Topics in Computational Bayesian Statistics With Applications to Hierarchical Models in Astronomy and Sociology

Swupnil Sahai

Submitted in partial fulfillment of the

requirements for the degree

of Doctor of Philosophy

in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2018

ABSTRACT

Topics in Computational Bayesian Statistics With Applications to Hierarchical Models in Astronomy and Sociology

Swupnil Sahai

This thesis includes three parts. The overarching theme is how to analyze structured hierarchical data, with applications to astronomy and sociology. The first part discusses how expectation propagation can be used to parallelize the computation when fitting big hierarchical bayesian models. This methodology is then used to fit a novel, nonlinear mixture model to ultraviolet radiation from various regions of the observable universe. The second part discusses how the Stan probabilistic programming language can be used to numerically integrate terms in a hierarchical bayesian model. This technique is demonstrated on supernovae data to significantly speed up convergence to the posterior distribution compared to a previous study that used a Gibbs-type sampler. The third part builds a formal latent kernel representation for aggregate relational data as a way to more robustly estimate the mixing characteristics of agents in a network. In particular, the framework is applied to sociology surveys to estimate, as a function of ego age, the age and sex composition of the personal networks of individuals in the United States.

Table of Contents

List of Figures

Acknowledgments

Firstly, I would like to thank my advisors Prof. Andrew Gelman and Prof. Tian Zheng for their continuous support during my PhD study. They have provided tremendous patience, encouragement, and wisdom in every interaction we have had over the past four years. I am the statistician I am today because of you two.

Additionally, I would like to thank Prof. Sarah Cowan and Prof. David Schiminovich for being the best collaborators and mentors I could have asked for. I learned immensely from our work together, and I feel privileged to have been able to learn more about interesting problems in sociology and astronomy. I would also like to thank Prof. Dave Blei and Prof. John Cunningham for being on my defense committee and widening my statistical perspective with your Graphical Models and Gaussian Processes courses.

I also want to express my gratitude to the entire Columbia Department of Statistics, including the faculty, staff, and my fellow graduate students. In particular, I would like to thank Yang, Tim, Jingjing, Peter, Haolei, and Nick for the great basketball memories; and Denis and Rohit for the countless late night tennis sessions. I would also like to thank Dood and Anthony for assisting me with anything and everything.

Lastly, I would like to thank my parents for working hard their entire lives and showering me with endless love so that I can pursue and achieve my dreams; Harshil for constantly motivating me to aim higher academically and in life; and Dora for being by my side through every failure and success.

To my parents, Harshil, and Dora.

Chapter 1

Introduction

Hierarchical Bayesian modeling of naturally partitioned data has become ubiquitous in social and physical sciences. For example, social surveys intrinsically have respondents belonging to different age groups, race groups, income levels, occupations, etc. Data collections in astronomy, on the other hand, typically involve distant stars and planets that belong to different subregions of the universe, such as galaxies. Hierarchical models, by partially pooling group specific coefficients (parametric cases) or functional forms (nonparametric cases), provide a natural way to regularize inferential results on such data sets, particularly when the number of hierarchical groups is large and the data models are sufficiently complex.

In this thesis, we discuss three projects related to partitioned datasets in astronomy and sociology, each constituting one chapter. In Chapter 2, which is based on [Gelman et al., 2017], we discuss distributed expectation propagation (EP) as a framework for hierarchical Bayesian modeling of partitioned data. While distributed computing has become an active area of research in recent years, such methods have had difficulty appropriately handling the prior distribution for Bayesian inference. We demonstrate, however, that EP combined with MCMC for tilted distribution approximation can be used as a natural distributed Bayesian inference tool for partitioned data. We further discuss how distributed EP provides massive computational gains over the standard

full MCMC approach, which aids in more efficiently modeling the relationship between infrared and ultraviolet radiation from distant stars.

In Chapter 3, the motivating example is an existing data set of 740 supernovae, previously analyzed in a hierarchical Bayesian model implemented with a Gibbs-type sampler in Python. We show how the Stan probabilistic programming language, combined with a No U-Turn Sampler and a transformation of the priors into an unconstrained space, allows us to estimate the expansion rate of the universe from the same data set in a much faster C++ implementation. Furthermore, we show how our probabilistic programming paradigm results in significantly less code being written for the sampler, while increasing the model's readability and mutability.

In Chapter 4 we introduce a novel latent kernel representation for aggregate relational data (ARD) as a way to more robustly estimate social mixing patterns. While previous work has allowed estimation of social mixing between discrete subgroups of a social network, our framework extends current approaches to allow estimation of social mixing between individuals of continuous characteristics, such as age. We further expand on our novel representation by allowing the kernel's bandwidth to be modeled by a continuous spline (rather than a fixed constant), which allows us to estimate, for the first time, how the age-based homophily of cross-gender social mixing changes by age. Lastly, we apply our methodology to an online survey and demonstrate how these new social mixing patterns can be estimated from ARD regarding the names and occupations of the respondents' social networks.

Partitioned data provide both opportunities and challenges for statistical analysis of astronomical and social research. We hope that the three essays here contribute to our understanding and toolbox of tackling hierarchical Bayesian modeling of partitioned data.

Chapter 2

Distributed expectation propagation for partitioned data

2.1 Introduction

Distributed algorithms are a natural approach for efficiently analyzing big data sets in a Bayesian context. In particular, expectation propagation (EP), first introduced by Minka [2001a], has shown how information from data pieces, or sites, can be iteratively processed to arrive at the posterior distribution while decreasing the overall computational effort. The classical idea of processing one data point per site, however, is just one extreme of the distributed algorithm spectrum. While this approach provides massive computational gains over processing the entire data set at once, it also suffers from a lack of information from the other data pieces as a way to regularize the results from each site's inference. As such, grouping multiple data points into each site is a half-way point that allows ample information sharing while still providing computational advantages over the full data approach.

This framework has many real world applications, particularly when the data have a naturally partitioned structure. For example, in astronomical models of the Big Bang, radiation emitted from various regions of the universe can be analyzed to

determine the age of the objects emitting such radiation. With deep spatial correlation of the radiation, it is valuable to pool the parameters of a generative radiation model together. With a fine grained map of the visible universe, however, the number of groups, and thus the number of local parameters, can be significantly large. When Bayesian methods such as MCMC are used to conduct inference on such models, the computation can become especially intractable given the large number of parameters. In such situations, splitting the groups of data and analyzing them in parallel can provide meaningful computational gains. At the same time, care must be taken to ensure that information is shared between groups to properly regularize the inference.

In this paper, we present an efficient distributed approach for hierarchical models, which by construction partition the data into conditionally separate pieces. In particular, we use the idea of EP's *cavity distribution*, which approximates the effect of inferences from all other $K-1$ data partitions, as a prior in the inference step for individual partitions. With the framework outlined, we implement an example algorithm using the Stan probabilistic programming language [Stan Development Team, 2016] for the approximation steps, and we leverage its sample-based inferences for the individual partitions.

We then demonstrate the example algorithm's effectiveness on four data sets. The first three are synthetic data sets simulated from three increasingly complex hierarchical models. The fourth data set is an actual data set from astronomy, whose modeling requires fitting a complex hierarchical mlixutre model with 9 local parameters for each of 360 groups. We show how the EP approach provides massive computational gains over the full MCMC implementation for each of these models, even when EP runs in serial. We also demonstrate that for sufficiently complex models, serial EP is outperformed by full MCMC; however, in this case distributed EP still provides massive computational gains over full MCMC. We also discover that increase in number of sites eventually leads to a drop in posterior approximation accuracy.

The remainder of the paper proceeds as follows. We first review the hierarchical EP

algorithm, demonstrate its applicability to partitioned data and hierarchical models, and discuss algorithmic considerations in Section 2.2. Section 2.3 then outlines a series of increasingly complex models fit by distributed EP to three synthetic data sets and one actual data set from astronomy, while Section 2.4 presents the computational results as well as local parameter fits to these four data sets. Lastly, Section 2.5 concludes the paper with a discussion.

2.2 Expectation propagation

Expectation propagation (EP) is an iterative algorithm in which a target density $f(\phi)$ is approximated by a density from some specified parametric family $g(\phi)$. In the following, we introduce EP in the context of hierarchical models and then discuss some algorithmic considerations related to EP's approximating steps.

2.2.1 Hierarchical framework

For this paper, the hierarchical setting we will assume is that our data are naturally separated into J groups, with local data pairs (x_{ij}, y_{ij}) and local parameter vectors α_j with P dimensions, and a global parameter vector ϕ with $2P$ dimensions. For simplicity, we will assume that that the first P elements of ϕ correspond to the prior expectations of α_j while the second P elements of ϕ correspond to the prior standard deviations of α_j. These dimensionality constraints aren't necessary for hierarchical EP in general, but we use them for consistency throughout the paper as they align with the models we fit in our experiments.

With the terminology now defined, our hierarchical model can be expressed as

$$\begin{aligned} \alpha_{jp} &\sim N(\phi_p, \exp(\phi_{p+P/2})) \\ \epsilon_{ij} &\sim N(0,1) \\ y_{ij} &= f(\alpha_j, x_{ij}, \epsilon_{ij}). \end{aligned} \tag{2.1}$$

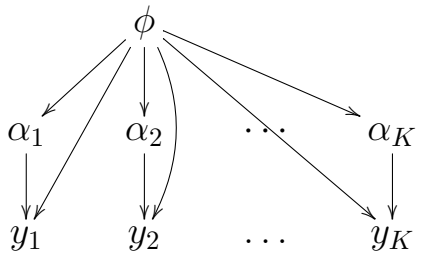

Figure 2.1: *"Model structure for the hierarchical EP frameork. In each site k, inference is based on the local model, $p(y_{(k)}|\alpha_k, \phi)p(\alpha_k|\phi)$. Computation using this site gives a distributional approximation on (α_k, ϕ) or simulation draws of (α_k, ϕ); in either case, we just use the inference for ϕ to update the local approximation $g_k(\phi)$. The algorithm has potentially large efficiency gains because, in each of the K sites, both the sample size and the number of parameters scale proportional to $1/K$."*

The joint posterior over the local parameters α_j and the global parameters ϕ is then

$$p(\phi, \alpha|y, x) \propto p(\phi) \prod_{j=1}^{J} p(\alpha_j|\phi) \prod_{i=1}^{I_j} p(y_{ij}|\alpha_j, x_{ij}), \tag{2.2}$$

and the marginal posterior of ϕ is

$$p(\phi|y, x) \propto p(\phi) \prod_{j=1}^{J} \int_{\alpha_j} p(\alpha_j|\phi) \prod_{i=1}^{I_j} p(y_{ij}|\alpha_j, x_{ij}) d\alpha_j. \tag{2.3}$$

The natural factorization here gives us the potential to learn incrementally about ϕ from each group j without having to process the data points from all J groups, which is where EP comes into play.

2.2.2 Hierarchical EP framework

Under the hierarchical context in Section 2.2.1, EP can be used to divide a problem with many parameters into subproblems with fewer parameters. This is achieved by

first shuffling and aggregating the J groups into K *sites* ($K < J$) so that

$$\begin{aligned} p(\phi|y,x) &\propto p(\phi)\prod_{k=1}^{K}\int_{\alpha_{(k)}} p(\alpha_{(k)}|\phi)\prod_{j=1}^{J_k}\prod_{i=1}^{Ij}(y_{ij}|\alpha_{(k)},x_{ij})d\alpha_{(k)} \\ &= p(\phi)\prod_{k=1}^{K}\prod_{j=1}^{J_k}\int_{\alpha_j} p(\alpha_j|\phi)\prod_{i=1}^{Ij}(y_{ij}|\alpha_j,x_{ij})d\alpha_j \\ &= p(\phi)\prod_{k=1}^{K} p(y_{(k)}|x_{(k)},\phi), \end{aligned} \tag{2.4}$$

where J_k corresponds to the number of groups in site k, $\alpha_{(k)}$ corresponds to the local parameters for those groups, and $p(y_{(k)}|x_{(k)},\phi)$ is the *likelihood factor* of site k. By distributing the hierarchical groups into separate sites, the sites can ignore the local parameters from the other groups, as each set local parameters $\alpha_{(k)}$ affects only one site.

With the above framework, then, EP aims to combine each site's likelihood factor approximation, $g_k(\phi)$, with a prior approximation $g_0(\phi)$ to create a global approximation to $p(\phi|y,x)$,

$$g(\phi) \propto g_0(\phi)\prod_{k=1}^{K} g_k(\phi). \tag{2.5}$$

The computational advantage of this approach is that the local parameters α are partitioned to create the model framework illustrated in Figure 2.1 [Gelman et al., 2017]. This is particularly important if we assume that computation costs are proportional to the sample size and number of parameters. For example, consider a model with 1 200 data points in each of 3 600 groups, 10 local parameters per group and 20 shared parameters. If we then divide the problem into $n = 3\,600$ pieces, we can reduce a $4\,320\,000 \times 36\,020$ problem to 3 000 parallel $1\,200 \times 30$ problems.

With respect to the approximating distributions $g_k(\phi)$, the standard choice is the multivariate normal family. This family is flexible enough to work for any constrained space given the appropriate transformations (e.g. logarithm, logit, etc.). For simplicity we can also reparametrize the prior $p(\phi)$ so that it, and consequently its approximation $g_0(\phi)$, is multivariate normal as well. This prolific use of multivariate

normal approximations is computationally efficient because any product or division between multivariate normals can be carried out analytically by adding and subtracting the respective natural parameters. The approximation can thus be rewritten as

$$g(\phi) \propto N(\phi|r_0, Q_0) \prod_{k=1}^{K} N(\phi|r_k, Q_k) = N(\phi|r, Q), \tag{2.6}$$

where $Q = \Sigma^{-1}$ denotes the precision matrix and $r = \Sigma^{-1}\mu$ denotes the precision mean. The idea is then to update the values of Q_k and r_k based on the data in site k until these natural parameters stabilize, hence the term expectation propagation.

2.2.3 EP algorithm

As such, in each iteration of the algorithm, and for $k = 1, \ldots, K$, we take the current approximating function $g(\phi)$ and remove the information from the site k approximation $g_k(\phi)$ to create the *cavity distribution*,

$$\begin{aligned} g_{-k}(\phi) &\propto \frac{g(\phi)}{g_k(\phi)} \\ &= N(\phi|r_0, Q_0) \prod_{k' \neq k} N(\phi|r_{k'}, Q_{k'}) = N(r_{-k}, Q_{-k}), \end{aligned} \tag{2.7}$$

where we compute the cavity's natural parameters analytically using the properties of multivariate normal multiplication,

$$Q_{-k} = Q_0 + \sum_{k' \neq k} Q_{k'}, \qquad r_{-k} = r_0 + \sum_{k' \neq k} r_{k'}. \tag{2.8}$$

With only site k's approximating information removed, we then add back in the likelihood factor for site k to create the *tilted distribution*,

$$\begin{aligned} g_{\backslash k}(\phi) &\propto g_{-k}(\phi) p(y_{(k)}|x_{(k)}, \phi) \\ &= g_{-k}(\phi) \int_{\alpha_{(k)}} p(y_{(k)}|x_{(k)}, \alpha_{(k)}, \phi) p(\alpha_{(k)}|\phi) d\alpha_{(k)} \end{aligned} \tag{2.9}$$

which is equivalent to performing inference on the model $p(y_{(k)}|x_{(k)}, \alpha_{(k)}, \phi)p(\alpha_{(k)}|\phi)$ while using the cavity distribution $g_{-k}(\phi)$ as a prior, and then integrating out the local

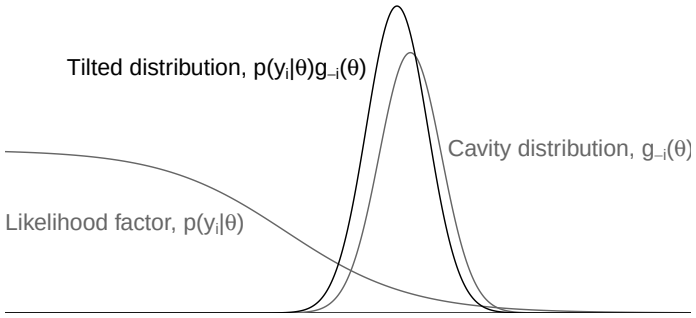

Figure 2.2: *"Example of a step of an EP algorithm in a simple one-dimensional example, illustrating the stability of the computation even when part of the likelihood is far from Gaussian. When performing inference on the likelihood factor $p(y_{(k)}|\phi)$, the algorithm uses the cavity distribution $g_{-k}(\phi)$ as a prior."*

parameters $\alpha_{(k)}$ of the site. This interpretation of the tilted distribution is illustrated graphically in Figure 2.2 [Gelman et al., 2017], which demonstrates how using the cavity distribution $g_{-k}(\phi)$'s information from all of the other $K-1$ sites as a prior allows the titled distribution's inference to spend less time on areas of low posterior mass.

The algorithm then proceeds by first constructing a normal approximation $q_{\backslash k}(\phi)$ to the tilted distribution $g_{\backslash k}(\phi)$ by matching moments,

$$g_{\backslash k}(\phi) \approx q_{\backslash k}(\phi) = N(\phi|r_{\backslash k}, Q_{\backslash k}). \tag{2.10}$$

The tilted distribution's normal approximation $q_{\backslash k}(\phi)$ then assists in creating the new site approximation by removing the information from the other $k' \neq k$ sites and prior,

$$g_k^{new} \approx \frac{q_{\backslash k}(\phi)}{g_{-k}(\phi)} = N(\phi|r_k^{new}, Q_k^{new}), \tag{2.11}$$

where the new site approximation's natural parameters are computed by

$$Q_k^{new} = Q_{\backslash k} - Q_{-k}, \qquad r_k^{new} = r_{\backslash k} - r_{-k}. \tag{2.12}$$

At the end of each iteration, after having conducted inference on all k sites, the global approximation updates to

$$g^{new}(\phi) = p(\phi) \prod_k g_k^{new}(\phi) = N(\phi | r^{\text{new}}, Q^{\text{new}}), \tag{2.13}$$

where the natural parameters of the new global approximation are calculated as

$$Q^{\text{new}} = Q_0 + \sum_k Q_k^{\text{new}}, \qquad r^{\text{new}} = r_0 + \sum_k r_k^{new}. \tag{2.14}$$

Iterating the site updates in sequence or in parallel thus gives the following algorithm [Gelman et al., 2017].

General expectation propagation algorithm

1. Choose initial site approximations $g_k(\phi)$.
2. Repeat for $k \in \{1, 2, \ldots, K\}$ (in serial or parallel batches) until all site approximations $g_k(\phi)$ converge:
   (a) Compute the cavity distribution, $g_{-k}(\phi) \propto g(\phi)/g_k(\phi)$.
   (b) Compute the tilted distribution, $g_{\backslash k}(\phi) \propto g_{-k}(\phi) p(y_{(k)}|\phi)$
   (c) Update site approximation $g_k(\phi)$ so that $g_k(\phi) g_{-k}(\phi)$ approximates $g_{\backslash k}(\phi)$.

2.2.4 Approximating the tilted distribution

In EP, if we use a multivariate normal family for the site approximations as described previously, the tilted distribution approximation in step 2c can be achieved by matching the first and second moments of $g_k(\phi) g_{-k}(\phi)$ to those of the possibly intractable tilted distribution $g_{\backslash k}(\phi)$. This corresponds to minimizing the Kullback-Leibler divergence $\text{KL}(g_{\backslash k}(\phi) || q(\phi))$. For low dimension problems, this moment matching can be performed analytically [e.g. Opper and Winther, 2000, Minka, 2001b] or reasonably

quickly using quadrature [e.g. Zoeter and Heskes, 2005]. In higher dimensions, however, analytic solutions are generally nonexistent and quadrature error becomes intractable.

As such, for higher dimensions, feasible alternatives to approximating the titled distribution include matching the mode via Laplace methods, minimizing the reverse KL divergence via variational inference, and using numerical simulations via Monte Carlo [Gelman et al., 2017]. While Laplacian methods offer numerous computational advantages and variational inference offers guarantees about the global divergence minimization, we implement the simulation based approach in this paper as a proof of concept to demonstrate how beneficial distributed EP can be for the most demanding approximation algorithm.

Under the MCMC framework, simulations are used to sample from the tilted distribution at each step and set the moments of the approximating family. Specifically, this is accomplished by sampling from the joint tilted distribution in (2.9) (i.e. without marginalizing the local parameters),

$$g_{\backslash k}(\phi, \alpha_{(k)}) = g_{-k}(\phi) p(y_{(k)}|x_{(k)}, \alpha_{(k)}, \phi) p(\alpha_{(k)}|\phi), \tag{2.15}$$

which is equivalent to doing inference on the model $p(y_{(k)}|x_{(k)}, a_{(k)}, \phi)p(a_{(k)}|\phi)$ while using $g_{-k}(\phi)$ as a prior. The MCMC samples from the joint distribution can then be approximated by a multivariate normal,

$$g_{\backslash k}(\phi, \mathbf{a}_{(k)}) \approx q(\phi, \mathbf{a}_{(k)}) = \text{Normal}(\phi, \alpha_{(k)}|r^*_{\backslash k}|Q^*_{\backslash k}), \tag{2.16}$$

at which point it is trivial to integrate out $\alpha_{(k)}$ and arrive at the marginal distribution $q_{\backslash k}(\phi)$ in (2.10).

The advantage of combining this approach with parallel EP is that the tilted approximation sampling only uses a fraction $1/K$ of the data (and parameters) per site. Since MCMC computations scale linearly in the number of data points, and superlinearly in the number of parameters, this is a huge computational advantage.

2.2.5 Keeping the covariance matrix positive definite

In Equation 2.12, the last update step in each EP iteration, the approximated natural parameters $Q_{\backslash k}$ and $r_{\backslash k}$ of the tilted distribution are used together with the parameters Q_{-k}^{new} and r_{-k}^{new} of the cavity distribution to determine the new site approximation parameters Q_k^{new} and r_k^{new}. Since the difference between the two positive definite matrices is not itself necessarily positive definite, there are situations in which the site approximation can be improper; this effectively breaks the EP algorithm.

The solution we propose for this problem is to smooth the tilted distribution's natural parameters with those of the cavity distribution. This can be achieved by taking a smoothing factor $\delta \in [0, 1]$ and creating

$$Q_k^n = (1 - \delta^n) \cdot Q_k^{\text{old}} + \delta^n \cdot Q_k^{\text{new}}, \qquad r_k^n = (1 - \delta^n) \cdot r_k^{\text{old}} + \delta^n \cdot r_k^{\text{new}},$$

for $n \in \mathbb{N}$. One can then keep increasing n until either $(Q_k^n)^{-1}$ is positive definite (in which case Q_k^n and r_k^n become the new approximation) or δ^n is smaller than some threshold ϵ (in which case we discard this site's update and keep the approximation the same as it was before).

2.3 Experiments with partitioned data and hierarchical models

We apply the EP algorithm discussed in Section 2.2 to four different data sets, using a different hierarchical model for each data set. The first three data sets are simulated from known parameters while the fourth is an actual data set from astronomy, whose fitting involves developing a novel hierarchical mixture model with interpretable parameters. All MCMC runs, for either each of the EP sites or the full MCMC computation, use 4 chains with 1000 iterations, of which half are discarded as warmup.

2.3.1 Simulated hierarchical linear regression

The first data set is simulated from a hierarchical linear model, our simplest example. Letting

$$a_j = [\beta_{0j}, \beta_{1j}, \log \sigma_j]^T$$

denote the local parameters for each group j, we simulate $J = 50$ groups, with $I = 500$ data points per group, from

$$\begin{aligned} a_{jp} &\sim N(\phi_p, e^{\phi_{p+3}}) \\ \epsilon_{ij} &\sim \text{Normal}(0, 1) \\ y_{ij} &= \beta_{0j} + \beta_{1j} \cdot x_{ij} + \sigma_j \cdot \epsilon_{ij}, \end{aligned} \tag{2.17}$$

where β_{0j}, β_{1j}, and σ_j correspond to the intercept, slope, and noise standard deviation for each group j; ϕ corresponds to the global parameter vector; and x_{ij} are sampled from an arbitrary uniform distribution.

