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, April 2009. ISSN 0090-5364. doi: 10.1214/07-AOS574.

[5] A. Apte, C. K. R. T. Jones, A. M. Stuart, and J. Voss. Data assimilation: Mathematical and statistical perspectives. *International Journal for Numerical Methods in Fluids*, 56(8):1033–1046, 2008. ISSN 1097-0363. doi: 10.1002/fld.1698.

[6] Thomas Bengtsson, Peter Bickel, and Bo Li. Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In *Institute of Mathematical Statistics Collections*, pages 316–334. Institute of Mathematical Statistics, Beachwood, Ohio, USA, 2008. ISBN 978-0-940600-74-4. doi: 10.1214/193940307000000518.

[7] Peter Bickel, Bo Li, and Thomas Bengtsson. *Sharp Failure Rates for the Bootstrap Particle Filter in High Dimensions*. Institute of Mathematical Statistics, 2008. ISBN 978-0-940600-75-1. doi: 10.1214/074921708000000228.

[8] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul

Horsfall, and Noah D. Goodman. Pyro: Deep Universal Probabilistic Programming. *arXiv preprint arXiv:1810.09538*, 2018.

[9] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep Universal Probabilistic Programming. *arXiv:1810.09538 [cs, stat]*, October 2018.

[10] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877, April 2017. ISSN 0162-1459. doi: 10.1080/01621459.2017.1285773.

[11] Olivier Cappe, Eric Moulines, and Tobias Ryden. *Inference in Hidden Markov Models*. Springer Series in Statistics. Springer, June 2009. ISBN 978-0-387-28982-3.

[12] Sourav Chatterjee and Persi Diaconis. The sample size required in importance sampling. *arXiv:1511.01437 [physics, stat]*, November 2015.

[13] Julie A. Edmunds, Lawrence Bernstein, Elizabeth Glennie, John Willse, Nina Arshavsky, Fatih Unlu, Deborah Bartz, Todd Silberman, W. David Scales, and Andrew Dallas. Preparing Students for College: The Implementation and Impact of the Early College High School Model. *Peabody Journal of Education*, 85(3):348–364, 2010. ISSN 0161-956X.

[14] Haw-ren Fang and Dianne P. O'Leary. Modified Cholesky algorithms: A catalog with new approaches. *Mathematical Programming*, 115(2): 319–349, October 2008. ISSN 0025-5610, 1436-4646. doi: 10.1007/s10107-007-0177-6.

[15] Alban Farchi and Marc Bocquet. Review article: Comparison of local particle filters and new implementations. *Nonlinear Processes in Geophysics*, 25(4):765–807, November 2018. ISSN 1023-5809. doi: https://doi.org/10.5194/npg-25-765-2018.

[16] D. A. Freedman, S. P. Klein, M. Ostland, and M. R. Roberts. Review of A Solution to the Ecological Inference Problem. *Journal of the American Statistical Association*, 93(444):1518–1522, 1998. ISSN 0162-1459. doi: 10.2307/2670067.

[17] Walter R Gilks and Carlo Berzuini. Following a moving target – Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society*, 63(1):127–146, 2001. doi: 1369-7412/01/63127.

[18] Simon Godsill and Tim Clapp. Improvement Strategies for Monte Carlo Particle Filters. In *Sequential Monte Carlo Methods in Practice*, pages 139–158. Springer-Verlag, 2000.

[19] D. James Greiner. Ecological Inference in Voting Rights Act Disputes: Where Are We Now, and Where Do We Want to Be. *Jurimetrics*, 47: 115–168, 2006.

[20] D. James Greiner and Kevin M. Quinn. RxC ecological inference: Bounds, correlations, flexibility and transparency of assumptions. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 172(1):67–81, 2009. ISSN 1467-985X. doi: 10.1111/j.1467-985X.2008.00551.x.

[21] Shaobo Han, Xuejun Liao, David B. Dunson, and Lawrence Carin. Variational Gaussian Copula Inference. *arXiv:1506.05860 [cs, stat]*, June 2015.

[22] Matthew D Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic Variational Inference. *Journal of Machine Learning Research*, 2013 (14):1303–1347, 2013.

[23] Waldemar Holubowski, Dariusz Kurzyk, and Tomasz Trawinski. A Fast Method for Computing the Inverse of Symmetric Block Arrowhead Matrices. *Applied Mathematics & Information Science*, 9(21):319–324, 2015. doi: 10.12785/amis/092L06.