This problem has $3 \cdot 2 = 6$ shared parameters, $3 \cdot J = 150$ local parameters, and a total of $I \cdot J = 25,000$ samples. Our implementation uses R for the message passing framework and the Stan probabilistic modeling language [Stan Development Team, 2016] for MCMC sampling from the tilted distribution. We fit the hierarchical linear model with various EP settings, partitioning the data into $K = 5, 10, 25$ sites and running EP in both serial and parallel for each K. Uniform distributions are used as the initial site approximations, while a smoothing factor of $\delta = 0.9$ is used in order to positive definiteness for each site approximation. We compare the results from the serial and distributed EP approximations to an MCMC approximation for the full model using Stan.

2.3.2 Simulated hierarchical logistic regression

The next data set is simulated from a generalized inverse logistic (sigmoid) model. Letting

$$a_j = [\beta_{0j}, \beta_{1j}, \log \sigma_j]^T$$

denote the local parameters for each group j, we simulate $J = 50$ groups, with $I = 500$ data points per group, from

$$\begin{aligned} a_{jp} &\sim N(\phi_p, e^{\phi_{p+5}}) \\ \epsilon_{ij} &\sim \text{Normal}(0, 1) \\ y_{ij} &= \beta_{0j} + \beta_{1j} \cdot \sigma\left(\frac{x_{ij} - \mu_{1j}}{\sigma_{1j}}\right) + \sigma_j \cdot \epsilon_{ij}, \end{aligned} \tag{2.18}$$

where $\sigma(\cdot) = \text{logit}^{-1}(\cdot)$; and β_{1j}, μ_{1j}, and σ_{1j} correspond to the maximal height, inflection point location, and (inverse) inflection point slope of the sigmoid in each group j.

This problem has $5 \cdot 2 = 10$ shared parameters, $5 \cdot J = 250$ local parameters, and a total of $I \cdot J = 25,000$ samples. As before, we use R and Stan for our implementation; we partition the data into $K = 5, 10, 25$ sites; and we compare the results from the EP approximations to an MCMC approximation for the full model using Stan.

2.3.3 Simulated hierarchical logistic regression with Gaussian dip

To add additional nonlinearities, for our last simulated data set we take the hierarchical sigmoid in Section 2.3.2 and multiply it by an inverted Gaussian, creating a "dip" in the regression curve. Letting

$$a_j = [\beta_{0j}, \beta_{1j}, \mu_{1j}, \log \sigma_{1j}, \sigma^{-1}(\beta_{2j}), \mu_{2j}, \log \sigma_{2j}, \log \sigma_j]^T$$

denote the local parameters for each group j, we simulate $J = 50$ groups, with $I = 500$ data points per group, from

$$
\begin{aligned}
a_{jp} &\sim N(\phi_p, e^{\phi_{p+8}}) \\
\epsilon_{ij} &\sim \text{Normal}(0, 1) \\
y_{ij} &= \beta_{0j} + \beta_{1j}\sigma\left(\frac{\log x_{ij} - \mu_{1j}}{\sigma_{1j}}\right) \cdot \left(1 - \beta_{2j}\exp\left(-\frac{1}{2}\left(\frac{\log x_{ij} - \mu_{2j}}{\sigma_{2j}}\right)^2\right)\right) + \sigma_j \cdot \epsilon_{ij},
\end{aligned} \tag{2.19}
$$

where μ_{2j} and σ_{2j} correspond to the center and scale of the inverted Gaussian, while $\beta_{2j} \in [0, 1]$ corresponds to the proportion by which the curve dips at the center of the Gaussian.

This problem has $8 \cdot 2 = 16$ shared parameters, $8 \cdot J = 400$ local parameters, and a total of $I \cdot J = 25,000$ samples. As before, we use R and Stan for our implementation; we partition the data into $K = 5, 10, 25$ sites; and we compare the results from the EP approximations to an MCMC approximation for the full model using Stan.

2.3.4 Galactic ultraviolet data

Lastly, we demonstrate the EP algorithm applied to an actual data set in astronomy. The goal of our inference is to model the nonlinear relationship between diffuse Galactic far ultraviolet radiation (FUV) and 100-μm infrared emission (i100) in various regions of the observable universe. Data is collected from the Galaxy Evolution Explorer telescope. It has been shown that there is a linear relationship between FUV and i100 below i100 values of 8 MJy sr^{-1} [Hamden et al., 2013]. Here we attempt to model this relationship across the entire range of i100 values.

Figure 2.3 shows scatterplots of FUV versus i100 in different longitudinal regions (each of width 1 degree) of the observable universe. The bifurcation in the scatterplots for i100 values greater than 8 MJy sr^{-1} suggests a non-linear mixture model is necessary to capture the relationship between the two variables. At the same time, a flexible parametric model is desired to handle the various mixture shapes, while maintaining interpretability in the parameters.

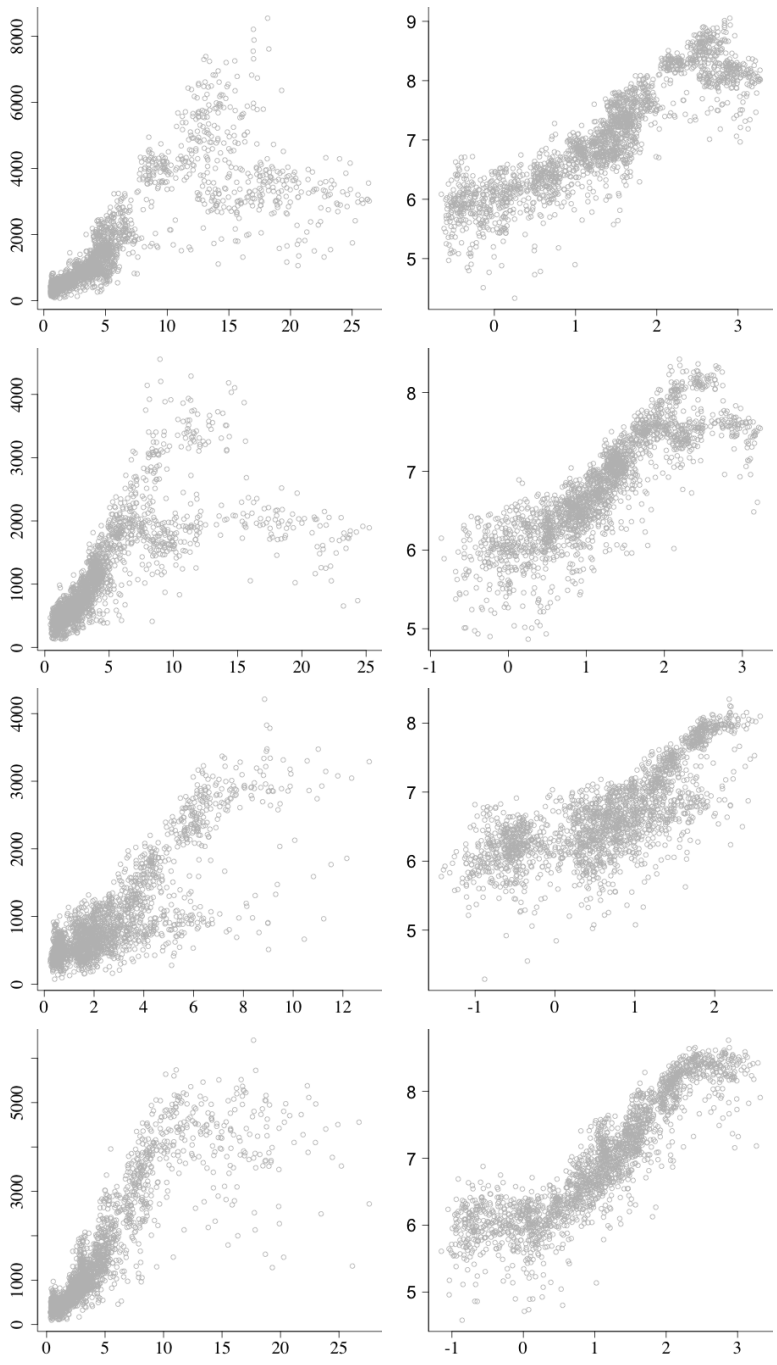

Figure 2.3: Scatterplots of ultraviolet radiation (FUV) versus infrared radiation (i100) in various regions of the universe. Data are shown for regions of longitude $12°, 23°, 92°$, and $337°$, and are presented with axes on the original scale (first column) and on the log scale (second column).

Letting

$$a_j = [\beta_{0j}, \beta_{1j}, \mu_{1j}, \log \sigma_{1j}, \sigma^{-1}(\beta_{2j}), \mu_{2j}, \log \sigma_{2j}, \sigma^{-1}(\pi_j), \log \sigma_j]^T$$

denote the local parameters for each group j, we model the top half of the bifurcation (i.e. the first component of the mixture) as a generalized inverse logistic function (as in Section 2.3.2),

$$f(a_j, x_{ij}) = \beta_{0j} + \beta_{1j}\sigma\left(\frac{\log x_{ij} - \mu_{1j}}{\sigma_{1j}}\right),$$

while the second mixture component is modeled as the same inverse logistic function multiplied by an inverted Gaussian (as in Section 2.3.3),

$$g(a_j, x_{ij}) = \beta_{0j} + \beta_{1j}\sigma\left(\frac{\log x_{ij} - \mu_{1j}}{\sigma_{1j}}\right) \cdot \left(1 - \beta_{2j}\exp\left(-\frac{1}{2}\left(\frac{\log x_{ij} - \mu_{2j}}{\sigma_{2j}}\right)^2\right)\right).$$

As such, the ultraviolet radiation (y_{ij}) is modeled as a function of infrared radiation (x_{ij}) through the following mixture model:

$$\begin{aligned} a_{jp} &\sim N(\phi_p, e^{\phi_{p+9}}) \\ \epsilon_{ij} &\sim N(0,1) \\ \log y_{ij} &= \pi_j \cdot f(a_j, x_{ij}) + (1-\pi_j) \cdot g(a_j, x_{ij}) + \sigma_j \epsilon_{ij}, \end{aligned} \tag{2.20}$$

where $\pi_j \in [0,1]$ corresponds to the proportion of data generated by the first mixture.

This problem has $9 \cdot 2 = 18$ shared parameters of interest. The number of local parameters, however, depends on how finely we split the data in the observable universe. Our study in particular is constructed with $J = 360$ hierarchical groups (one for each longitudinal degree of width one degree), resulting in a total of $9 \cdot J = 3,240$ local parameters. We also sample the number of observations per group as $I = 2,000$, resulting in a total of $I \cdot J = 720,000$ samples. As for the simulated data, we use R and Stan for our implementation, and we compare the results from the EP approximations to an MCMC approximation for the full model using Stan. However, we partition the data into $K = 5, 10, 30$ sites because these values divide the $J = 360$ groups cleanly.

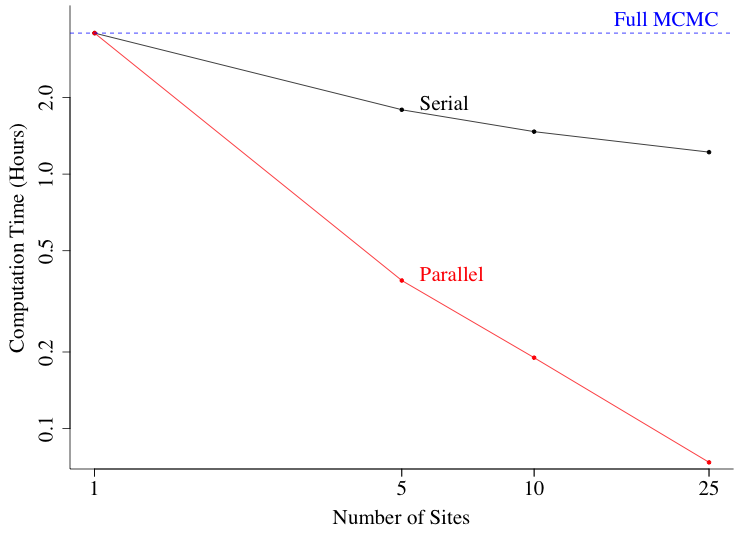

Figure 2.4: Computation times for the distributed EP algorithm applied to the simulated linear data set, as a function of the number of sites. The full MCMC computation time is equivalent to that of EP with $K = 1$ site.

2.4 Results

We present the results of running EP for the four data sets described in Section 2.3.

2.4.1 Simulated hierarchical linear regression

Figure 2.4 illustrates the computation times for the EP runs with both serial and parallel updates, applied to data simulated from the hierarchical linear model. Because the model is quite simple, serial EP alone gives serious computational advantages. With K = 10 sites, for example, EP provides a 59% decrease in computation time. The advantages of distributed EP are clear as well, with K = 10 sites resulting in an overall 95% decrease in computation time. Splitting the computation across K = 25 sites, however, does not provide much additional computational advantages.

Figure 2.5 shows a comparison of the local scatterplot fits for each EP setting on various hierarchical groups. All of the runs show similar results for all groups, which

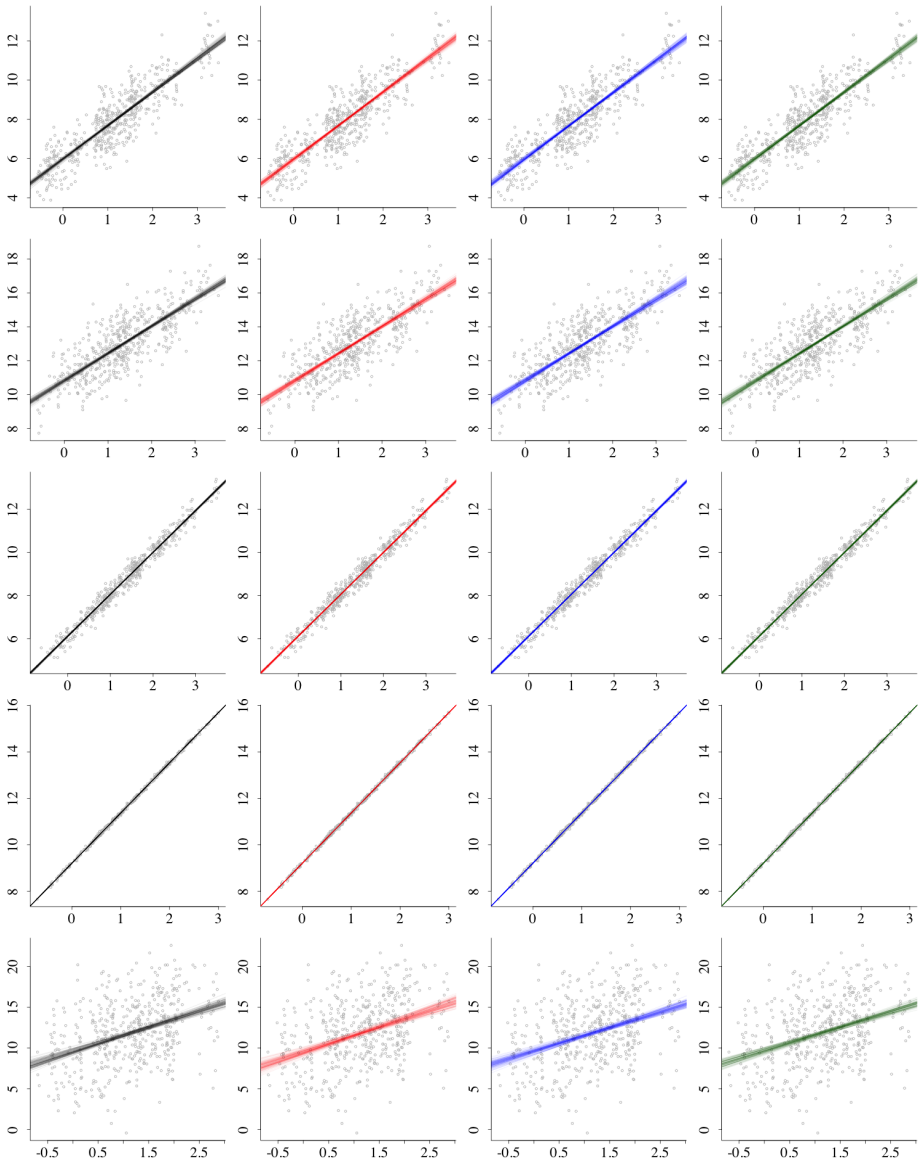

Figure 2.5: Comparison of the local fits of the full MCMC computation (black) for the hierarchical linear model and the final distributed EP approximations when the groups are distributed into $K = 5$ (red), $K = 10$ (blue), and $K = 30$ (green) sites. Posterior draws are shown for each of 5 groups (one group per row) with $j = 19, 22, 26, 45$, and 47.

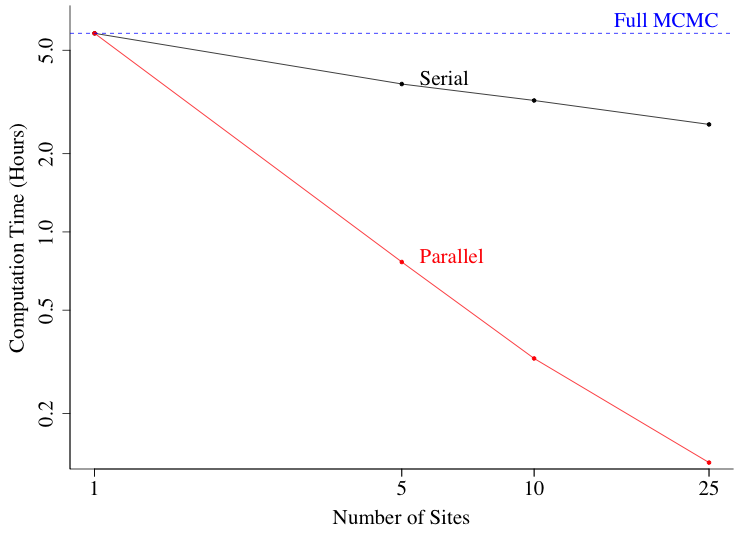

Figure 2.6: Computation times for the distributed EP algorithm applied to the simulated sigmoid data set, as a function of the number of sites. The full MCMC computation time is equivalent to that of EP with $K = 1$ site.

is somewhat expected for a model this simple.

2.4.2 Simulated hierarchical logistic regression

Figure 2.6 illustrates the computation times for the EP runs with both serial and parallel updates, applied to data simulated from the hierarchical sigmoid model. Compared to the linear model results in Figure 2.4, serial EP doesn't provide as large computation advantages. With K = 10 sites, for example, EP provides a 45% decrease in computation time. The advantages of distributed EP, however, are as pronounced, with K = 10 sites creating an overall 94% decrease in computation time over the full MCMC approach. Once again, splitting the computation across K = 25 sites does not provide much additional computational advantages.

Figure 2.7 shows a comparison of the local scatterplot fits for each EP setting on various hierarchical groups. As with the linear model in Figure 2.5, all of the runs show similar results across all groups, which is expected for a model this simple. Models

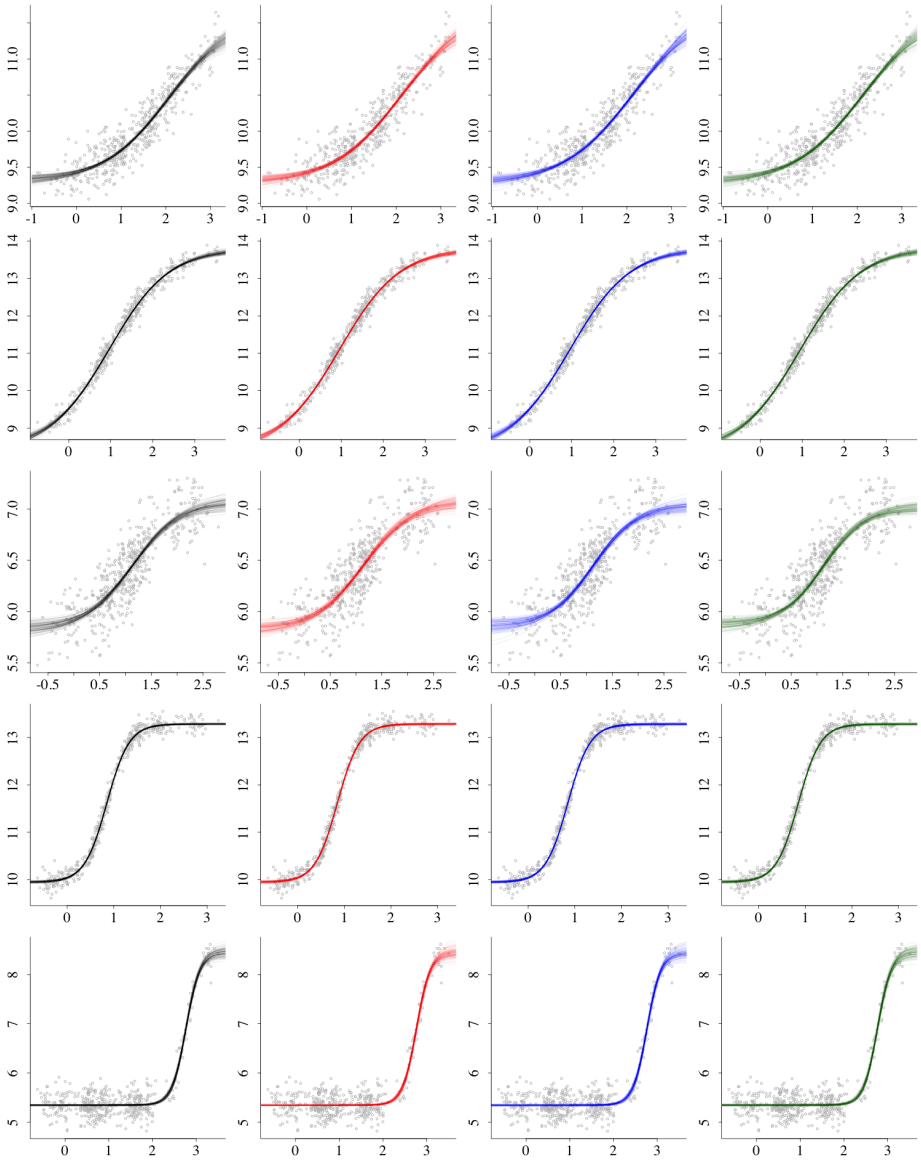

Figure 2.7: Comparison of the local fits of the full MCMC computation (black) for the hierarchical sigmoid model and the final distributed EP approximations when the groups are distributed into $K = 5$ (red), $K = 10$ (blue), and $K = 30$ (green) sites. Posterior draws are shown for each of 5 groups (one group per row) with $j = 25, 27, 42, 48$, and 49.

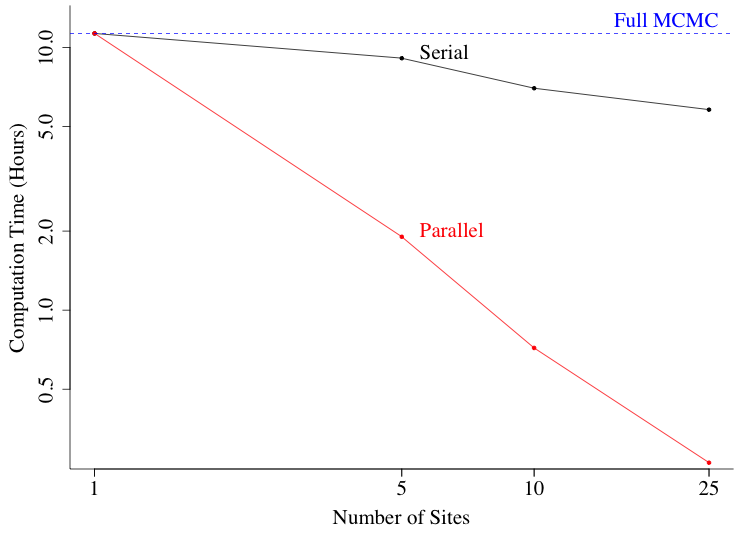

Figure 2.8: Computation times for the distributed EP algorithm applied to the simulated sigmoid with dip data set, as a function of the number of sites. The full MCMC computation time is equivalent to that of EP with $K = 1$ site.

involving more nonlinearities, however, would be expected to take some performance hits as the number of sites is increased.

2.4.3 Simulated hierarchical logistic regression with dip

Figure 2.8 illustrates the computation times for the EP runs with both serial and parallel updates, applied to data simulated from the hierarchical sigmoid with dip model. Compared to the linear and sigmoid model results in Figures 2.4 and 2.6, serial EP provides even fewer computation advantages. With K = 10 sites, EP provides only a 38% decrease in computation time. The advantages of distributed EP, however, are just as pronounced, with K = 10 sites creating an overall 94% decrease in computation time over the full MCMC approach. Once again, splitting the computation across K = 25 sites does not provide much additional computational advantages.

Figure 2.9 shows a comparison of the local scatterplot fits for each EP setting on various hierarchical groups. While most of the runs show similar results across

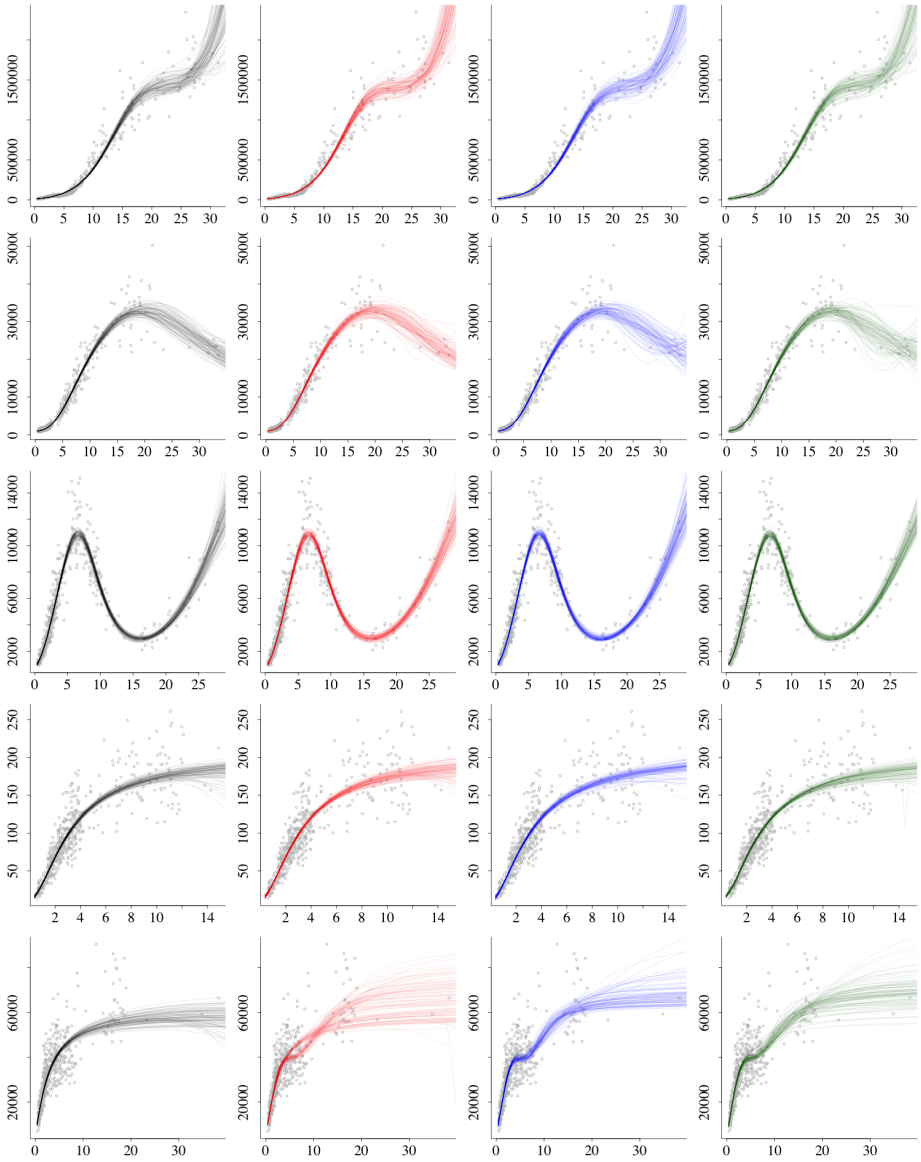

Figure 2.9: Comparison of the local fits of the full MCMC computation (black) for the hierarchical sigmoid with dip model and the final distributed EP approximations when the groups are distributed into $K = 5$ (red), $K = 10$ (blue), and $K = 30$ (green) sites. Posterior draws are shown for each of 5 groups (one group per row) with $j = 1, 3, 33, 42$, and 48.

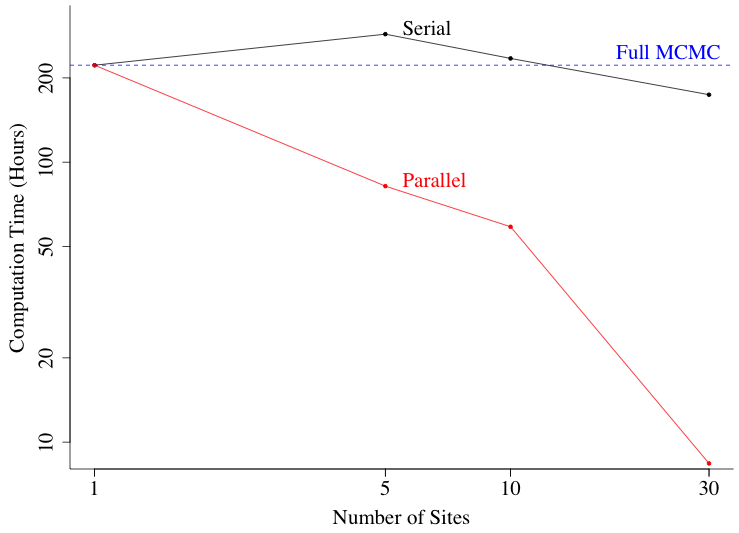

Figure 2.10: Computation times for the distributed EP algorithm applied to the astronomy data set, as a function of the number of sites. The full MCMC computation time is equivalent to that of EP with $K = 1$ site.

all groups, this is the first model were we begin to see some of the limitations of distributed EP. Namely, for group 48 (the fifth column), the run with $K = 5$ sites arrives at a different posterior distribution over the local parameters compared to the full MCMC run. This limitation is something we would expect to see every so often for more complex nonlinear models such as the one experimented with here.

2.4.4 Galactic ultraviolet data

Figure 2.10 illustrates the computation times for the EP runs with both serial and parallel updates, applied to the real astronomy data. Unlike the simulated data results in Figures 2.4, 2.6, and 2.8, the hierarchical mixture model is so complex that the serial EP computation is slower than the full MCMC computation for K = 5 and 10 sites. As such, the advantages of distributed EP are more important, as seen by K = 10 sites resulting in a 74% decrease in computation time over full MCMC. This advantage in computation time, however, depends on the implementation of the

parallelization. By using the time spent on the sampling of the tilted distribution as our benchmarking criterion, we can focus on the crucial part of the algorithm and neglect the implementation-specific factor.

Figure 2.11 shows a comparison of the local scatterplot fits for each EP setting on various hierarchical groups, each representing a one-degree longitudinal slice of the observable universe. While all of the runs show similar results for most groups, there are some cases where increasing the number of sites results in poorer performance. In particular, EP with 30 sites converges to a different mixture for 82°, while EP with 10 sites converges to a different mixture for 194°.

2.5 Discussion

This paper presents EP as a framework for distributed Bayesian inference of hierarchical models on partitioned data sets by using the principle of message passing with cavity and tilted distributions. We create an example EP algorithm that uses Stan [Stan Development Team, 2016] for the tilted distribution approximation, and we demonstrate on four data sets that distributed EP provides significant computational gains over the full MCMC approach.

In using EP for posterior inference, we assume that convergence will occur within a few iterations. For our three synthetic data sets and actual astronomy data set, this assumption is not violated. Convergence in general, however, is not guaranteed for EP. Additionally, we approach the inference problem in an exhaustive manner, trying various combinations of the number of sites K, number of data points per site N_k, and using a fixed precision smoothing value $\delta = 0.9$ throughout. In practice, it is not efficient to take such an exhaustive approach, as the computational cost of doing so far outweighs that of simply running full MCMC in the first place. Additional research is still required in order to determine efficient approaches to setting the optimal number of sites, smoothing value, etc.

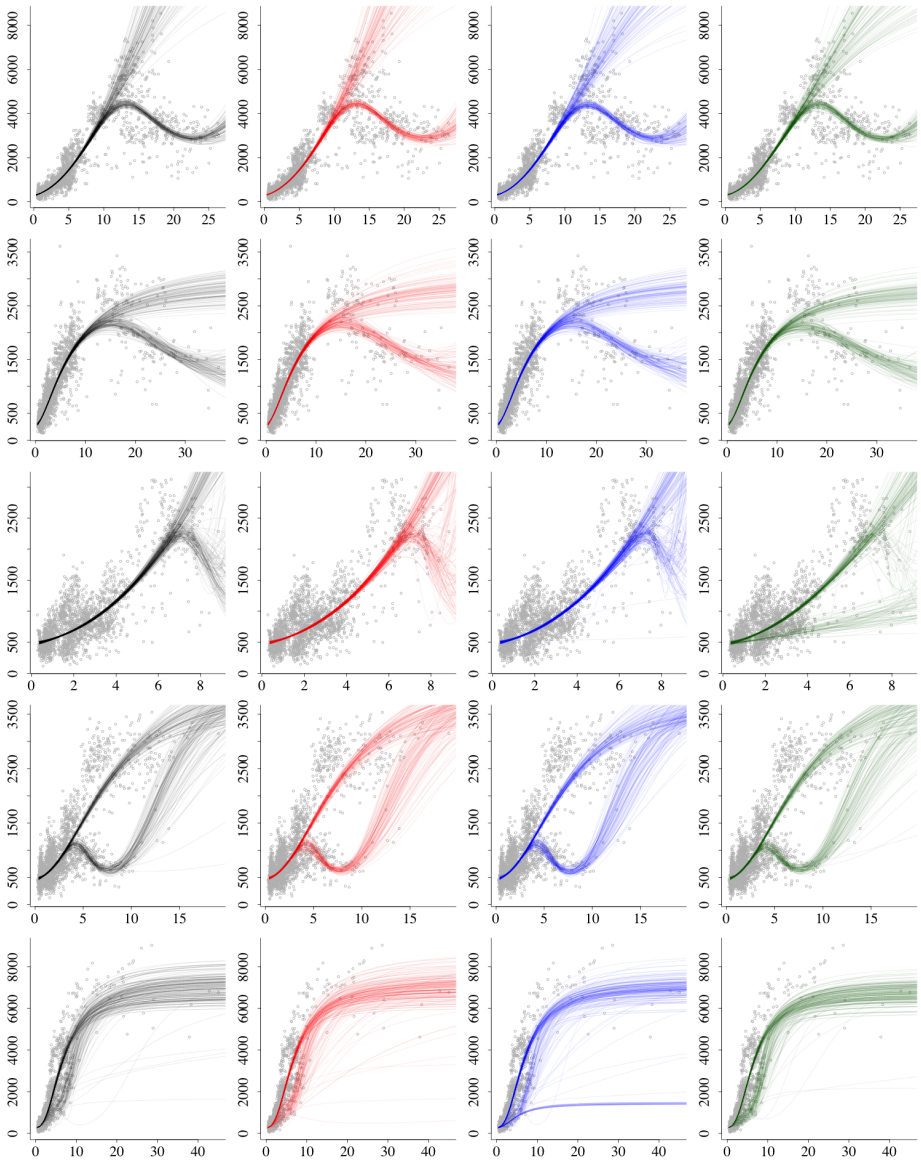

Figure 2.11: Comparison of the local fits of the full MCMC computation (black) for the astronomy example and the final distributed EP approximations when the groups are distributed into $K = 5$ (red), $K = 10$ (blue), and $K = 30$ (green) sites. Posterior draws are shown for longitudes $12°, 32°, 82°, 93°$, and $194°$ (one per row).

Aside from finding a more efficient way to determine optimal tuning parameters for our particular implementation, we also believe the algorithm itself has room for improvement. In approximating the tilted distribution we assume that the tilted distribution is multivariate normal, which guides our estimation of its precision matrix. In the general case when the titled distribution is not normal, however, care must be taken to shrink the eigenvalues of the sample covariance matrix or impose sparse structure constraints on it [Bodnar et al., 2014, Friedman et al., 2008].

From the computational perspective, variations of the fully distributed EP approach are also worth exploring. For example, one could process the sites in parallel, but asynchronously, so that EP can move to the next iteration even faster. While this may decrease the amount of information available as a prior in the next iteration, the computational gains of the asynchronous approach may outweigh the costs in accuracy.

Chapter 3

Ordinary differential equation integration for dark energy estimation

3.1 Introduction

Supernovae, the astronomical events at the end of a massive star's life marked by a catastrophic explosion, have tremendous potential in helping researchers learn instrinsic properties about the entire universe. In particular, supernovae of type Ia (SNe Ia) have been instrumental in establishing the accelerated expansion of the universe, starting with the unexpected discovery by the Supernova Cosmology Project [Riess et al., 1998] and the High-z Supernova Search Team [Perlmutter et al., 1999]. Because SNe Ia emit radiation that probes the low-redshift (i.e. less visible) universe, they are ideal tools to measure the properties of "dark energy", a form of energy that permeates all of space and tends to accelerate the expansion of the universe. In the last decade, the sample of SNe Ia has increased dramatically (e.g., Wood-Vasey et al. [2007]; Kowalski et al. [2008]; Kessler et al. [2009]; Contreras et al. [2010]; Suzuki et al. [2012]; Betoule et al. [2014]), and it now comprises several hundred spectroscopically

confirmed SNe Ia.

SNe Ia occur when material from a companion star accreting onto a white dwarf star triggers carbon fusion, proceeding until a core of typical mass $0.7M_{\odot}$ of ^{56}Ni is created (the mass of our sun is $1M_{\odot} = 1.99 \times 10^{30}$kg). After a type Ia supernova's explosion, its luminosity can be observed over time, creating a light curve (LC). Within the more restricted subclass of so-called "normal" SNe Ia, the fundamental assumption underlying their use to measure expansion history is that they can be standardized so that their peak luminosity magnitude (in the LC) are sufficiently homogeneous. The relative uniformity of these intrinsic magnitudes allows us to measure the distance of the SNe Ia from their host galaxy (known as distance modulus) because a type Ia supernova's observed peak magnitude depends primarily on its intrinsic magnitude and its distance modulus.

A supernova's distance modulus is of utmost importance to estimate because it is a function of integrated luminosity, an integral whose value is determined by both the supernova's specific redshift and a global set of cosmological parameters (one of which is the dark energy component). One of the most widely used frameworks for estimating the distance modulus from LC data is the SALT2 method [Guy et al., 2005], which derives color and stretch corrections for the magnitude from the LC fit, and then uses the corrected distance modulus to fit the underlying cosmological parameters.

Recent work has applied the SALT2 methodology to a larger catalog of SNe Ia. March et al. [2011] demonstrated with simulated data that a Bayesian hierarchical model has a reduced posterior uncertainty, smaller mean squared error, and better coverage properties than the standard approach. Betoule et al. [2014] then reanalyzed 740 spectroscopically confirmed SNe Ia obtained by the SDSS-II and SNLS collaboration, known as the joint light curve analysis (JLA). Most recently, Shariff et al. [2016] introduced BAHAMAS (BAyesian Hier-Archical Modeling for the Analysis of Supernova cosmology), an extension of the bayesian method first introduced by March

et al. [2011], and applied it to the SNe Ia sample from the JLA. Shariff et al. [2016] tested for evolution with redshift in the properties of SNe Ia, and investigated whether the posterior variance of the cosmological parameters can be reduced by exploiting correlations between the intrinsic magnitudes of SNe Ia and their host galaxy mass.

While the Shariff et al. [2016] analysis is exhaustive in its investigation, there is room for statistical improvements. In this paper, we address some of the drawbacks of the Shariff et al. [2016] implementation of BAHAMAS. In particular, we make adjustments to the priors, replacing rigid uniform and inverse gamma distributions and with normal distributions. We also demonstrate several computational improvements to the MCMC sampling by using the Stan probabilistic programming language [Stan Development Team, 2016]. Lastly, we investigate whether the residual scatter around the Hubble law can be further reduced by exploiting correlations between the intrinsic magnitudes of SNe Ia and their metallicity, formation rate, and host galaxy age.

This paper is organized as follows. In Section 3.2 we introduce the SALT2 method as well as the original BAHAMAS methdology. In Section 3.3 we describe some issues with the BAHAMAS framework, as well as our extensions and improvements. In Section 3.4 we present results obtained when fitting our improved model in Stan, while conclusions appear in Section 3.5.

3.2 Previous research

We first outline the standard supernova modeling framework. Let $\mathscr{C}$ denote the cosmological parameters of interest,

$$\mathscr{C} = \{H_0, \Omega_m, \Omega_\Lambda, \Omega_\kappa, w\}. \tag{3.1}$$

Here H_0 is a constant known as the value of the Hubble parameter today, equal to 67.8 as of May 2017. The remaining four parameters may or may not be constants as well, depending on whether the universe is assumed to be flat or curved. Under the

flat universe assumption we have that

$$\begin{aligned} \Omega_\Lambda &= 1 - \Omega_m \\ \Omega_\kappa &= 0 \\ w &\neq -1, \end{aligned} \tag{3.2}$$

so that Ω_m and w are the only unknown cosmological parameters. Alternatively, under a curved universe assumption we have that

$$\begin{aligned} \Omega_\kappa &= 1 - \Omega_m - \Omega_\Lambda \\ w &= -1, \end{aligned} \tag{3.3}$$

so that Ω_m and Ω_Λ are the only unknown cosmological parameters.

Then, if we let $z = \{z_{\text{hel}}, z_{\text{cmb}}\}$ denote the heliocentric and cosmic-microwave background adjusted redshifts, respectively, the distance modulus, $\mu(z, \mathscr{C})$, is

$$\mu(z, \mathscr{C}) = 25 + 5 \log_{10} d_L(z, \mathscr{C}), \tag{3.4}$$

where the luminosity distance to redshift $d_L(z, \mathscr{C})$ is

$$d_L(z, \mathscr{C}) = \frac{c(1 + z_{\text{hel}})}{H_0} \text{sinn}_{\Omega_\kappa}(l(z, \mathscr{C})), \tag{3.5}$$

and the integrated luminosity is $l(z, \mathscr{C})$ is

$$l(z, \mathscr{C}) = \int_0^{z_{\text{cmb}}} \left(\Omega_m (1+t)^3 + \Omega_\Lambda (1+t)^{3+3w} + \Omega_\kappa (1+t)^2 \right)^{-\frac{1}{2}} dt. \tag{3.6}$$

Here c is the speed of light and $\text{sinn}_{\Omega_\kappa}(x)$ is defined as

$$\text{sinn}_{\Omega_\kappa}(x) = \begin{cases} x & \text{if } \Omega_\kappa = 0 \\ \frac{\sinh(\sqrt{\Omega_\kappa} x)}{\sqrt{\Omega_\kappa}} & \text{if } \Omega_\kappa > 0 \\ \frac{\sin(\sqrt{\Omega_\kappa} x)}{\sqrt{\Omega_\kappa}} & \text{if } \Omega_\kappa < 0 \end{cases} \tag{3.7}$$

The integrated luminosity expression in (3.6) can be further simplified for each of the two universe assumptions. In the flat universe, the luminosity becomes

$$l(z, \mathscr{C}) = \int_0^{z_{\text{cmb}}} \left(\Omega_m (1+t)^3 + (1 - \Omega_m)(1+t)^{3+3w} \right)^{-\frac{1}{2}} dt. \tag{3.8}$$

whereas in the curved universe, the luminosity becomes

$$l(z, \mathscr{C}) = \int_0^{z_{\text{cmb}}} \left(\Omega_m (1+t)^3 + \Omega_\Lambda + (1 - \Omega_m - \Omega_\Lambda)(1+t)^2 \right)^{-\frac{1}{2}} dt. \tag{3.9}$$

3.2.1 The Shariff et al. [2016] BAHAMAS model

With the distance modulus defined, we now introduce the explicit hierarchical model. Firstly, for each supernova i let M_i denote its intrinsic magnitude and let $\mathscr{D}_i = \{m_{Bi}, x_{1i}, c_{li}\}$ denote the supernova's true peak B-band magnitude, stretch correction, and color correction, respectively. Then, $\hat{\mathscr{D}}_i = \{\hat{m}_{Bi}, \hat{x}_{1i}, \hat{c}_{li}\}$ is the supernova's observed peak magnitude, stretch correction, and color correction, respectively. Lastly, let C_i denote the covariance matrix of $\hat{\mathscr{D}}_i$. Then, the BAHAMAS "baseline" likelihood is

$$\begin{aligned} m_{Bi} &= \mu(z_i, \mathscr{C}) + M_i - \alpha x_{1i} + \beta c_{li} \\ \hat{\mathscr{D}}_i &\sim \text{Normal}(\mathscr{D}_i, C_i), \end{aligned} \tag{3.10}$$

with the hierarchical priors

$$\begin{aligned} M_i &\sim \text{Normal}(M_0, \sigma_{res}) \\ x_{1i} &\sim \text{Normal}(x_{10}, R_{x1}) \\ c_{li} &\sim \text{Normal}(c_{l0}, R_{cl}). \end{aligned} \tag{3.11}$$

While a generative model could be fit to estimate z_{hel} and z_{cmb}, Shariff et al. [2016] found that the posterior distributions of the cosmological parameters and regression coefficients are unaffected if one just treats the observed redshift as the true values

$$\begin{aligned} z_{\text{hel}} &= \hat{z_{\text{hel}}} \\ z_{\text{cmb}} &= \hat{z_{\text{cmb}}}. \end{aligned} \tag{3.12}$$

Additionally, Shariff et al. [2016] treat the observed covariance matrix of $\mathscr{D}_i$ as the true value (i.e. $C_i = \hat{C}_i$), rather than modeling a latent covariance matrix. Finally, Shariff

et al. [2016] use the following priors for the cosmological parameters and regression coefficients

$$\begin{aligned} \Omega_m &\sim \text{Uniform}(0, 2) \\ \Omega_\Lambda &\sim \text{Uniform}(0, 2) \\ H_0 &\sim \text{Normal}(67.8, 0) \\ w &\sim \text{Uniform}(-2, 0) \\ \alpha &\sim \text{Uniform}(0, 1) \\ \beta &\sim \text{Uniform}(0, 4), \end{aligned} \tag{3.13}$$

and use the following priors for the hyperparameters

$$\begin{aligned} M_0 &\sim \text{Normal}(-19.3, 2) \\ \sigma_{res}^2 &\sim \text{InvGamma}(0.003, 0.003) \\ x_{10} &\sim \text{Normal}(0, 10) \\ \log_{10} R_{x1} &\sim \text{Uniform}(-5, 2) \\ c_{l0} &\sim \text{Normal}(0, 1) \\ \log_{10} R_{cl} &\sim \text{Uniform}(-5, 2). \end{aligned} \tag{3.14}$$

In addition to the above "baseline" model, Shariff et al. [2016] also try adding additional covariates to (3.10) such as interaction terms between x_1 and c_l, interactions between z_{cmb} and c_l, and supernova host galaxy mass M_g; but for this paper we restrict our attention to the baseline model. In particular we address some issues that this model presents and also expand it with our own covariates.