[24] Christopher Jackson, Nicky Best, and Sylvia Richardson. Improving ecological inference using individual-level data. *Statistics in Medicine*, 25 (12):2136–2159, 2006. ISSN 1097-0258. doi: 10.1002/sim.2370.

[25] Jaclyn Kimble. *The Voting Rights Act, Shelby County, and Redistricting: Improving Estimates of Racially Polarized Voting in a Multiple-Election Context*. Phd, California Institute of Technology, 2015.

[26] Gary King. *A Solution to the Ecological Inference Problem: Reconstructing Individual Behavior from Aggregate Data*. Princeton University Press, 1997. ISBN 978-0-691-01240-7.

[27] Gary King, Martin A. Tanner, and Ori Rosen. *Ecological Inference: New Methodological Strategies*. Cambridge University Press, September 2004. ISBN 978-0-521-54280-7.

[28] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, December 2014.

[29] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*, December 2013.

[30] André Klima, Thomas Schlesinger, Paul W. Thurner, and Helmut Küchenhoff. Combining Aggregate Data and Exit Polls for the Estimation of Voter Transitions. *Sociological Methods & Research*, 48(2):296–325, May 2019. ISSN 0049-1241. doi: 10.1177/0049124117701477.

[31] Olivia Lau, Ryan T. Moore, and Michael Kellermann. *eiPack: Ecological Inference and Higher-Dimension Data Management*. 2019. R package version 0.1-9.

[32] X. Li, C. Li, J. Chi, J. Ouyang, and W. Wang. Black-box Expectation Propagation for Bayesian Models. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, Proceedings, pages 603–611. Society for Industrial and Applied Mathematics, May 2018. doi: 10.1137/1.9781611975321.68.

[33] Andrew C. Miller, Nicholas Foti, and Ryan P. Adams. Variational Boosting: Iteratively Refining Posterior Approximations. *arXiv:1611.06585 [cs, stat]*, November 2016.

[34] Andriy Mnih and Danilo J. Rezende. Variational inference for Monte Carlo objectives. *arXiv:1602.06725 [cs, stat]*, February 2016.

[35] M. Morzfeld, D. Hodyss, and J. Poterjoy. Variational particle smoothers and their localization. *Quarterly Journal of the Royal Meteorological Society*, 144(712):806–825, 2018. ISSN 1477-870X. doi: 10.1002/qj.3256.

[36] Matthias Morzfeld, Xin T. Tong, and Youssef M. Marzouk. Localization for MCMC: Sampling high-dimensional posterior distributions with local structure. *arXiv:1710.07747 [math, stat]*, October 2017.

[37] North Carolina State Board of Elections. North Carolina voter registration for 11/08/2016. https://dl.ncsbe.gov/?prefix=ENRS/2016_11_08/, August 2016. Metadata at https://s3.amazonaws.com/dl.ncsbe.gov/ENRS/layout_voter_stats.txt.

[38] Brendan O'Connor, Brandon Stewart, and Noah A Smith. Supplementary appendix to "Learning to Extract International Relations from Political Context" (ACL 2013). page 9, 2013.

[39] Manfred Opper and David Saad, editors. *Advanced Mean Field Methods*. Bradford Books: Neural Information Processing. The MIT Press, 2001. ISBN 978-0-262-15054-5.

[40] Jonathan Poterjoy. A Localized Particle Filter for High-Dimensional Nonlinear Systems. *Monthly Weather Review*, 144(1):59–76, October 2015. ISSN 0027-0644. doi: 10.1175/MWR-D-15-0163.1.

[41] Jonathan Poterjoy, Ryan A. Sobash, and Jeffrey L. Anderson. Convective-Scale Data Assimilation for the Weather Research and Forecasting Model Using the Local Particle Filter. *Monthly Weather Review*, 145(5):1897–1918, March 2017. ISSN 0027-0644. doi: 10.1175/MWR-D-16-0298.1.

[42] Rajesh Ranganath, Sean Gerrish, and David M. Blei. Black Box Variational Inference. *arXiv:1401.0118 [cs, stat]*, December 2013.

[43] Rajesh Ranganath, Dustin Tran, and David M. Blei. Hierarchical Variational Models. *arXiv:1511.02386 [cs, stat]*, November 2015.

[44] Patrick Rebeschini and Ramon van Handel. Can local particle filters beat the curse of dimensionality? *The Annals of Applied Probability*, 25(5): 2809–2866, October 2015. ISSN 1050-5164, 2168-8737. doi: 10.1214/14-AAP1061. 00070 MR: MR3375889.

[45] Patrick Rebeschini and Ramon van Handel. Phase transitions in nonlinear filtering. *Electronic Journal of Probability*, 20, 2015. ISSN 1083-6489. doi: 10.1214/EJP.v20-3281.