3.3 An improved statistical method for expansion rate estimation

The Shariff et al. [2016] BAHAMAS model, while exhaustive and guided closely by astronomy domain knowledge, presents several statistical and computational issues.

Firstly, improvements can be made to the priors to make them more statistically robust and intuitive. Next, additional covariates beyond those proposed in the original paper can help reduce variability in intrinsic magnitude estimates. Lastly, numerous computational improvements can be made by to the MCMC implementation as well as the numerical integration of the luminosity. We now discuss these issues and our proposed improvements.

3.3.1 Improved priors

The first issue we address in the BAHAMAS model is the priors. Inverse gamma priors are used for the scale parameters in (3.14), which have been shown to pull parameters much closer to zero than desired, particularly for hierarchical variances [Gelman, 2006]. Additionally, the uniform priors prevalent in (3.13) too tightly restrict the ranges of the cosmological parameters and regression coefficients, as the uniform priors' limits aren't guided by physical constraints. Lastly, natural logarithms are preferred to the base ten logarithms in (3.14) when transforming the scale parameters.

To improve these model choices, we replace all uniform priors by normal priors (or half normal priors in the cases where the range of a cosmological parameter is physically restricted to be positive or negative), which results in the following changes:

$$\begin{aligned}
\Omega_m &\sim \text{Uniform}(0,2) &\Rightarrow \quad \Omega_m &\sim \text{Normal}^{+}(0,1) \\
\Omega_\Lambda &\sim \text{Uniform}(0,2) &\Rightarrow \quad \Omega_\Lambda &\sim \text{Normal}^{+}(1,1) \\
w &\sim \text{Uniform}(-2,0) &\Rightarrow \quad w &\sim \text{Normal}^{-}(-1,1) \\
\alpha &\sim \text{Uniform}(0,1) &\Rightarrow \quad \alpha &\sim \text{Normal}(0.5,0.5) \\
\beta &\sim \text{Uniform}(0,4) &\Rightarrow \quad \beta &\sim \text{Normal}(2,2)
\end{aligned} \tag{3.15}$$

Additionally, inverse gamma priors for the scale parameters are replaced by half normal priors, while logarithms with base ten are replaced with natural logarithms, resulting

in the following changes:

$$\begin{aligned} \sigma_{res}^2 \sim \text{InvGamma}(0.003, 0.003) \quad &\Rightarrow \quad \sigma_{res} \sim \text{Normal}^+(0, 1) \\ x_{10} \sim \text{Normal}(0, 10) \quad &\Rightarrow \quad x_{10} \sim \text{Normal}(0, 1) \\ \log_{10} R_{x1} \sim \text{Uniform}(-5, 2) \quad &\Rightarrow \quad \log R_{x1} \sim \text{Normal}(-1.5, 2.5) \\ \log_{10} R_{cl} \sim \text{Uniform}(-5, 2) \quad &\Rightarrow \quad \log R_{cl} \sim \text{Normal}(-1.5, 2.5) \end{aligned} \tag{3.16}$$

Despite changing these priors, we set the centers and scales of the normal distributions to represent the information that is incorporated in the original Shariff et al. [2016] priors. Thus, while our new priors have less rigid supports, they still place mass over roughly the same regions that the BAHAMAS priors do.

3.3.2 Additional covariates

We also explore some additional covariates that could reduce the variability in estimating the intrinsic magnitude. Namely, the Campbell Institute observed additional covariates on a subset of 113 supernovae from the joint light curve analysis. Of these newer variables, we believe that star metallicity ($\hat{m}_t$), star formation rate ($\hat{f}_r$), and galaxy age ($\hat{a}_g$) are the most correlated with intrinsic magnitude.

As such we try the following extensions of the baseline model in (3.10). Firstly, we try including metallicity with

$$\begin{aligned} m_{Bi} &= \mu(z_i, \mathscr{C}) + M_i - \alpha x_{1i} + \beta c_{li} + \gamma_m m_{ti} \\ \gamma_m &\sim \text{Normal}(0, 1). \end{aligned} \tag{3.17}$$

Then we try including star formation rate with

$$\begin{aligned} m_{Bi} &= \mu(z_i, \mathscr{C}) + M_i - \alpha x_{1i} + \beta c_{li} + \gamma_f f_{ri} \\ \gamma_f &\sim \text{Normal}(0, 1), \end{aligned} \tag{3.18}$$

and age with

$$m_{Bi} = \mu(z_i, \mathscr{C}) + M_i - \alpha x_{1i} + \beta c_{li} + \gamma_a a_{gi} \tag{3.19}$$
$$\gamma_a \sim \text{Normal}(0, 1).$$

Unlike our data recordings for stretch (x_1) and color (c_l) corrections, we do not have noise estimates recorded for metallicity, formation rate, and age. As such, we treat the observed values of these covariates as the true values

$$\begin{aligned} m_t &= \hat{m}_t \\ f_r &= \hat{f}_r \\ a_g &= \hat{a}_g \end{aligned} \tag{3.20}$$

Additionally, if we were to fit the models in (3.17), (3.18), and (3.19) using just the subset of 113 supernovae for which these covariates have been recorded, the cosmological parameter estimates would be very noisy. As such, we fit these 113 supernovae together with the 627 supernovae that do not have the new covariates observed. For each of the metallicity, formation rate, and age model extensions we assign one of the above three likelihoods for the 113 supernovae; while we assign the baseline likelihood in (3.10) to the remaining 627 supernovae. In this implementation, we allow both sets of data to have different values of the regression coefficients α and β. However, we ensure that the cosmological parameters $\mathscr{C}$ are the same for both sets of data.

3.3.3 Computational improvements

The last improvement we make is in the computation time of the fitting the model itself. While BAHAMAS takes a few days to run, our implementation takes only hours. This is achieved by using the Stan programming language [Stan Development Team, 2016], which provides us with three key advantages.

Firstly, Stan uses the Hoffman and Gelman [2014] No-U-Turn sampler (NUTS), an extension of Hamiltonian Monte Carlo (HMC), that is much faster than the Gibbs-type sampler used by Shariff et al. [2016]. Gibbs and Metropolis sampling both require a long time to converge to the target distribution for complicated models with many parameters, in large part due to the tendency of these methods to explore parameter space via inefficient random walks [Neal, 1993]. HMC, however, is able to suppress such random walk behavior by transforming the problem of sampling from a target distribution into the problem of simulating Hamiltonian dynamics [Neal et al., 2011]. HMC typically requires practitioners to specify the step size ϵ and the number of steps L. NUTS, however, takes an exponentially increasing number of steps forward and backward in time until the direction of the simulation turns around, then uses slice sampling [Neal, 2003] to select a point on the simulated trajectory. By vectorizing the parameters and running NUTS in C++, Stan is able to provide huge computational advantages over the Gibbs sampler implemented by Shariff et al. [2016] in Python.

Secondly, Stan's differential equation integrator can be easily included into the executable that is compiled prior to running HMC. By using a fourth and fifth order Runge-Kutta method [Dormand and Prince, 1980] for integration at compile time, Stan is able to integrate directly in compiled C++ [Ahnert and Mulansky, 2011], which is much faster than the higher-level Python implementation of BAHAMAS. At the same time, however, the code for writing the integration solver in the Stan language is simpler than doing so in Python. The only adjustment required for a more efficient implementation is to simplify the limits of the integral with a change of variables. This is achieved by replacing t in (3.8) and (3.9) with $u \cdot z_{\text{cmb}}$. In the flat universe, the integral then becomes

$$l(z, \mathscr{C}) = \int_0^1 z_{\text{cmb}} \left(\Omega_m (1 + z_{\text{cmb}} u)^3 + (1 - \Omega_m)(1 + z_{\text{cmb}} u)^{3+3w} \right)^{-\frac{1}{2}} du. \tag{3.21}$$

whereas in the curved universe, the integral becomes

$$l(z, \mathscr{C}) = \int_0^1 z_{\text{cmb}} \left(\Omega_m (1 + z_{\text{cmb}} u)^3 + \Omega_\Lambda + (1 - \Omega_m - \Omega_\Lambda)(1 + z_{\text{cmb}} u)^2 \right)^{-\frac{1}{2}} du. \tag{3.22}$$

Lastly, the probabilisitic programming language of Stan makes the model easily readable, shareable, and mutable. Due to the algorithmic differentiation included in Stan, one simply has to code the prior and likelihood in order to run the MCMC algorithm. This means that the model can easily be modified by collaborators, which ultimately results in more efficient research progress. While this isn't a computational gain in the literal sense of the algorithm, it is a valuable contribution that should not go unnoticed. The power of this approach is most clearly seen by comparing the appendix of BAHAMAS, which includes over a dozen pages of MCMC implementation details behind each of the different models that were tried, and the much shorter Stan model included in Appendix B. Indeed, the Stan model contains all of the various model cases in one compact form, in addition to the curved and flat universe integrals, while hiding all of the algorithmic implementation details. This allows the researcher to focus strictly on the mathematical model without worrying about the computational implementation itself.

3.4 Results

We present the posterior results of fitting the original BAHAMAS baseline, our baseline with improved priors, and our extensions with metallicity, formation rate, and age; all using Stan. All models are run using 4 chains with 1500 iterations each, of which half are discarded for warmup.

3.4.1 Baseline

We first fit the original baseline model from Shariff et al. [2016] in Stan.

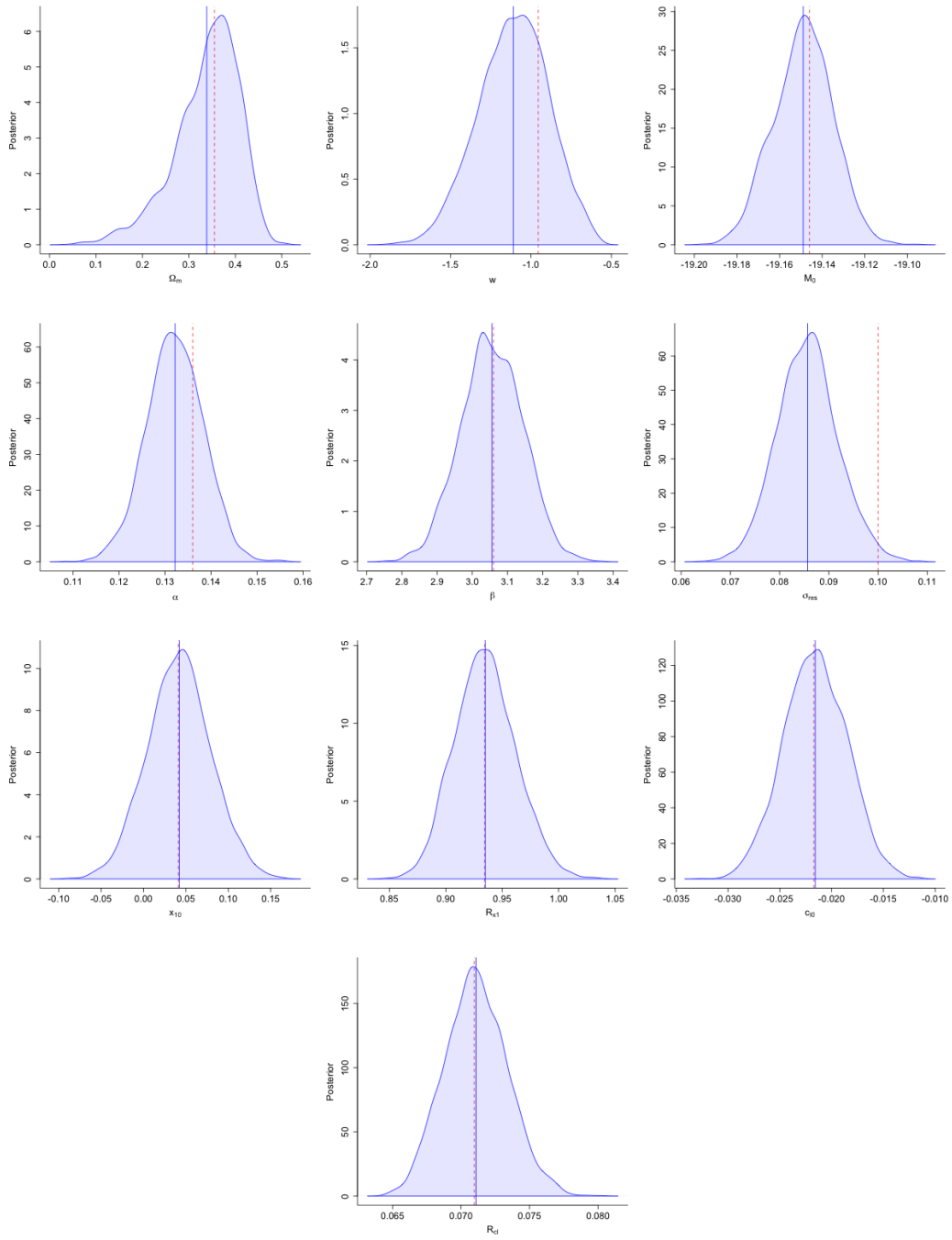

Figure 3.1: Posterior distributions of the global parameters for the baseline (flat universe) model. Vertical blue lines correspond to posterior means while red lines correspond to BAHAMAS posterior means.

Figure 3.1 displays the global parameter posteriors of the baseline model assuming a flat universe. Posterior means from the Shariff et al. [2016] implementation are shown in red. While most of the parameters have similar posterior means, the σ_{res} is quite off. This may be due to the original implementation using a more outdated set of values for the covariance matrix $\hat{C}$. Our results shown here use the latest values from the JLA study and are easily reproducible using our code. We believe the inconsistency in the $\hat{C}$, particularly with respect to the variance terms of $\hat{m}_B$, is also what prevents our cosmological parameters from lining up exactly with the BAHAMAS study. This is evidenced by the posterior means of x_{10}, R_{x1}, c_{l0}, and R_{cl} matching perfectly with the Shariff et al. [2016] posterior means.

Figure 3.2 displays the global parameter posteriors of the baseline model assuming a curved universe. The difference in posterior means between our implementation and the Shariff et al. [2016] implementation are similar to those for the flat universe. Namely, the posterior distributions of $\beta, x_{10}, R_{x1}, c_{l0}$, and R_{cl} are nearly identical to those arrived at by Shariff et al. [2016]. However the posteriors of $\sigma_{res}, \alpha, \Omega_m$, and Ω_Λ are quite different. Again, the difference in these latter four parameters is due the difference in the values of $\hat{C}$.

Aside from the technical issues behind the choice of the covariance matrix $\hat{C}$, there is no other reason to believe a Metropolis or Gibbs sampler would arrive to such starkly different posterior distributions as a NUTS or HMC sampler would. With that in mind, for the remaining model extensions we compare the newer models' posteriors to our own baseline posteriors for the curved and flat universe assumptions. This allows us to focus on the incremental impact of model choice rather than issues arising from data consistency.

3.4.2 Baseline with new priors

We next present the results of adjusting the priors of the baseline model, as specified in Section 3.3.1.

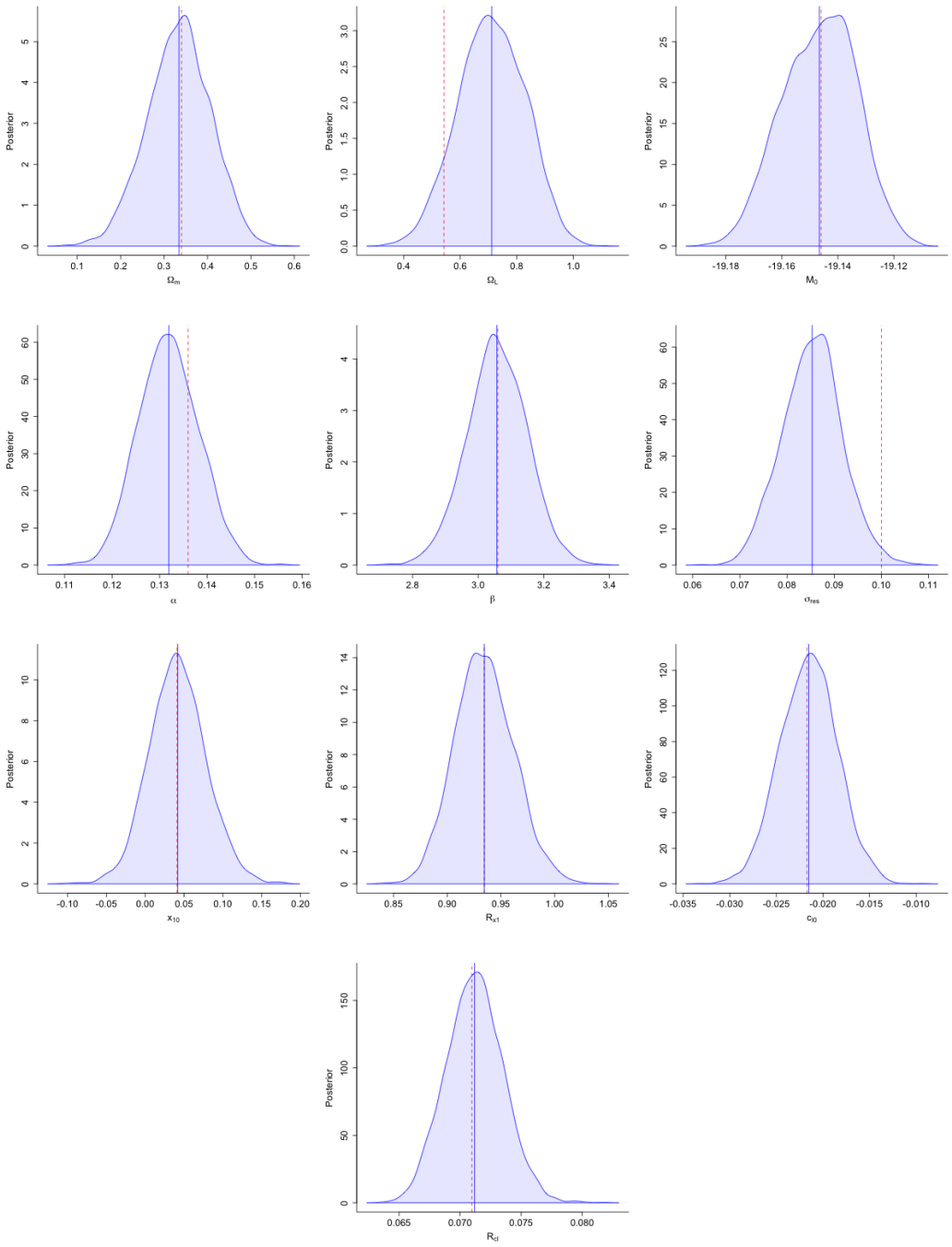

Figure 3.2: Posterior distributions of the global parameters for the baseline (curved universe) model. Vertical blue lines correspond to posterior means while red lines correspond to BAHAMAS posterior means.

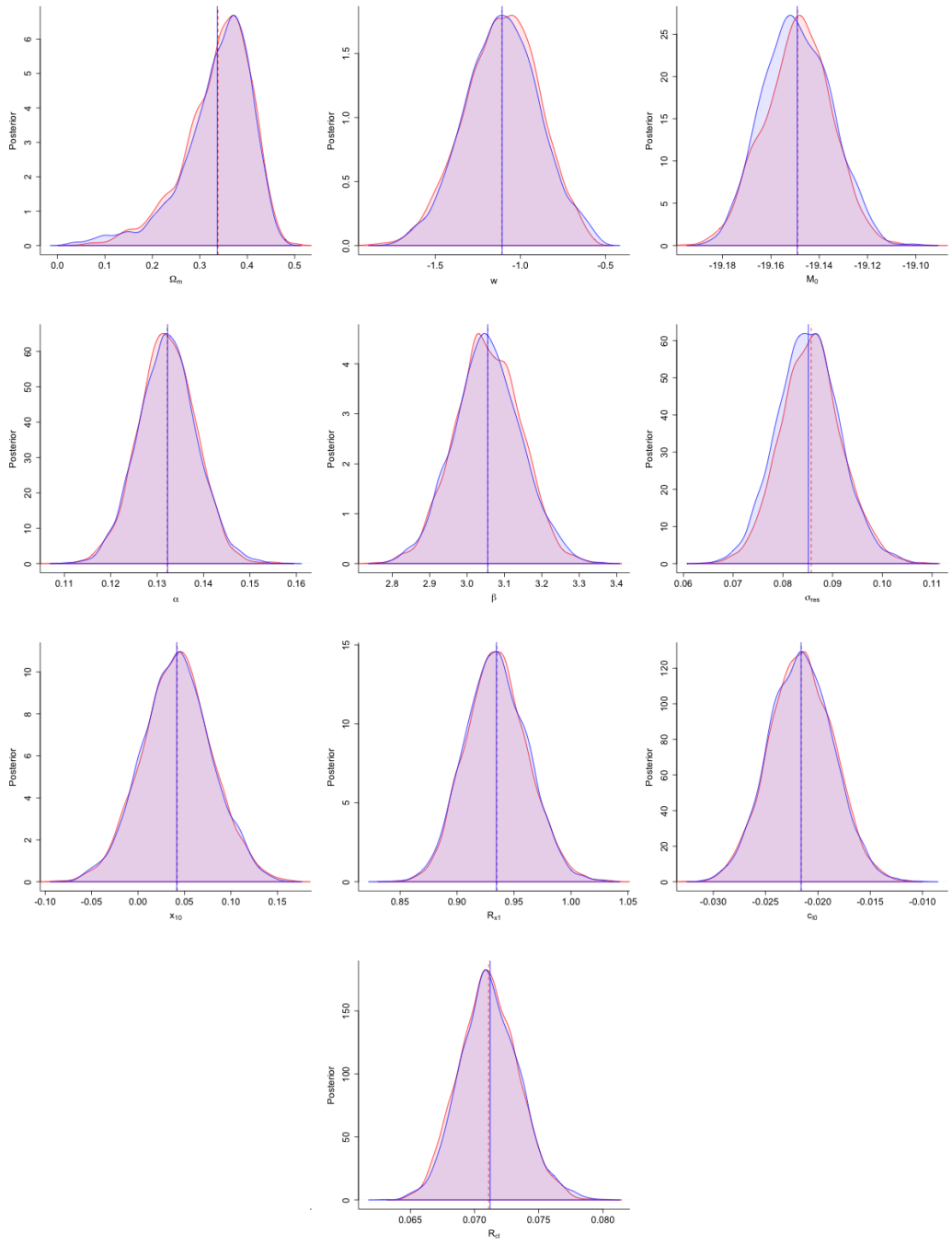

Figure 3.3: Posterior distributions of the global parameters for the baseline (flat universe) model with new priors. Results from the older priors are shown in red. Vertical lines correspond to posterior means.

Figure 3.3 displays the global parameter posteriors when using new priors for the baseline model assuming a flat universe. The posteriors resulting from the Shariff et al. [2016] priors are shown in red. The posteriors are nearly identical, showing that the results aren't sensitive to the prior choice.

Figure 3.4 displays the global parameter posteriors when using new priors for the baseline model assuming a curved universe. The posteriors resulting from the Shariff et al. [2016] priors are shown in red. The posteriors are nearly identical for the curved universe as well, showing that the results aren't sensitive to the prior choice.

For the remaining three model extensions, we omit the posterior results for x_{10}, R_{x1}, c_{l0}, and R_{cl} since these global parameters are unaffected by inclusion of a third covariate.

3.4.3 Star metallicity

We next present the results of adding star stametallicity to the baseline model with new priors. Comparisons are made to the posteriors resulting from fitting the baseline model with new priors, as in Section 3.4.2. Here α_1 and β_1 correspond to the stretch and color correction regression coefficients learned from the subset of 113 supernovae from the Campbell study, which are entered into the likelihood via (3.17). Additionally, γ_m is the star metallicity regression coefficient learned from the Campbell supernovae subset. Meanwhile, α and β correspond to regression coefficients learned from the remaining 627 supernovae, which are entered into the likelihood via (3.10). The remaining parameters are shared in the likelihood by both subsets of supernovae.

Figure 3.5 displays the global parameter posteriors when using new priors for the baseline model assuming a flat universe, and adding star metallicity as a covariate. The posteriors resulting from the new priors without metallicity are shown in red. The global shared parameters $\Omega_m, \Omega_\Lambda, M_0$, and σ_{res} are relatively unchanged by the addition of metallicity as a covariate, while the regression parameters α and β are changed primarily because they are learned from a subset of 627 supernovae, as

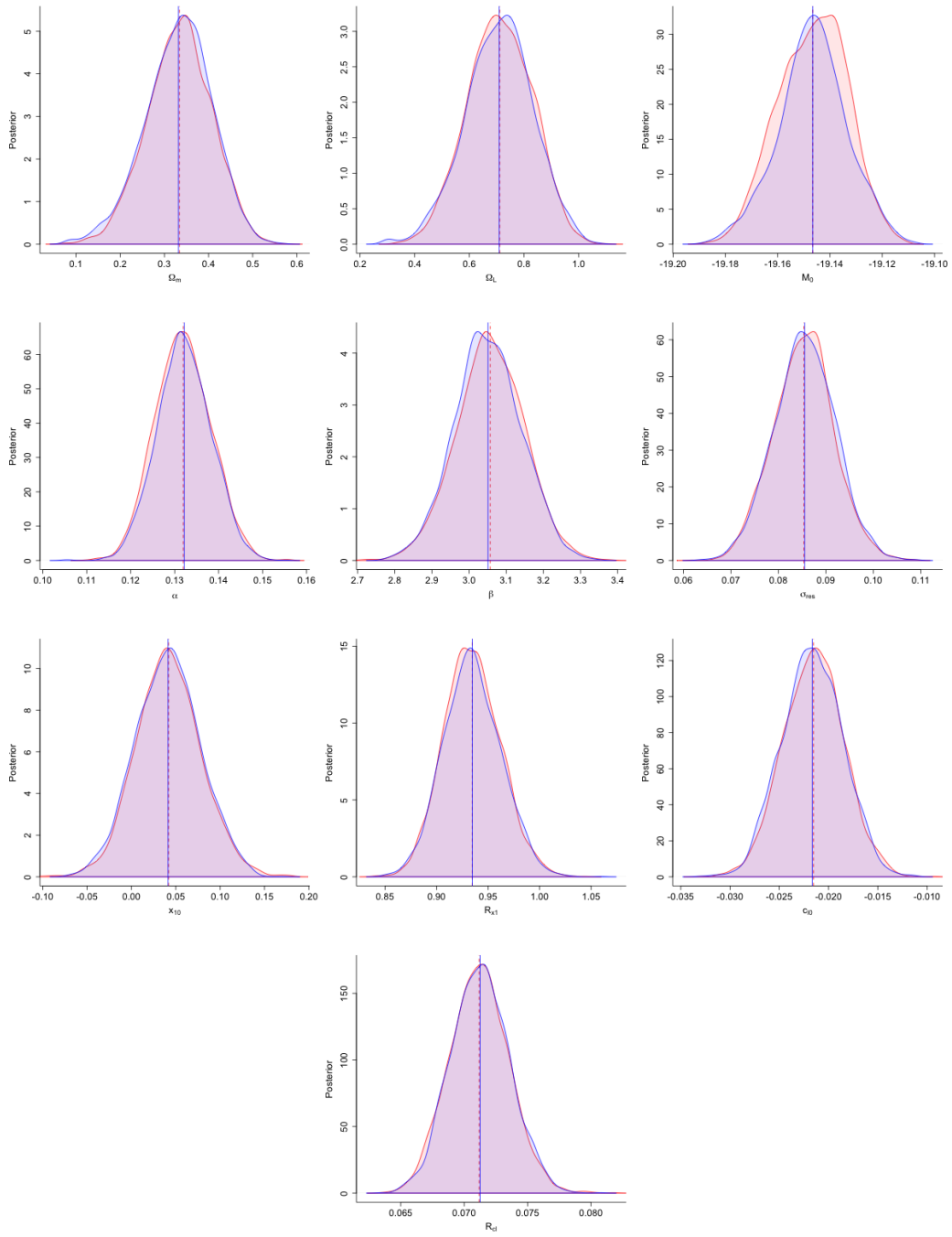

Figure 3.4: Posterior distributions of the global parameters for the baseline (curved universe) model with new priors. Results from the older priors are shown in red. Vertical lines correspond to posterior means.

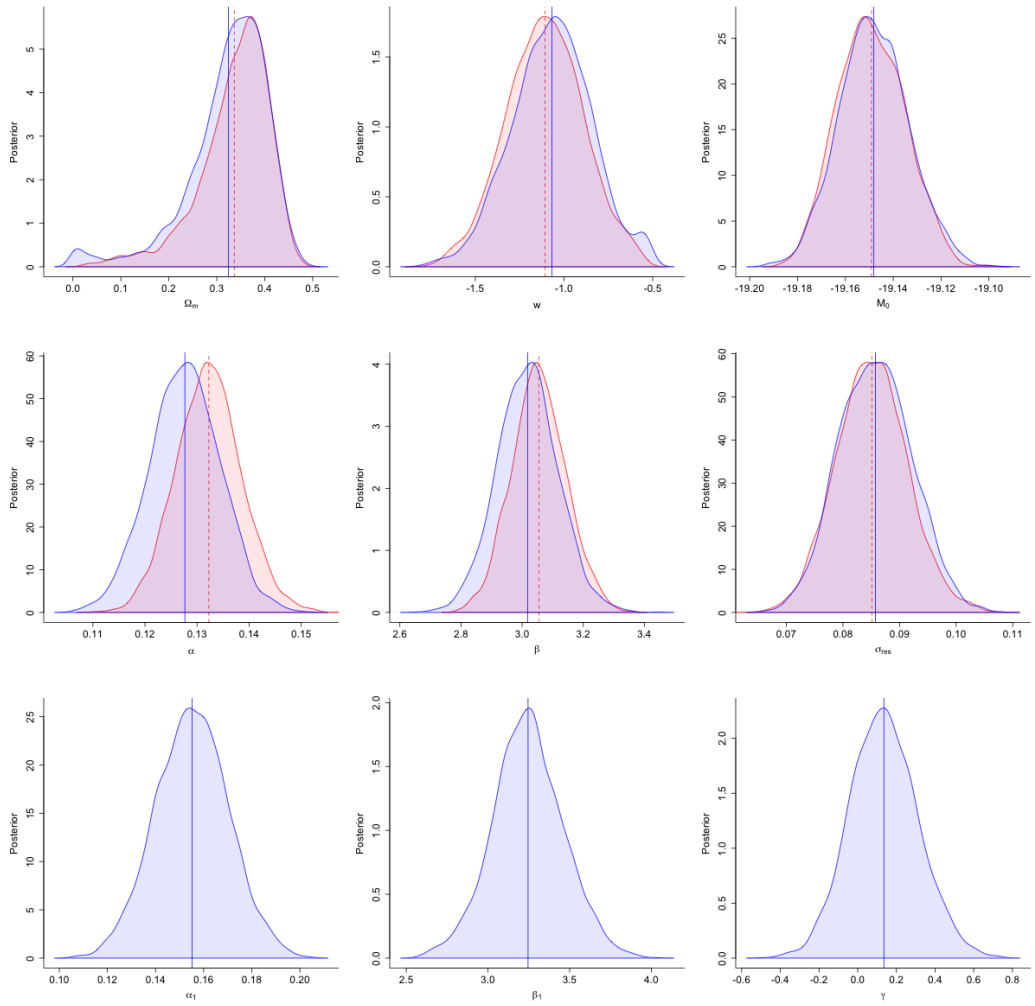

Figure 3.5: Posterior distributions of the global parameters for the baseline (flat universe) model with new priors and metallicity. Results without metallicity are shown in red. Vertical lines correspond to posterior means.

opposed to all 740. Most crucially, however, the posterior of γ_m contains 0 even in its central 50% interval, suggesting that star metallicity does not help reduce the variation with which we can estimate the cosmological parameters.

Figure 3.6 displays the global parameter posteriors when using new priors for the baseline model assuming a curved universe, and adding star metallicity as a covariate. The posteriors resulting from the new priors without metallicity are shown in red. As under the flat universe assumption, the posterior of γ_m contains 0 even in its central 50% interval, suggesting that star metallicity does not help reduce the variation with which we can estimate the cosmological parameters.

3.4.4 Star formation rate

We next present the results of adding star formation rate to the baseline model with new priors. Comparisons are made to the posteriors resulting from fitting the baseline model with new priors, as in Section 3.4.2. Here α_1 and β_1 correspond to the stretch and color correction regression coefficients learned from the subset of 113 supernovae from the Campbell study, which are entered into the likelihood via (3.18). Additionally, γ_f is the star formation rate regression coefficient learned from the Campbell supernovae subset. Meanwhile, α and β correspond to regression coefficients learned from the remaining 627 supernovae, which are entered into the likelihood via (3.10). The remaining parameters are shared in the likelihood by both subsets of supernovae.

Figure 3.7 displays the global parameter posteriors when using new priors for the baseline model assuming a flat universe, and adding star formation rate as a covariate. The posteriors resulting from the new priors without metallicity are shown in red. The posterior of γ_f contains 0 even in its central 50% interval, suggesting that star formation rate does not help reduce the variation with which we can estimate the cosmological parameters.

Figure 3.8 displays the global parameter posteriors when using new priors for

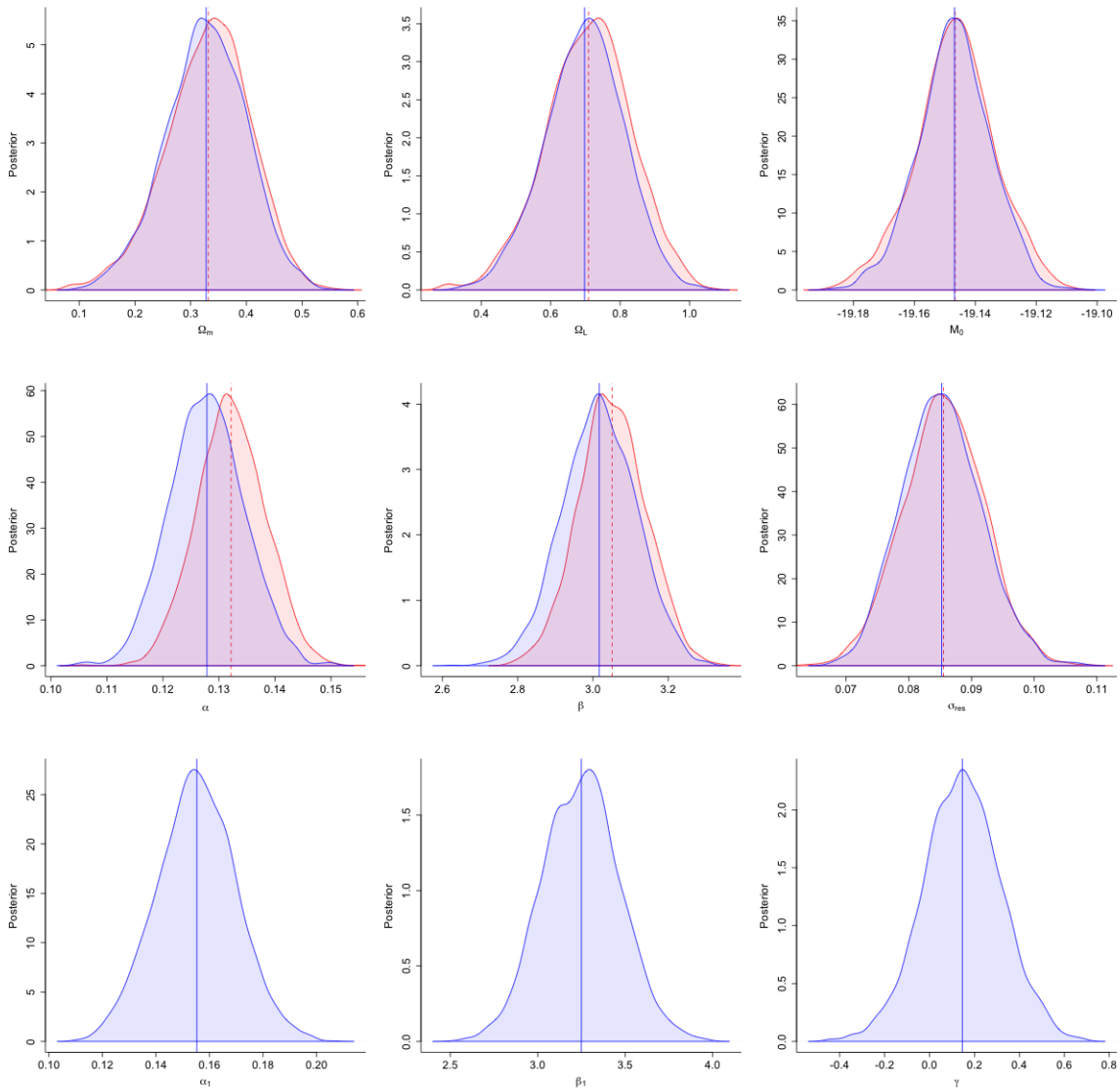

Figure 3.6: Posterior distributions of the global parameters for the baseline (curved universe) model with new priors and metallicity. Results without metallicity are shown in red. Vertical lines correspond to posterior means.

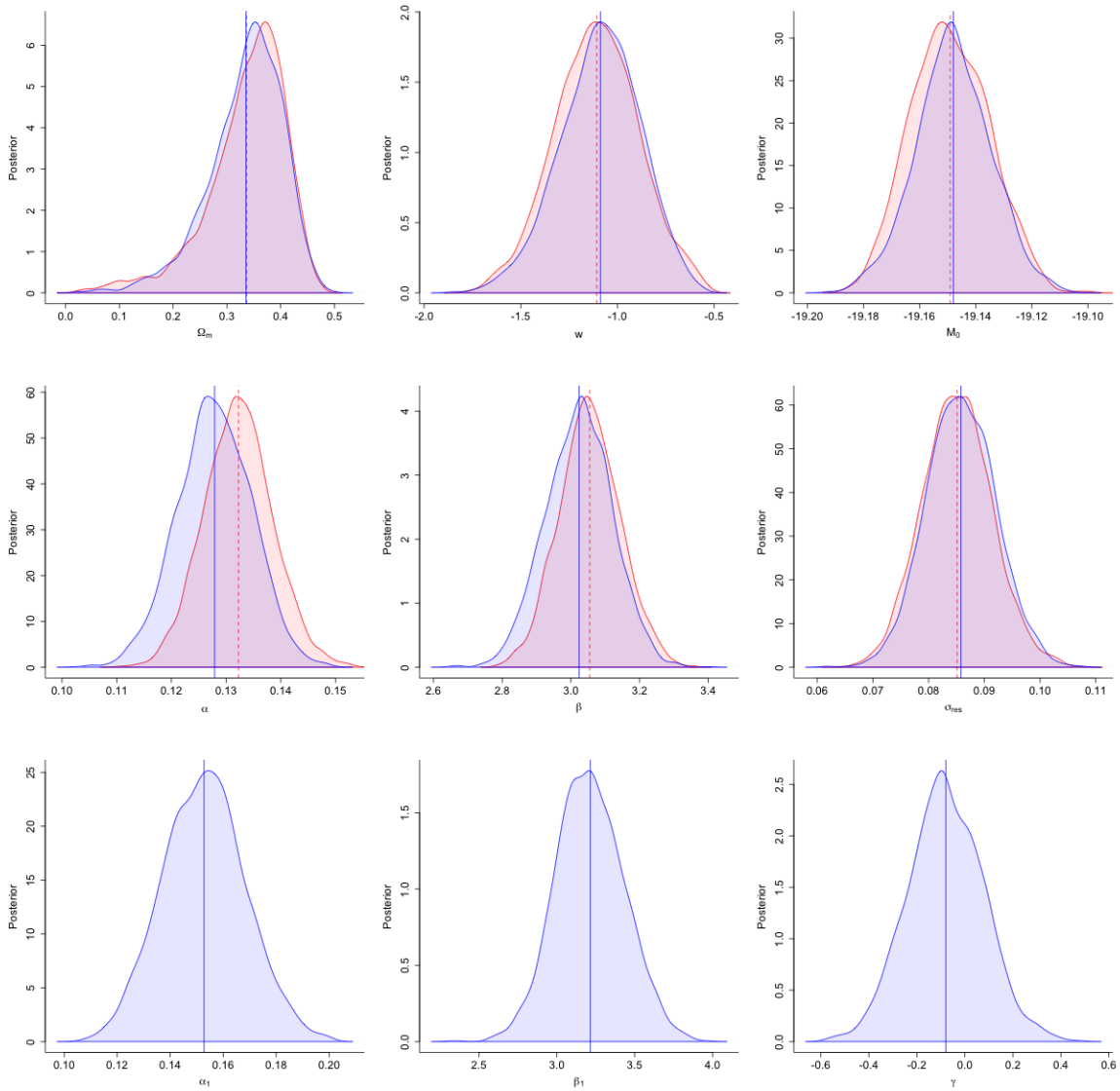

Figure 3.7: Posterior distributions of the global parameters for the baseline (flat universe) model with new priors and formation rate. Results without formation rate are shown in red. Vertical lines correspond to posterior means.

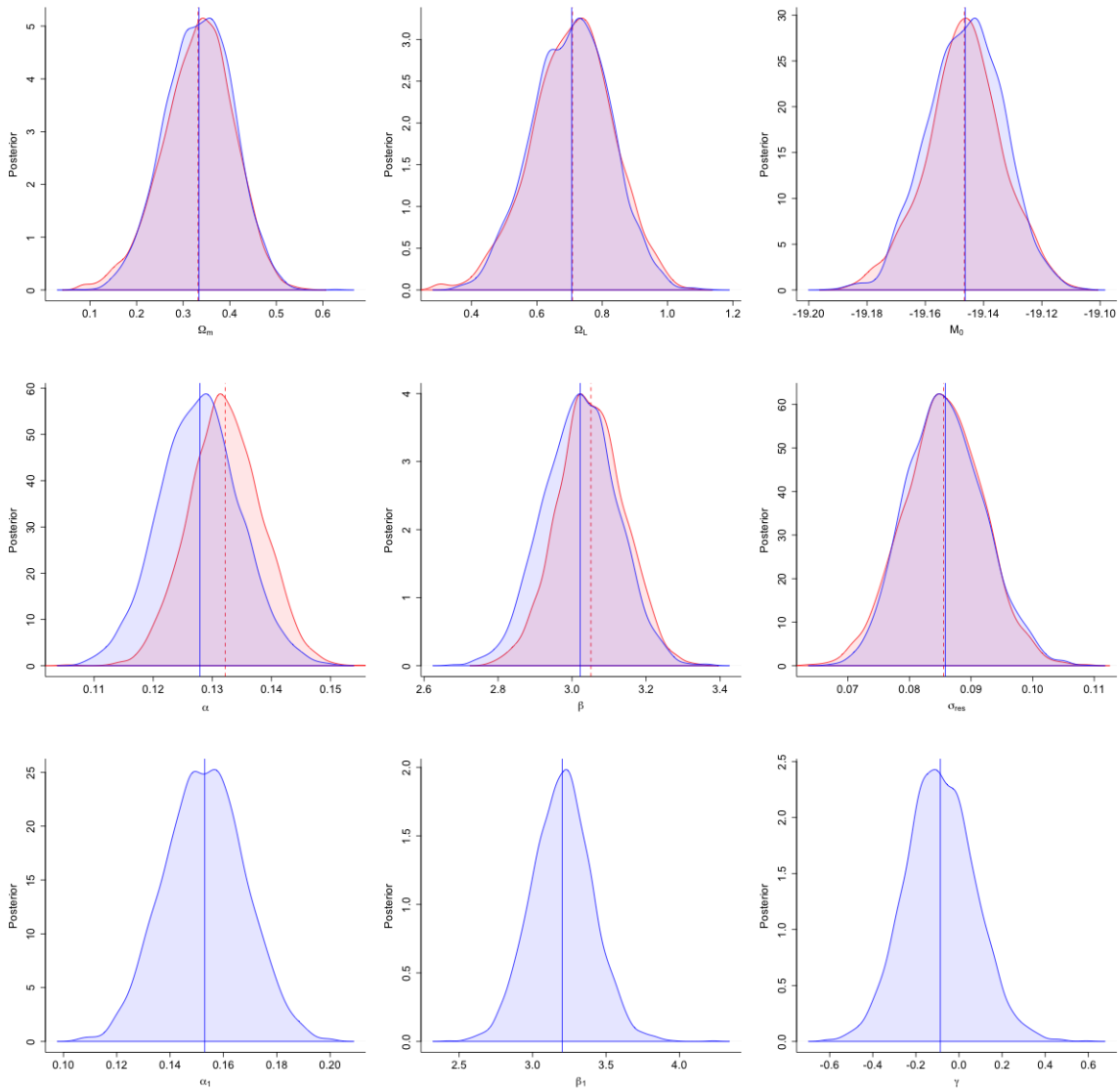

Figure 3.8: Posterior distributions of the global parameters for the baseline (curved universe) model with new priors and formation rate. Results without formation rate are shown in red. Vertical lines correspond to posterior means.

the baseline model assuming a curved universe, and adding star formation rate as a covariate. The posteriors resulting from the new priors without metallicity are shown in red. As under the flat universe assumption, the posterior of γ_f contains 0 even in its central 50% interval, suggesting that star formation rate does not help reduce the variation with which we can estimate the cosmological parameters.

3.4.5 Galaxy age

We next present the results of adding galaxy age to the baseline model with new priors. Comparisons are made to the posteriors resulting from fitting the baseline model with new priors, as in Section 3.4.2. Here α_1 and β_1 correspond to the stretch and color correction regression coefficients learned from the subset of 113 supernovae from the Campbell study, which are entered into the likelihood via (3.19). Additionally, γ_a is the galaxy age coefficient learned from the Campbell supernovae subset. Meanwhile, α and β correspond to regression coefficients learned from the remaining 627 supernovae, which are entered into the likelihood via (3.10). The remaining parameters are shared in the likelihood by both subsets of supernovae.

Figure 3.9 displays the global parameter posteriors when using new priors for the baseline model assuming a flat universe, and adding galaxy age as a covariate. The posteriors resulting from the new priors without metallicity are shown in red. The posterior of γ_a is essentially 0, suggesting that galaxy age does not help reduce the variation with which we can estimate the cosmological parameters.

Figure 3.10 displays the global parameter posteriors when using new priors for the baseline model assuming a curved universe, and adding galaxy age as a covariate. The posteriors resulting from the new priors without metallicity are shown in red. As under the flat universe assumption, the posterior mean of γ_a is essentially 0, suggesting that galaxy age does not help further reduce the variation with which we can estimate the cosmological parameters.