[46] Danilo Jimenez Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. May 2015.

[47] Geoffrey Roeder, Yuhuai Wu, and David Kristjanson Duvenaud. Sticking the Landing: An Asymptotically Zero-Variance Gradient Estimator for Variational Inference. *ArXiv*, abs/1703.09194, 2017.

[48] Ori Rosen, Wenxin Jiang, Gary King, and Martin A Tanner. Bayesian and frequentist inference for ecological inference: The R x C case. *Ecological inference*, page 23, 2001.

[49] François Septier and Gareth W. Peters. An Overview of Recent Advances in Monte-Carlo Methods for Bayesian Filtering in High-Dimensional Spaces. In Gareth William Peters and Tomoko Matsui, editors, *Theoretical Aspects of Spatial-Temporal Modeling*, pages 31–61. Springer Japan, Tokyo, 2015. ISBN 978-4-431-55335-9. doi: 10.1007/978-4-431-55336-6_2.

[50] Chris Snyder. Particle filters, the "optimal" proposal and high-dimensional systems. Technical report, ECMWF, Reading, UK, 2011.

[51] Chris Snyder, Thomas Bengtsson, Peter Bickel, and Jeff Anderson. Obstacles to High-Dimensional Particle Filtering. *Monthly Weather Review*, 136(12):4629–4640, December 2008. ISSN 0027-0644. doi: 10.1175/2008MWR2529.1.

[52] Cyrill Stachniss, John J. Leonard, and Sebastian Thrun. Simultaneous Localization and Mapping. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, Springer Handbooks, pages 1153–1176. Springer International Publishing, Cham, 2016. ISBN 978-3-319-32552-1. doi: 10.1007/978-3-319-32552-1_46.

[53] Dustin Tran, David Blei, and Edo M Airoldi. Copula variational inference. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3564–3572. Curran Associates, Inc., 2015.

[54] Peter Jan van Leeuwen. Particle Filtering in Geophysical Systems. *Monthly Weather Review*, 137(12):4089–4114, December 2009. ISSN 0027-0644. doi: 10.1175/2009MWR2835.1.

[55] Peter Jan Van Leeuwen, Yuan Cheng, and Sebastian Reich. *Nonlinear Data Assimilation*. Number 2 in Frontiers in Applied Dynamical Systems Reviews and Tutorials. Springer, Cham, 2015. ISBN 978-3-319-18347-3 978-3-319-18346-6.

[56] Chong Wang and David M. Blei. Variational Inference in Nonconjugate Models. *arXiv:1209.4360 [stat]*, September 2012.

[57] Xu Yaxian. *Sequential Monte Carlo Algorithms Fir High-Dimensional Filtering and Smoothing*. Thesis, April 2018.

[58] Lo-Hua Yuan, Avi Feller, and Luke W. Miratrix. Identifying and Estimating Principal Causal Effects in Multi-site Trials. *arXiv:1803.06048 [stat]*, March 2018.

[59] Cheng Zhang, Judith Butepage, Hedvig Kjellstrom, and Stephan Mandt. Advances in Variational Inference. *arXiv:1711.05597 [cs, stat]*, November 2017.

[60] Chun-Xia Zhang, Shuang Xu, and Jiang-She Zhang. A novel variational Bayesian method for variable selection in logistic regression models. *Computational Statistics & Data Analysis*, 133:1–19, May 2019. ISSN 0167-9473. doi: 10.1016/j.csda.2018.08.025.

[61] S. Zhang, Y. Liu, and X. Li. Autofocusing for Sparse Aperture ISAR Imaging Based on Joint Constraint of Sparsity and Minimum Entropy. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(3):998–1011, March 2017. ISSN 1939-1404. doi: 10.1109/JSTARS.2016.2598880.

[62] S. Zhang, Y. Liu, X. Li, and G. Bi. Variational Bayesian Sparse Signal Recovery With LSM Prior. *IEEE Access*, 5:26690–26702, 2017. ISSN 2169-3536. doi: 10.1109/ACCESS.2017.2765831.

Colophon

This thesis was typeset using LaTeX, originally developed by Leslie Lamport and based on Donald Knuth's TeX. The body text is set in 11 point Arno Pro, designed by Robert Slimbach in the style of book types from the Aldine Press in Venice, and issued by Adobe in 2007. A template, which can be used to format a PhD thesis with this look and feel, has been released under the permissive MIT (X11) license, and can be found online at github.com/suchow/ or from the author at suchow@post.harvard.edu.