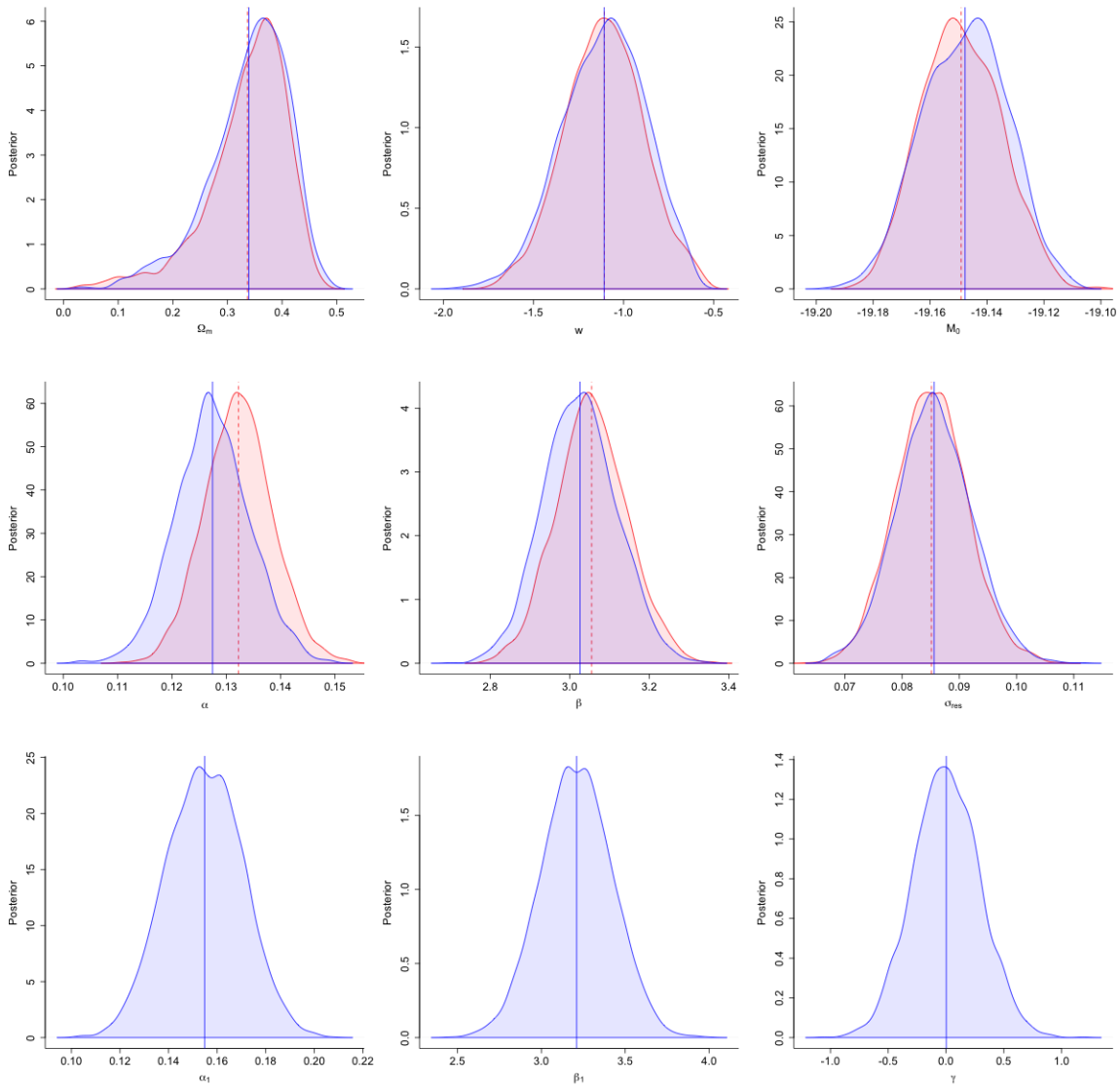

Figure 3.9: Posterior distributions of the global parameters for the baseline (flat universe) model with new priors and age. Results without age are shown in red. Vertical lines correspond to posterior means.

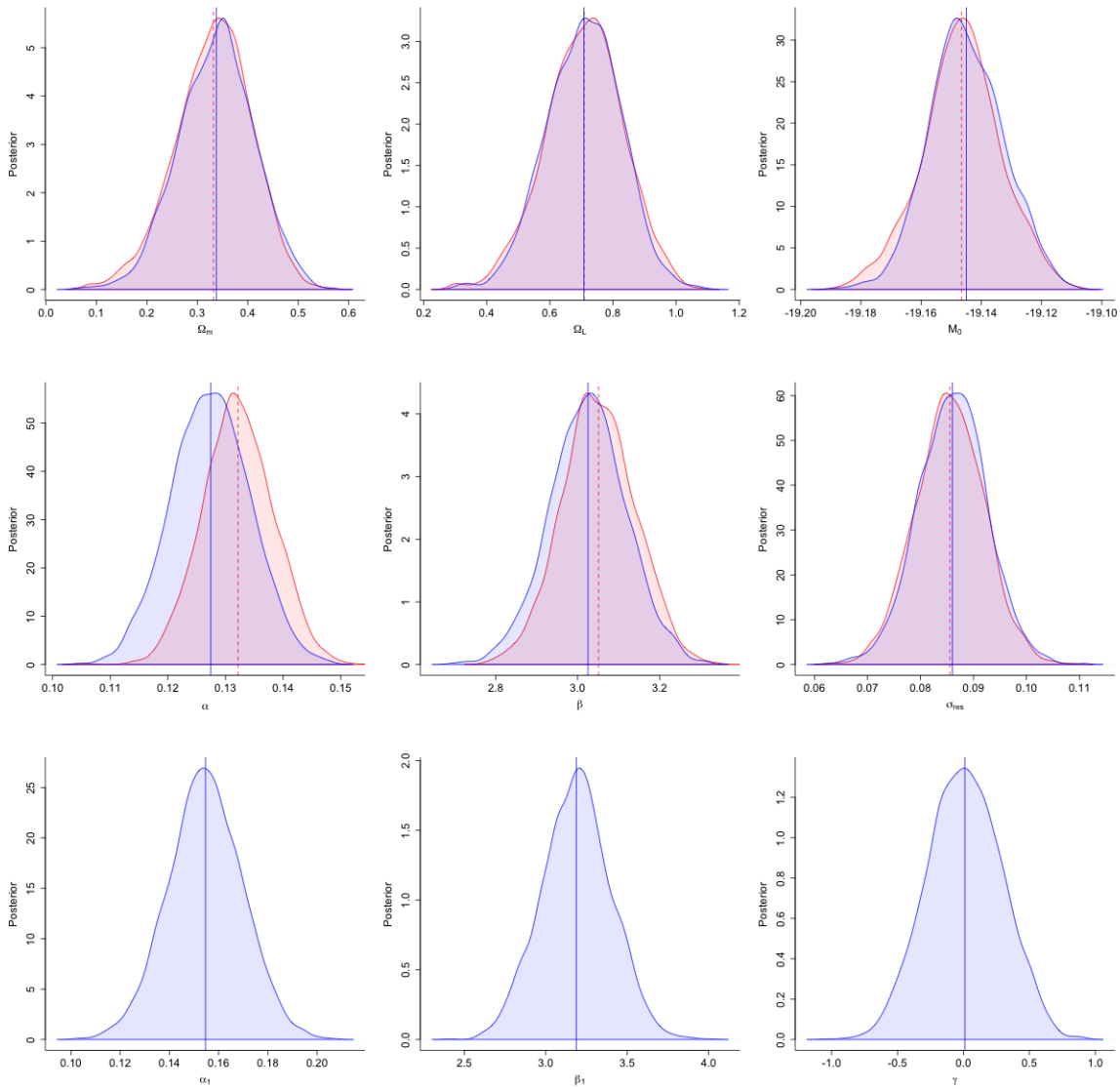

Figure 3.10: Posterior distributions of the global parameters for the baseline (curved universe) model with new priors and age. Results without age are shown in red. Vertical lines correspond to posterior means.

3.5 Discussion

This paper presents an improved statistical method of learning cosmological parameters from type-Ia supernovae. We show that the Stan programming language, combined with an improved set of priors, provides massive computational gains in both speed and stability over current approaches.

Recent work has shown that cosmological parameters can be learned by using a clever hierarchical model that pools various supernovae-related covariates together. In particular, Shariff et al. [2016] show that a Gibbs sampler can be applied to a series of models containing various interesting covariates, resulting in more precise estimates of the cosmological parameters. This implementation, however, uses rigid uniform priors, and also takes multiple days to converge to the posterior. In contrast, the NUTS implementation in Stan uses more flexible normally distributed priors and a fourth and fifth order Runge-Kutta method in C++ to converge to the posterior in just a few hours. Perhaps most crucially, the Stan code for the model is easily interpretable because of its probabilistic nature, and its results can be easily reproduced with the model file spanning less than 250 lines.

Despite the improvements presented here, numerous other changes could be made to the BAHAMAS model. The heliocentric and cosmic microwave background adjusted redshifts, for example, could be incorporated into a measurement error model instead of simply treating the observed values as the true redshifts. While this would normally cause numerical issues when computing the integrated luminosity (because redshift is the upper limit of the integral), our change of variables allows the integral's limits to be constants, making the computation more tractable. Additionally, the empirical covariance matrix of the peak magnitude, stretch correction, and color correction could be incorporated into a measurement error model as well. In particular, methods using Wishart (not inverse-Wishart) priors have shown guarantees of positive definite posterior modes [Chung et al., 2015].

Chapter 4

Latent mixing kernel for aggregate relational data

4.1 Introduction

Currently, the dominant framework for empirical social science is the sample survey. Such surveys, however, have made it traditionally difficult to understand the composition of social networks. Indeed, sample surveys been have described by Barton [1968] as a "meatgrinder" that completely removes people from their social contexts. This is unfortunate because social networks have become an increasingly common framework for understanding and explaining social phenomena. In recent years, however, an abundance of sophisticated models have shown how partially observed data, via sample surveys, can be used to accurately estimate properties about an individual's social network.

McCarty et al. [2001] showed that Aggregated Relational Data (ARD), which ask respondents how many connections they have with members of a certain subpopulation (e.g. How many individuals do you know who are gay?), can be used to estimate personal network size via the scale-up method. Zheng et al. [2006] further expanded the limits of ARD by introducing overdispersion into the scale-up method as a way to

estimate non-random social mixing. Most recently, McCormick et al. [2010] introduced the latent non-random mixing matrix framework as a way to quantify the mixing patterns between people from different age groups and genders using ARD. It is this last model that we use as inspiration to develop a more powerful, yet statistically robust, framework for estimating social mixing patterns from partially observed data.

In particular, this paper extends the discrete McCormick et al. [2010] mixing matrix into a continuous, structured framework by deriving a latent kernel representation of social mixing patterns. Instead of binning ego and alter ages into categories, which is more conducive to a discrete mixing matrix, we treat age as continuous. Furthermore, we replace the discrete rows of the mixing matrix with a continuous mixing kernel whose center is equal to the ego's age. Lastly, we allow the scale (bandwidth) of the mixing kernels to depend not only on ego and alter gender, but also on ego age. The result is a pair of gender-specific kernels, each uniquely defined for an ego's given age and gender, that encapsulate the age distributions of the ego's female and male acquaintanceships.

This paper is organized as follows. We begin in Section 4.2 with a review of previous attempts to measure personal network size and social mixing patterns, focusing on the latent non-random mixing matrix of McCormick et al. [2010] which is promising, but suffers from instability in estimating the rows of the discrete mixing matrix. In Section 4.3 we derive a continuous latent mixing kernel framework which resolves these problems, and as a byproduct enables estimation of the mixing kernel bandwidth as a function of age. In Section 4.4 we outline the priors and likelihood of our model, and describe our model fitting algorithm. In Section 4.5 we then discuss the results of fitting the model to 1,190 survey responses from an online survey we designed, asking respondents how many people they know with certain first names or occupations. In Section 4.6, we draw on insights developed during the statistical modeling to offer guidelines for future improvements.

4.2 Previous research

Before presenting our model for estimating social mixing patterns using ARD, it is important to review previous work in estimating social mixing. These methodologies were created as byproducts of trying to more accurately estimate personal network size via the scale-up method [Killworth et al., 1998].

To gain an intuition for the scale-up method, consider a population of size N. We can store the information about the social network connecting the population in an adjacency matrix $\triangle = [\delta_{ij}]_{NxN}$ such that $\delta_{ij} = 1$ if person i knows person j. Throughout this paper we will assume the McCarty et al. [2001] definition of know: "that you know them and they know you by sight or by name, that you could contact them, that they live within the United States, and that there has been some contact (either in person, by telephone or mail) in the past 2 years." The personal network size or degree of person i is then $d_i = \sum_j \delta_{ij}$.

Since it is unrealistic to ask survey respondents whether they know every single individual in the population, the Killworth et al. [1998] scale-up method uses responses to ARD to estimate personal network size. For example, if you report knowing 3 women who gave birth, this represents about one-millionth of all women who gave birth within the last year. Assuming that your personal network's demographics are similar to that of the whole country, we can then use this information to estimate that you know about one-millionth of all Americans,

$$\frac{3}{3.6 \text{ million}} \cdot (300 \text{ million}) \approx 250 \text{ people}. \tag{4.1}$$

As such, in Section 4.2.1 we present an earlier model by Zheng et al. [2006] that formalizes an overdispersion framework of the scale-up method, accounting for the variation in propensity to know individuals from certain subpopulations. This overdispersion framework, while providing a probabilistic interpretation of the scale-up method, suffers from transmission errors and barrier effects. In Section 4.2.2 we present a more recent model by McCormick et al. [2010] that addresses these issues,

and as a byproduct allows estimation of the mixing patterns between age and gender groups.

4.2.1 The Zheng et al. [2006] model with overdispersion

The multilevel overdispersed Poisson model of Zheng et al. [2006] was the first to treat random mixing as something important to estimate for its own sake.

Let y_{ik} denote the number of people that person i knows in subpopulation $\mathscr{G}_k$, N_k denote he size of subpopulation $\mathscr{G}_k$, and N denote the size of the population. Zheng et al. [2006] noted that under simple random mixing the responses to the ARD questions, y_{ik}'s, would follow a Poisson distribution with rate parameter determined by the degree of person i, d_i, and the network prevalence of group $\mathscr{G}_k$, b_k:

$$\begin{aligned} y_{ik} &\sim \text{Poisson}(\lambda_{ik}) \\ \lambda_{ik} &= f(d_i, b_k). \end{aligned} \tag{4.2}$$

Here b_k is the proportion of ties that involve individuals in subpopulation k in the entire social network. If we can assume that individuals in the group being asked about (e.g. people named Michael), on average, as popular as the rest of the population, then $b_k \approx N_k/N$.

Like Killworth et al. [1998] before, Zheng et al. [2006] used the telephone survey conducted by McCarty et al. [2001] to test the above hypothesis. The responses to many of the questions in the survey data did not follow a Poisson distribution, however. In fact, most of the responses show overdispersion (i.e. excess variance given the mean). For example, consider the responses to the question: "How many males do you know incarcerated in state or federal prison?" The mean of the responses to this question was 1.0, but the variance was 8.0, indicating that some people are much more likely to know someone in prison than others. To model this increased variance Zheng et al. [2006] allowed individuals to vary in their propensity to form ties to different groups. If these propensities follow a gamma distribution with a mean value of 1 and

a shape parameter of $1/(\omega_k - 1)$ then the y_{ik} can be modeled with a negative binomial distribution,

$$y_{ik} \sim \text{Neg-Binom}(\text{mean} = \mu_{ik}, \text{overdispersion} = \omega_k). \tag{4.3}$$

Thus, the ω_k estimate the variation in individual propensities to form ties to people in different groups and represent one way of quantifying non-random mixing.

Then, letting $\{i \to j\}$ denote the event that ego i knows alter j and letting $\{j \in \mathscr{G}_k\}$ denote the event that alter j is a member of subpopulation $\mathscr{G}_k$, the expected number of people known in subpopulation $\mathscr{G}_k$ by ego i with degree d_i can be derived as

$$\begin{aligned}
\mu_{ik} &= \mathbb{E}\left[\sum_{j=1}^{d_i} \mathbb{I}\{j \in \mathscr{G}_k\}\right] \\
&= \sum_{j=1}^{d_i} \mathbb{E}\left[\mathbb{I}\{j \in \mathscr{G}_k\} \middle| i \to j\right] \\
&= \sum_{j=1}^{d_i} \mathbb{P}(j \in \mathscr{G}_k | i \to j) \\
&= d_i \mathbb{P}(j \in \mathscr{G}_k | i \to j) \\
&= d_i \left(\frac{N_k}{N}\right).
\end{aligned} \tag{4.4}$$

Consequently, in addition to estimating ω_k, this model also produces personal network size estimates, d_i, but this methodology has a couple drawbacks. First, this framework does not provide estimation of the dependence of social mixing on inherent agent characteristics, such as age or gender. Secondly, the degree estimates from this model are susceptible to biases due to transmission errors and barrier effects.

Transmission errors occur when a respondent knows someone in a subpopulation, but is not aware that the alter is actually in that subpopulation. For example, a respondent might know a man who has prostate cancer, but might not know that he has prostate cancer. Certain subpopulations may have higher rates of transmission errors due to reasons such as social stigma (e.g. women who have had an abortion) or political reasons (e.g. men who are in the NRA). In general, these errors are difficult to

quantify because the amount of information respondents have about their connections is unknown [Killworth et al., 2006].

Barrier effects occur whenever some individuals systematically know more (or fewer) members of a specific subpopulation than would be expected under random mixing (i.e. non-random mixing). For example, since people tend to know others of similar age and gender [McPherson et al., 2001], a 25-year old woman probably knows more women who have recently had an abortion than would be predicted just based on her personal network size and the number of women who have had an abortion. Similarly, an 80-year old man probably knows fewer than would be expected under random mixing.

4.2.2 The McCormick et al. [2010] non-random mixing model

McCormick et al. [2010] resolved the issues found in the Zheng et al. [2006] overdispersion model by making a couple important improvements. First, they restricted their analysis of ARD to questions in which $\mathscr{G}_k$ are only first names. This removed transmission errors, a common problem in previous ARD studies, because individual i must know individual j's name if j is indeed "known" by i under the McCarty et al. [2001] definition. Most crucially, however, McCormick et al. [2010] allowed the expected number known μ_{ik} to depend not on the overall prevalence of $\mathscr{G}_k$ in the network, but specifically on the prevalence of $\mathscr{G}_k$ within specific age and gender demographics. Combining these subprevalences with the known age and gender of the respondents, McCormick et al. [2010] were then able to effectively remove barrier effects as well.

Specifically, by assuming that egos of certain ages and genders mix differently with alters of other ages and genders, the McCormick et al. [2010] framework models non-random mixing by age and gender. The mixing patterns are modeled by first

letting each individual i's age $\mathscr{A}_i$ belong to a discrete age category

$$\mathscr{A}_i \in \{0-20, 21-40, 41-60, 61+\}. \tag{4.5}$$

Additionally, an individual i's gender g_i is allowed to take on one of two numeric values

$$g_i = \begin{cases} 1 & \text{if individual } i \text{ is female} \\ 2 & \text{if individual } i \text{ is male.} \end{cases} \tag{4.6}$$

With these quantities defined, the probability of alter j being in age category $\mathscr{A}_j$ and gender g_j, given that ego i in age category $\mathscr{A}_i$ and gender g_i knows alter j ($\{i \to j\}$), becomes

$$p(\mathscr{A}_j, g_j | \mathscr{A}_i, g_i, i \to j) = M_{[(\mathscr{A}_i, g_i),(\mathscr{A}_j, g_j)]}, \tag{4.7}$$

where $M_{[(\mathscr{A}_i, g_i),(\mathscr{A}_j, g_j)]}$ is the mixing rate between egos with gender g_i in age category $\mathscr{A}_i$ and alters with gender g_j in age category $\mathscr{A}_j$. In other words, $100 \times M_{[(\mathscr{A}_i, g_i),(\mathscr{A}_j, g_j)]}\%$ of the personal network of an ego with gender g_i and age category $\mathscr{A}_i$ is expected be composed of alters with gender g_j and age category $\mathscr{A}_j$.

Treating $M_{[(\mathscr{A}_i, g_i),(\mathscr{A}_j, g_j)]}$ as elements of a matrix M, the rows of M are then constrained to sum to 1 with

$$\sum_{\mathscr{A}_j, g_j} M_{[(\mathscr{A}_i, g_i),(\mathscr{A}_j, g_j)]} = 1. \tag{4.8}$$

The expected number of people known in $\mathscr{G}_k$ by ego i then becomes

$$\begin{aligned}
\mu_{ik} &= \sum_{j=1}^{d_i} p(j \in \mathscr{G}_k | \mathscr{A}_i, g_i, i \to j) \qquad (4.9)\\
&= d_i \mathbb{P}(j \in \mathscr{G}_k | \mathscr{A}_i, g_i, i \to j) \\
&= d_i \sum_{\mathscr{A}_j, g_j} \mathbb{P}(j \in \mathscr{G}_k, \mathscr{A}_j, g_j | \mathscr{A}_i, g_i, i \to j) \\
&= d_i \sum_{\mathscr{A}_j, g_j} \mathbb{P}(\mathscr{A}_j, g_j | \mathscr{A}_i, g_i, i \to j) \mathbb{P}(j \in \mathscr{G}_k | \mathscr{A}_j, g_j, \mathscr{A}_i, g_i, i \to j) \\
&= d_i \sum_{\mathscr{A}_j, g_j} M_{[(\mathscr{A}_i, g_i), (\mathscr{A}_j, g_j)]} \mathbb{P}(j \in \mathscr{G}_k | \mathscr{A}_j, g_j, \mathscr{A}_i, g_i, i \to j) \\
&= d_i \sum_{\mathscr{A}_j, g_j} M_{[(\mathscr{A}_i, g_i), (\mathscr{A}_j, g_j)]} \mathbb{P}(j \in \mathscr{G}_k | \mathscr{A}_j, g_j) \\
&= d_i \sum_{\mathscr{A}_j, g_j} M_{[(\mathscr{A}_i, g_i), (\mathscr{A}_j, g_j)]} \left(\frac{N_{k, \mathscr{A}_j, g_j}}{N_{\mathscr{A}_j, g_j}} \right),
\end{aligned}$$

where $N_{k,\mathscr{A}_j,g_j}$ denotes the number of people in subpopulation $\mathscr{G}_k$ and age category $\mathscr{A}_j$ with gender g_j, and $N_{\mathscr{A}_j,g_j}$ denotes the number of people in age category $\mathscr{A}_j$ with gender g_j.

Since survey respondents are typically required to be 18 years or older, this model generally assumes the existence of 3 ego age categories

$$\mathscr{A}_i \in \{18-24, 25-64, 65+\}, \qquad (4.10)$$

4 alter age categories as defined in Equation 4.5, and 2 alter and ego genders as defined in Equation 4.6. This results in M being a matrix with $3 \times 2 = 6$ rows and $4 \times 2 = 8$ columns. Accounting for the summation constraints on the rows of M, this implies that the non-random mixing model requires estimating $6 \times (7-1) = 42$ independent parameters.

After defining this new framework, McCormick et al. [2010] then applied it to survey responses from McCarty et al. [2001], estimating not only the individual degrees of the respondents d_i but also the latent non-random mixing matrix M. Figure 4.1 displays the fitted mixing matrix separated by ego age category and gender. The 6

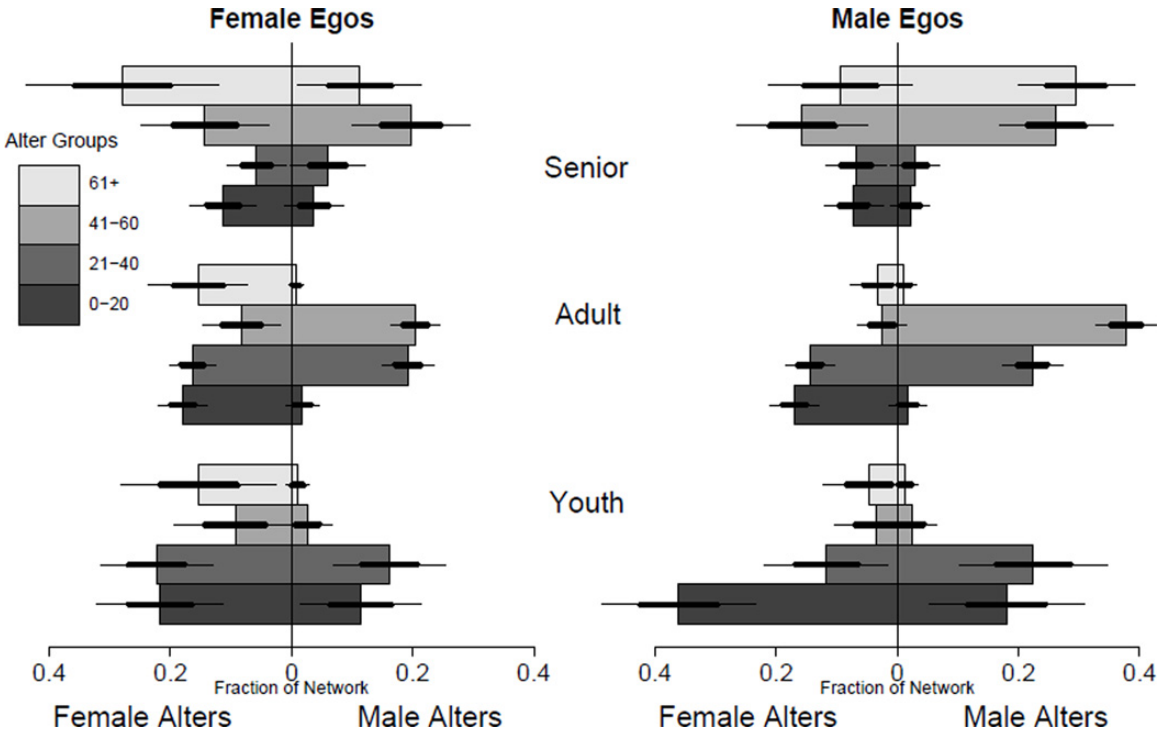

Figure 4.1: The latent non-random mixing matrix estimated from survey respondents who were asked "How many X's do you know?", where X were 12 different names. Each horizontal bar represents the magnitude of an element of the mixing matrix $M_{6\times 8}$.

plots correspond to the 6 rows of M, while the 8 bars within each plot correspond to the 8 columns of M. Most notable is that the alter age distributions for each of the six egos peak at the egos' age categories (i.e. the social networks of young egos are dominated proportionally by young alters, while the social networks of old egos are dominated proportionally by old alters), a phenomenon known as homophily in sociology. The model's ability to estimate homophily without requiring priors on the elements of the mixing matrix is a testament to the power of the latent non-random mixing matrix.

However, this model is not without its faults. In particular, Figure 4.1 also implies that the all six ego categories know more 0-20 year old females than they do 0-20

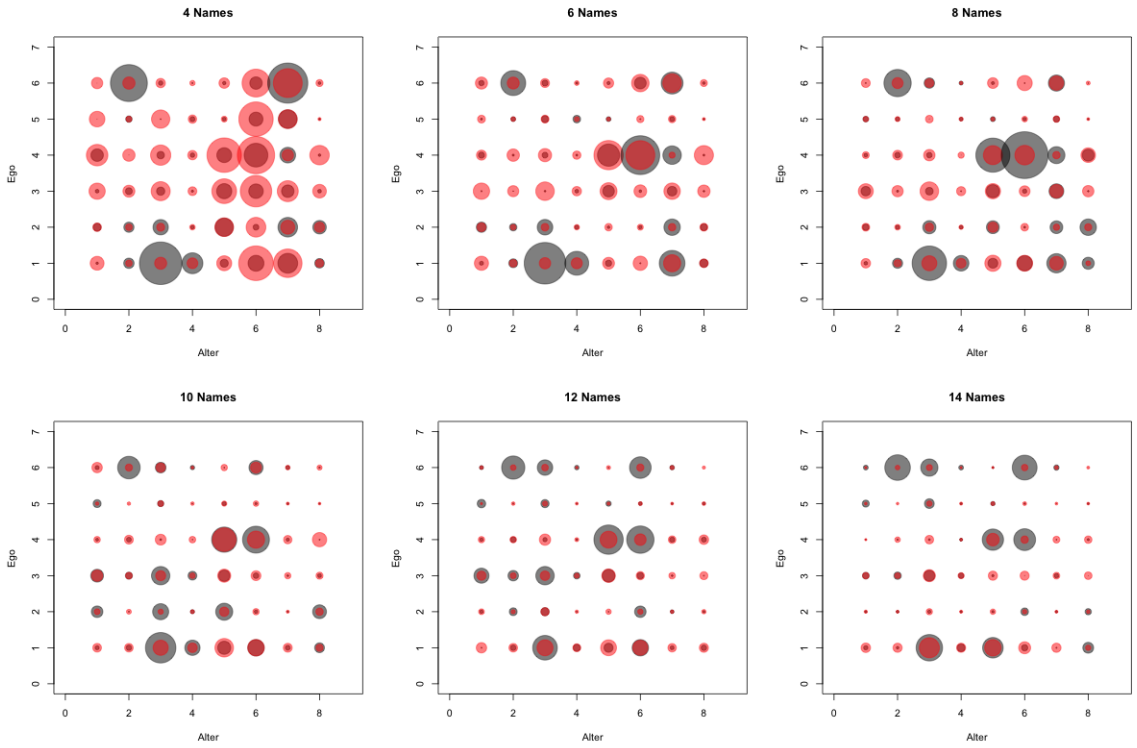

Figure 4.2: The bias and standard error of the posterior mean of each element of the mixing matrix, estimated using simulated responses to 14 names and fitting to 4, 6, 8, 10, 12, and 14 names. The size of the black circles corresponds to bias, and the size of red circles correspond to standard error.

year old males, in some cases by several orders of magnitude (e.g., the difference is dramatic for adult egos). This extreme behavior is unexpected, especially given that the national prevalence of males and females is nearly identical amongst 0-20 year olds. While it seems plausible that such a result could be due to the variability of the survey respondents (i.e. perhaps this particular survey's respondents included many individuals who know a lot of young females), our simulated data experiments point to a different explanation.

Figure 4.2 shows the results of constructing a balanced mixing matrix, simulating responses to questions about 14 names, and then trying to recover the mixing matrix

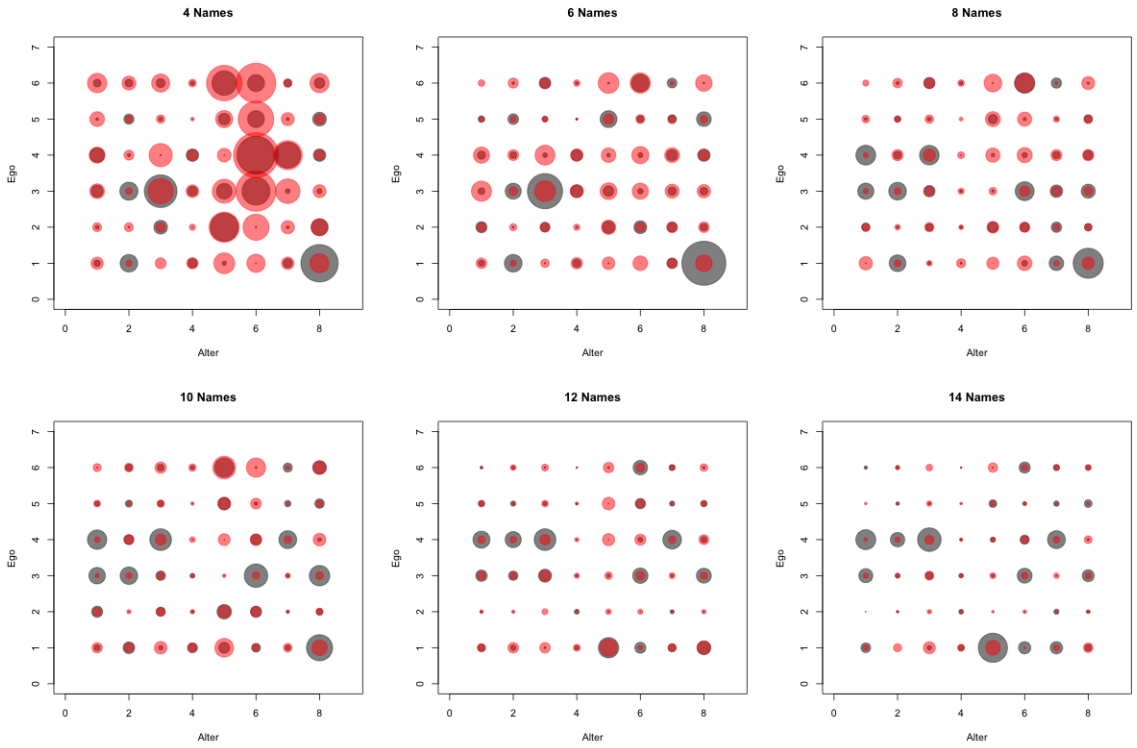

Figure 4.3: The bias and standard error of the posterior mean each element of the mixing matrix, estimated using simulated responses to 14 names and fitting to 4, 6, 8, 10, 12, and 14 names. The size of the black circles corresponds to bias, and the size of red circles correspond to standard error.

by fitting the McCormick et al. [2010] model to a subset of the responses (starting with just 4 names and going up to 14 names). The estimated mixing matrices show extreme biases for certain alter groups, even when using all 14 names. Further still, repeated response simulations from the same fixed mixing matrix produced different posterior estimates of the mixing matrix. Figure 4.3 displays the bias and variance of the mixing matrix elements from another set of responses simulated from the same balanced mixing matrix in Figure 4.2. The bias and variance patterns appear sporadic between the two sets of simulations. We believe this is due to the non-identifiability of the mixing proportions, which stems from the lack of structure imposed on M's elements.

4.3 Latent kernel representation of social mixing

As a way to place constraints on the elements of the mixing matrix, we attempt to model the social mixing rates using a more structured framework. In particular, we wish to replace the discrete histograms in Figure 4.1 with smooth continuous curves. In order to do this, we first treat age as continuous

$$a_i \in (-\infty, \infty) \tag{4.11}$$

rather than binning age into categories as in Equation 4.5. Then, to create a standard functional form that can be consistent across all ego ages, we use a Gaussian kernel as our continuous curve. The result is that we replace the 42 free parameters of the non-random mixing matrix with a non-random mixing kernel that both imposes structure and requires significantly fewer parameters to be estimated.

4.3.1 Latent mixing kernel

Probabilistically, the biggest change we make is to use a Gaussian kernel to model the likelihood of alter j being age a_j, given that ego i with gender g_i and age a_i knows

alter j and j has gender g_j. Hence, we have

$$p(a_j|g_j, a_i, g_i, i \to j) = \frac{1}{\sqrt{2\pi\lambda_{g_i g_j}}} \exp\left\{-\frac{(a_i - a_j)^2}{2\lambda_{g_i g_j}}\right\} \tag{4.12}$$
$$= \text{Normal}(a_j|a_i, \lambda_{g_i g_j}),$$

where $\lambda_{g_i g_j}$ is a latent variable that can be interpreted as the bandwidth of the age mixing kernel. Intuitively, small values of $\lambda_{g_i g_j}$ indicate that egos of gender g_i only know alters of gender g_j that are close to the ego in age, whereas larger values indicate the egos know alters of a wide range of ages.

As for gender, if we assume that gender mixing does not depend on an ego's age, then we can use a gender-only mixing matrix to model the likelihood of alter j being gender g_j, given that ego i with gender g_i and age a_i knows alter j. Here we have

$$p(g_j|a_i, g_i, i \to j) = p(g_j|g_i, i \to j) \tag{4.13}$$
$$= \rho_{g_i g_j},$$

where we constrain the rows of the gender mixing matrix to sum to 1 with

$$\sum_{g_j} \rho_{g_i g_j} = 1. \tag{4.14}$$

With the above quantities defined, the alter demographic distribution of an ego's network can now be modeled with rigid structure as

$$p(a_j, g_j|a_i, g_i, i \to j) = p(g_j|a_i, g_i, i \to j)p(a_j|g_j, a_i, g_i, i \to j) \tag{4.15}$$
$$= \rho_{g_i g_j}\text{Normal}(a_j|a_i, \lambda_{g_i g_j}).$$

4.3.2 Expectation derivation

With the kernel framework outlined, we can derive the expected number of alters known in $\mathscr{G}_k$ by ego i by replacing the sum over discrete A_j in Derivation 4.9 with an

integral over continuous a_j. The result is

$$\begin{aligned}
\mu_{ik} &= d_i \sum_{g_j} \int_{a_j} p(j \in \mathscr{G}_k, a_j, g_j | a_i, g_i, i \to j) da_j \qquad (4.16)\\
&= d_i \sum_{g_j} \int_{a_j} p(j \in \mathscr{G}_k | a_j, g_j) p(a_j, g_j | a_i, g_i, i \to j) da_j \\
&= d_i \sum_{g_j} \int_{a_j} p(j \in \mathscr{G}_k | a_j, g_j) \rho_{g_i g_j} \text{Normal}(a_j | a_i, \lambda_{g_i g_j}) da_j \\
&= d_i \sum_{g_j} \rho_{g_i g_j} \int_{a_j} p(j \in \mathscr{G}_k | a_j, g_j) \text{Normal}(a_j | a_i, \lambda_{g_i g_j}) da_j \\
&= d_i \sum_{g_j} \rho_{g_i g_j} \int_{a_j} \left(\frac{\int_a p(j \in \mathscr{G}_k | a, g_j) da}{\int_a p(j \in \mathscr{G}_k | a, g_j) da} \right) p(j \in \mathscr{G}_k | a_j, g_j) \text{Normal}(a_j | a_i, \lambda_{g_i g_j}) da_j \\
&= d_i \sum_{g_j} \rho_{g_i g_j} \left(\int_a p(j \in \mathscr{G}_k | a, g_j) da \right) \int_{a_j} \left(\frac{p(j \in \mathscr{G}_k | a_j, g_j)}{\int_a p(j \in \mathscr{G}_k | a, g_j) da} \right) \text{Normal}(a_j | a_i, \lambda_{g_i g_j}) da_j \\
&\approx d_i \sum_{g_j} \rho_{g_i g_j} \left(\int_a p(j \in \mathscr{G}_k | a, g_j) da \right) \int_{a_j} \text{Normal}(a_j | \mu_{k g_j}, \sigma^2_{k g_j}) \text{Normal}(a_j | a_i, \lambda_{g_i g_j}) da_j \\
&= d_i \sum_{g_j} \rho_{g_i g_j} \left(\int_a p(j \in \mathscr{G}_k | a, g_j) da \right) \text{Normal}(a_i | \mu_{k g_j}, \lambda_{g_i g_j} + \sigma^2_{k g_j}) \\
&= d_i \sum_{g_j} \rho_{g_i g_j} \left(\int_a p(j \in \mathscr{G}_k | a, g_j) da \right) \frac{e^{-\frac{(a_i - \mu_{k g_j})^2}{2(\lambda_{g_i g_j} + \sigma^2_{k g_j})}}}{\sqrt{2\pi(\lambda_{g_i g_j} + \sigma^2_{k g_j})}},
\end{aligned}$$

where we take the probability of alter j being in group $\mathscr{G}_k$ given alter's age a_j and gender g_j, normalize it with respect to the alter's age, approximate the normalized quantity with its discrete age counterpart, and approximate the discrete age normalized quantity with a Normal distribution so that:

$$\begin{aligned}
\frac{p(j \in \mathscr{G}_k | a_j, g_j)}{\int_a p(j \in \mathscr{G}_k | a, g_j) da} &\approx \frac{N_{k, a_j, g_j} / N_{a_j, g_j}}{\sum_a N_{k, a, g_j} / N_{a, g_j}} \qquad (4.17)\\
&\approx \text{Normal}(a_j | \mu_{k g_j}, \sigma^2_{k g_j}).
\end{aligned}$$

The approximating normal's center $\mu_{k g_j}$ and scale $\sigma^2_{k g_j}$ are estimated from $\mathscr{G}_k$'s population distributions $\frac{N_{k, a_j, g_j}}{N_{a_j, g_j}}$. These quantities are analogous, though not exactly equal, to the center and scale of the age distribution of individuals in $\mathscr{G}_k$ with gender g_j.

In practice, the survey respondents' ages are usually observed discretely. In such cases, we can approximate the integral $\int_a p(j \in \mathscr{G}_k | a, g_j) da$ by the summation

$\sum_\alpha \mathbb{P}(j \in \mathscr{G}_k|\alpha, g_j)$ over discrete age $\alpha \in \{0, 1, 2, ..\}$ so that the expression becomes

$$\begin{aligned}
\mu_{ik} &= d_i \sum_{g_j} \rho_{g_i g_j} \left(\int_a p(j \in \mathscr{G}_k|a, g_j) da \right) \frac{e^{-\frac{(a_i - \mu_{kg_j})^2}{2(\lambda_{g_i g_j} + \sigma^2_{kg_j})}}}{\sqrt{2\pi(\lambda_{g_i g_j} + \sigma^2_{kg_j})}} \\
&\approx d_i \sum_{g_j} \rho_{g_i g_j} \left(\sum_{\alpha} \mathbb{P}(j \in \mathscr{G}_k|\alpha, g_j) \right) \frac{e^{-\frac{(a_i - \mu_{kg_j})^2}{2(\lambda_{g_i g_j} + \sigma^2_{kg_j})}}}{\sqrt{2\pi(\lambda_{g_i g_j} + \sigma^2_{kg_j})}} \\
&= d_i \sum_{g_j} \rho_{g_i g_j} \left(\sum_{a} \frac{N_{k,a,g_j}}{N_{a,g_j}} \right) \frac{e^{-\frac{(a_i - \mu_{kg_j})^2}{2(\lambda_{g_i g_j} + \sigma^2_{kg_j})}}}{\sqrt{2\pi(\lambda_{g_i g_j} + \sigma^2_{kg_j})}},
\end{aligned} \tag{4.18}$$

Compared to the unrestricted elements of the McCormick et al. [2010] mixing matrix in Equation 4.7, the latent kernel framework allows us to have more control over the shape of the distributions in Figure 4.1. Additionally, this framework allows us to estimate the alter demographic distributions for egos of any arbitrary age, rather than just for egos in 6 age categories.

This latent mixing kernel framework is also more parsimonious. Indeed, our gender matrix $\rho_{2\times 2}$ only requires estimation of 2 parameters and the mixing kernel only requires estimation of 4 kernel bandwidths $\lambda_{g_i g_j}$. Consequently, this framework requires estimation of only 6 parameters whereas the McCormick et al. [2010] mixing matrix requires estimation of 42 free parameters.

4.3.3 Dependence on Alter Degree

We also considered modeling the dependence of mixing on alter degree d_j, but found that the dependence does not exist because d_j is integrated out of the expression. The

proof is as follows:

$$
\begin{aligned}
\mu_{ik} &= d_i \sum_{g_j} \int_{a_j} \int_{d_j} p(j \in \mathscr{G}_k, a_j, g_j, d_j | a_i, g_i, i \to j) dd_j da_j \qquad (4.19)\\
&= d_i \sum_{g_j} \int_{a_j} \int_{d_j} p(j \in \mathscr{G}_k | a_j, g_j, d_j, a_i, g_i, i \to j) p(a_j, g_j, d_j | a_i, g_i, i \to j) dd_j da_j \\
&= d_i \sum_{g_j} \int_{a_j} \int_{d_j} p(j \in \mathscr{G}_k | a_j, g_j) p(a_j, g_j | a_i, g_i, i \to j) p(d_j | a_j, g_j, a_i, g_i, i \to j) dd_j da_j \\
&= d_i \sum_{g_j} \int_{a_j} p(j \in \mathscr{G}_k | a_j, g_j) p(a_j, g_j | a_i, g_i, i \to j) \left(\int_{d_j} p(d_j | a_j, g_j, a_i, g_i, i \to j) dd_j \right) da_j \\
&= d_i \sum_{g_j} \int_{a_j} p(j \in \mathscr{G}_k | a_j, g_j) p(a_j, g_j | a_i, g_i, i \to j) da_j \\
&= \dots \\
&= d_i \sum_{g_j} \rho_{g_i g_j} \left(\sum_a \frac{N_{k,a,g_j}}{N_{a,g_j}} \right) \frac{e^{-\frac{(a_i - \mu_{k g_j})^2}{2(\lambda_{g_i g_j} + \sigma^2_{k g_j})}}}{\sqrt{2\pi(\lambda_{g_i g_j} + \sigma^2_{k g_j})}}.
\end{aligned}
$$

4.3.4 Kernel Bandwidth Spline

While the latent mixing kernel framework allows us to impose structure on the social mixing patterns between subpopulations, it is also restrictive because the scales $\lambda_{g_i g_j}$ of the alter age distributions depend only on ego and alter gender, and not on ego age. This is a strong assumption because it is unlikely that, for example, a female 20-year-old's male acquaintances are as tightly concentrated around age 20 as are a female 70-year-old's male acquaintances are around age 70. Indeed, it seems reasonable that the age spread of one's acquaintanceships may depend on his or her age.

To resolve this issue, we extend the model in Section 4.3.1 by allowing the kernel bandwidth to be a function of not only discrete gender but also continuous ego age $\lambda_{g_i g_j}(a_i)$. Because this bandwidth still does not depend on alter age a_j, it is a constant with respect to the integral in Derivation 4.16. As such, the solution of the integral remains unchanged with respect to all of the other terms in the derivation, and we

can essentially replace $\lambda_{g_i g_j}$ with $\lambda_{g_i g_j}(a_i)$. The resulting model is

$$\mu_{ik} = d_i \sum_{g_j} \rho_{g_i g_j} \left(\sum_a \frac{N_{k,a,g_j}}{N_{a,g_j}} \right) \frac{e^{-\frac{(a_i - \mu_{kg_j})^2}{2(\lambda_{g_i g_j}(a_i) + \sigma^2_{kg_j})}}}{\sqrt{2\pi(\lambda_{g_i g_j}(a_i) + \sigma^2_{kg_j})}}, \tag{4.20}$$

where we also impose structure on $\lambda_{g_i g_j}(a_i)$ itself so that the model remains tractable.

In order to keep $\lambda_{g_i g_j}(a_i)$ flexible without drastically increasing parameter complexity, we model the bandwidth as a spline (De Boor [1978]) in a_j. Splines are a linear combination of basis splines, or B-splines, that are uniquely defined by two parameters: (i) the polynomial degree, p; and (ii) a non-decreasing sequence of knots, $t_1, \ldots, t_q$, defined over the input range of the data that is being fit. The order of a spline family is defined as $p + 1$. B-splines of order 1 ($p = 0$) are a set of piece-wise constant functions

$$B_{n,1}(x) := \begin{cases} 1, & \text{if} \quad t_n \leq x < t_{n+1} \\ 0, & \text{otherwise,} \end{cases} \tag{4.21}$$

where $B_{n,k}$ denotes the n^{th} member of a family of B-splines of order k (or equivalently of degree $k - 1$). B-splines of higher orders are defined recursively as

$$B_{n,k}(x) := w_{n,k} B_{n,k-1}(x) + (1 - w_{n+1,k}) B_{n+1,k-1}(x), \tag{4.22}$$

where

$$w_{n,k} := \begin{cases} \frac{x - t_n}{t_{n+k-1} - t_n}, & \text{if} \quad t_n \neq t_{n+k-1} \\ 0, & \text{otherwise.} \end{cases} \tag{4.23}$$

Thus, at a given point, x, a B-spline function of order k is a linear combination of two B-splines of order $k - 1$. Consequently, a spline of order k (degree $k - 1$) with knot sequence $\mathbf{t} = t_1, \ldots, t_q$ is defined as a linear combination of the B-splines, $B_{n,k}$, corresponding with that knot sequence. The set of all such splines can be denoted as

$$S_{k,t}(x) = \left\{ \sum_{n=1}^{N} \alpha_n B_{n,k}(x), \alpha_n \in \mathbb{R} \right\} \tag{4.24}$$

$$N = |\mathbf{t}| + k - 2,$$

where N, the number of basis functions, is equal to the number of knots $|\mathbf{t}|$ plus the order k minus 2.

For our particular implementation, we use a fourth order spline with the knots set to the deciles of the population age distributions (i.e. $|\mathbf{t}| = 11$ so that $N = 11+4-2 = 13$). With this setup, and after adding an intercept term, the expression for the kernel bandwidth becomes

$$\lambda_{g_i g_j}(a_i) = \alpha_0^{g_i g_j} + \sum_{n=1}^{13} \alpha_n^{g_i g_j} B_{n,k}(a_i). \tag{4.25}$$

Using the spline framework then, there are 14 coefficients to estimate for each of the four splines $\lambda_{g_i g_j}(a_i)$, resulting in 56 parameters. The total number of parameters to estimate for this model is then 58, compared to 6 in the base model of Section 4.3.1 and 42 for the McCormick et al. [2010] discrete mixing matrix. Because this model allows us to estimate the specific bandwidth value for any continuous ego age, however, we believe the added parameter complexity is worthwhile.

In general, we recommend modeling the bandwidth's dependence on categorical variables (e.g. gender, race) by estimating a separate bandwidth for each category; while the bandwidth's dependence on continuous covariates should be estimated using splines.

4.4 Latent kernel model and computation

In this section we describe model fitting for our latent kernel model for ARD. We first describe formally our model and prior structure and conclude by presenting our model fitting algorithm.

4.4.1 Likelihood, priors and posterior

The likelihood of modeling the survey responses to the number of people known in $\mathscr{G}_k$ is unchanged from the Zheng et al. [2006] negative binomial model described in Equation 4.3.

Our first prior is for the negative binomial overdispersion ω_k. In particular, we place a prior on the inverse overdispersion $\omega'_k = 1(1/\omega_k - 1)$ so that

$$\omega'_k \sim \text{Beta}(\alpha = 4.5, \beta = 0.5), \tag{4.26}$$

where α and β are chosen so that the priors $p(\omega'_k)$ match the empirical distribution of inverse overdispersion estimates from McCormick et al. [2010].

With respect to the negative binomial expectation μ_{ik} defined in Equation 4.20, we place a structured prior on the survey respondents' degree estimates that allows us to place a latent dependence of gender and age on ego degree. This is achieved by first defining the expected degree δ_i as

$$\delta_i = \beta_1 + \beta_2 \mathbb{I}\{g_i = \text{Female}\} - e^{\beta_3}\left(\frac{a_i - \bar{a}}{\bar{a}}\right)^2, \tag{4.27}$$

where β_1, β_2, and β_3 are latent coefficients and $\bar{a}$ is the average age of the population, estimated from the population age distribution. The priors on the actual degrees d_i are then

$$\begin{aligned} \log d_i &\sim \text{Normal}(\delta_i, \eta) \\ \log \eta &\sim \text{Normal}(-0.7, 0.1) \\ \beta_1 &\sim \text{Normal}(6, 1) \\ \beta_2 &\sim \text{Normal}(0, 1) \\ \beta_3 &\sim \text{Normal}(-3, 1), \end{aligned} \tag{4.28}$$

where the priors on η, β_1, β_2, and β_3 are chosen so that the priors $p(\delta_i)$ match the empirical distribution of degree estimates from McCormick et al. [2010].

We also place a neutral prior on the gender mixing matrix's rows

$$\begin{aligned} \rho_{1\cdot} &\sim \text{Dirichlet}(1, 1) \\ \rho_{2\cdot} &\sim \text{Dirichlet}(1, 1). \end{aligned} \tag{4.29}$$

Finally, we place the following priors on the spline coefficients from Equation 4.25 for regularization

$$\begin{aligned} \alpha_0^{g_i g_j} &\sim \text{Normal}(0, 2) \\ \alpha_n^{g_i g_j} &\sim \text{Normal}(0, \tau) \\ \tau &\sim \text{Normal}^+(0, 2). \end{aligned} \tag{4.30}$$

Letting θ denote the set of all unknown parameters and letting y denote the survey responses, the posterior is then

$$\begin{aligned} \theta|y \sim & \prod_{k=1}^{K} \text{Beta}(\omega_k'|4.5, 0.5) \prod_{i=1}^{N} \text{NegBinomial}(y_{ik}|\mu_{ik}, \omega_k') \\ & \times \text{Normal}(\log \eta| - 0.7, 0.1) \prod_{i=1}^{N} \text{Normal}(\log d_i|\delta_i, \eta) \\ & \times \text{Normal}(\beta_3| - 3, 1)\text{Normal}(\beta_2|0, 1)\text{Normal}(\beta_1|6, 1)\text{Normal}^+(\tau|0, 2) \\ & \times \prod_{g_i=1}^{2} \text{Dirichlet}(\rho_{g_i \cdot}|1, 1) \prod_{g_j=1}^{2} \text{Normal}(\alpha_0^{g_i g_j}|0, 2) \prod_{n=1}^{13} \text{Normal}(\alpha_n^{g_i g_j}|0, \tau), \end{aligned} \tag{4.31}$$

which is not easy to sample from directly. Hence, we use Markov-chain Monte Carlo (MCMC) to sample from the posterior.

4.4.2 MCMC algorithm

As previously mentioned, the ties between the survey respondents and alters from the subpopulation are never directly observed. Instead, we make inferences about the expected number of people known in $\mathscr{G}_k$ using the expression obtained from the integration in Equation 4.16. The closed form solution of this integration eases our MCMC computation significantly by removing the need for numerical integration.

Since our likeilhood and priors can be expressed with standard probabilistic programming terminology, we use the No U-Turn HMC sampler of the Stan probabilistic programming language [Stan Development Team, 2016] to converge to and obtain samples from the posterior.

We now turn attention to implementing this algorithm on the survey we designed.

4.5 Results

We present degree and kernel mixing estimates from three different types of aggregate relational data (asking how many people do you know in subpopulation $\mathscr{G}_k$) gathered from the same survey:

1. responses to questions where $\mathscr{G}_k$ are 12 names
2. responses to questions where $\mathscr{G}_k$ are 8 occupations
3. both sets of responses from (1) and (2)

4.5.1 Names

We first let $\mathscr{G}_k$ be equal to the 12 subpopulations with the following names

- Linda ($\mu_1 = 63.3, \sigma_1 = 10.5$)
- Jennifer ($\mu_1 = 37.4, \sigma_1 = 10.6$)
- Karen ($\mu_1 = 56.1, \sigma_1 = 14.0$)
- Kimberly ($\mu_1 = 39.8, \sigma_1 = 13.0$)
- Emily ($\mu_1 = 28.4, \sigma_1 = 23.4$)
- Stephanie ($\mu_1 = 35.6, \sigma_1 = 14.4$)
- Mark ($\mu_2 = 49.3, \sigma_2 = 15.1$)
- Jacob ($\mu_2 = 22.7, \sigma_2 = 18.5$)
- Kevin ($\mu_2 = 38.8, \sigma_2 = 16.2$)
- Kyle ($\mu_2 = 25.6, \sigma_2 = 10.8$)
- Adam ($\mu_2 = 31.2, \sigma_2 = 16.6$)

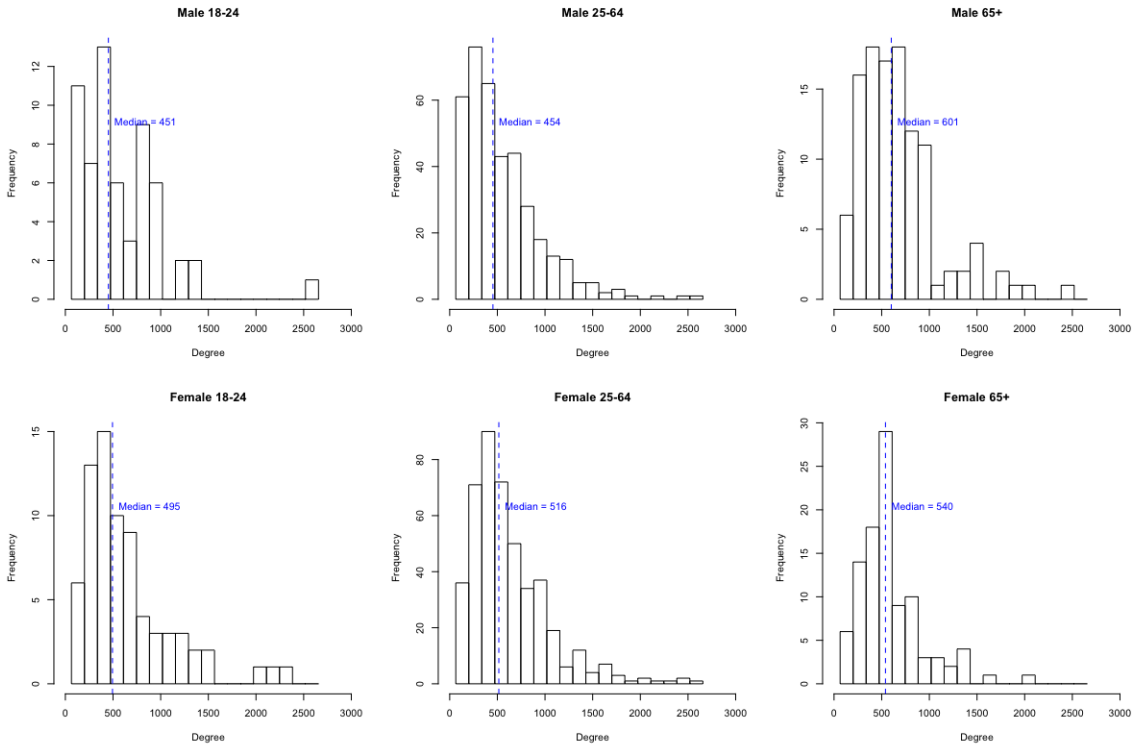

Figure 4.4: Name-based degree estimates for both genders across three age groups. A pattern of degree increasing with age is clear.

- Bruce ($\mu_2 = 62.5, \sigma_2 = 16.8$).

where the μ_{kg_j} and $\sigma^2_{kg_j}$ are defined in Equation 4.17. As per the guidelines provided by McCormick et al. [2010], we choose names that are prevalent across a broad range of ages so that we can minimize barrier effects. Since each of the above subpopulations are specific to either males or females, however, only one value of g_j is defined for each subpopulation $\mathscr{G}_k$. Consequently, in the negative binomial expectation of Equation 4.20, the first summation is taken over only one value of g_j for each subpopulation $\mathscr{G}_k$.

Figure 4.4 displays the degree estimates obtained from fitting the kernel mixing model to the names responses, for six different age-sex groups. The pattern of degree increasing with age is surprising because we usually expect degree to decrease for older individuals. However, this may partially be explained by a combination of people

retiring at increasingly older ages and in general being more active at older ages than in the past.

Figure 4.5 displays the kernel estimates for male and female egos of three different ages, each. Compared to the mixing matrix in Figure 4.1, the kernels show more regularity in their structure from ego to ego (by design). Additionally, estimating kernel bandwidth as a function of age allows us to clearly see how the age concentration of alters changes by age. Meanwhile, the gender mixing matrix, while simpler than the age-gender mixing matrix, is still able to capture the variability in network composition by gender. In particular, it appears that female egos' social networks are roughly equally split between males (51.1%) and females (49.9%), but male egos' social networks contain more males (55.5%) than females (44.5%).

Figure 4.6 displays the kernel bandwidth spline estimates by ego and alter gender. Male egos' social network age distributions are more spread out (i.e. larger bandwidth) when they first enter the work force in their late 20s, and then the networks' age distributions gradually tighten as the egos get older. Female egos' social network age distributions, on the other hand, spread out slightly earlier on, and sharply decrease going into the 30s, perhaps due to childbirth. However, after age 40, their networks' age distributions spread out again, stabilizing in the 60s.

4.5.2 Occupations

We next let $\mathscr{G}_k$ be equal to the 8 subpopulations with the following occupations

- Professor ($\mu_1 = 47.8, \sigma_1 = 15.0; \mu_2 = 48.8, \sigma_2 = 16.3$)
- Childcare Worker ($\mu_1 = 40.0, \sigma_1 = 15.3; \mu_2 = 34.0, \sigma_2 = 18.2$)
- Police Officer ($\mu_1 = 39.5, \sigma_1 = 9.2; \mu_2 = 40.6, \sigma_2 = 10.9$)
- Lawyer ($\mu_1 = 48.0, \sigma_1 = 11.3; \mu_2 = 51.2, \sigma_2 = 13.3$)
- Social Worker ($\mu_1 = 45.3, \sigma_1 = 12.9; \mu_2 = 45.0, \sigma_2 = 13.8$)

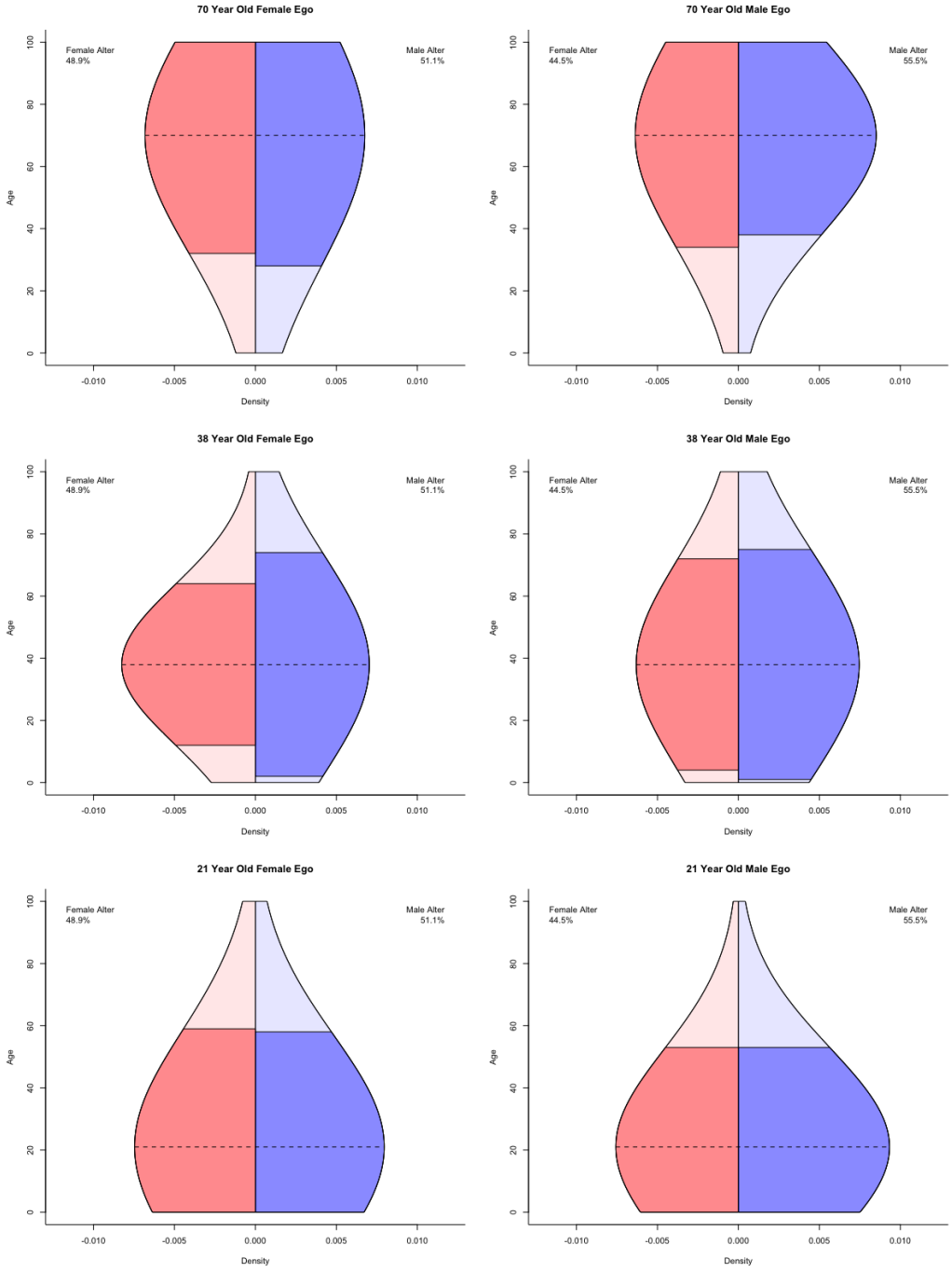

Figure 4.5: Name-based kernel estimates for male and female egos of age 21, 38, and 70. The darker regions show one standard deviation of the kernel.

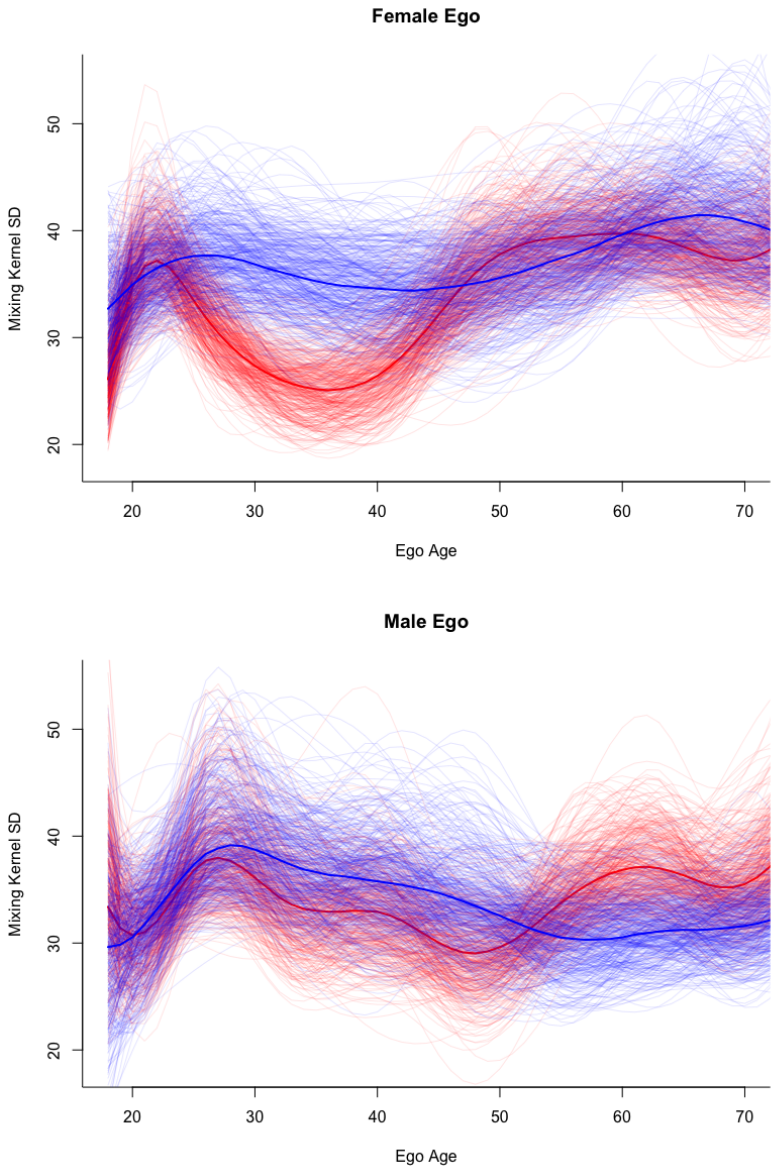

Figure 4.6: Posterior draws of name-based kernel bandwidth splines for male and female egos. Blue draws correspond to male alters while red draws correspond to female alters. Individual draws are shown with 0.1 alpha while medians are shown with 1.0 alpha.

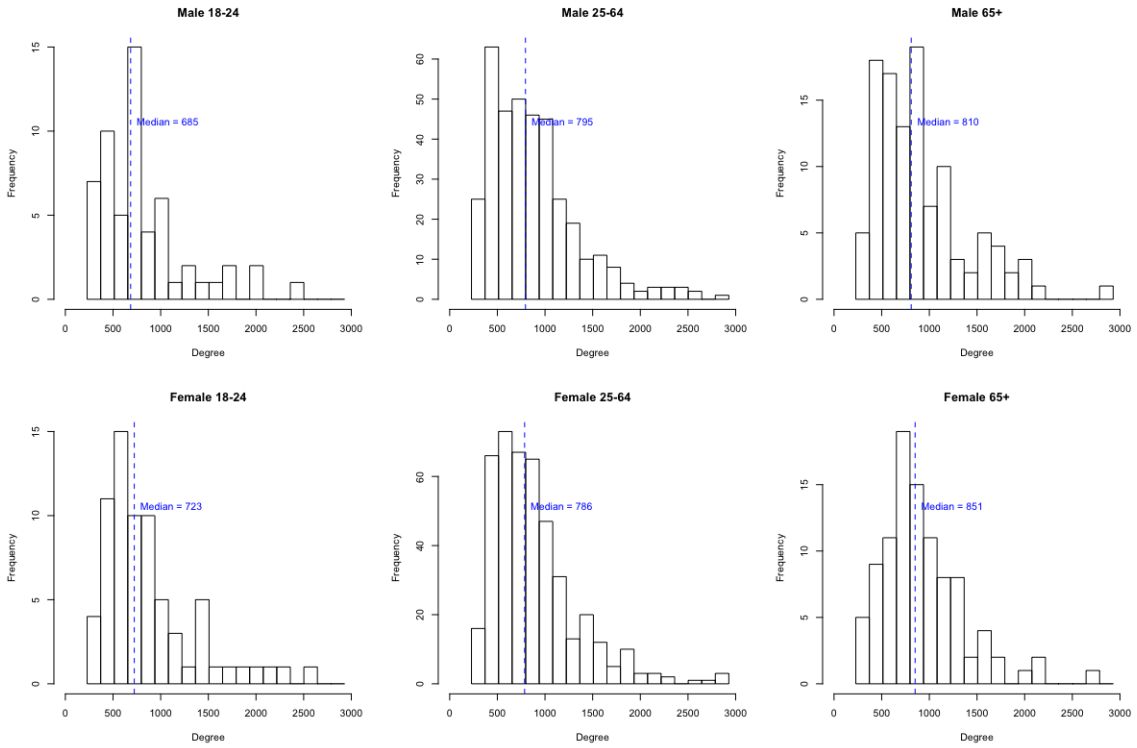

Figure 4.7: Occupation-based degree estimates for both genders across three age groups.

- Electrician ($\mu_1 = 45.7, \sigma_1 = 13.4; \mu_2 = 43.1, \sigma_2 = 12.5$)
- Cosmetologist ($\mu_1 = 42.4, \sigma_1 = 14.9; \mu_2 = 45.6, \sigma_2 = 19.5$)
- Bartender ($\mu_1 = 37.9, \sigma_1 = 13.6; \mu_2 = 38.2, \sigma_2 = 13.1$).

Unlike the names in the previous section, there are two genders of subpopulations for each occupation. As such, μ_{kg_j} and σ_{kg_j} are defined for both values of g_j, and the first summation of Equation 4.20 is taken over both values of g_j.

Figure 4.7 displays the degree estimates obtained from fitting the kernel mixing model to the occupations responses, for six different age-sex groups. As with the names, here we see a pattern of degree increasing with age. However, the the degree estimates themselves are in the 700-850 range, compared to the 450-600 range for the

names-based estimates. This is likely due to the fact that the population occupation data is only available for individuals 18 and older. As a result, the normalizing constant in Equation 4.20

$$C_{kg_j} = \sum_a \frac{N_{k,a,g_j}}{N_{a,g_j}} \tag{4.32}$$

is smaller than it is for names, which are observed in the population for all ages, including under 18 years old. Since the summation C_{kg_j} is multiplicative with respect to ego degree d_i in the negative binomial expectation of Equation 4.20, a decrease in the summation's value causes an increase in the degree estimates when all other terms in the expectation are held constant. This suggests that occupations (and subpopulations $\mathscr{G}_k$ that are age restricted in general) should not be used if one's goal is to obtain accurate degree estimates.

Figure 4.8 displays the kernel estimates for male and female egos of three different ages, each. Compared to the name-based kernels in Figure 4.5, the occupation-based kernels are more heavily male skewed for male and female alters. In particular, female egos' social networks are estimated to be composed of 60.4% males and 39.6% females, while male egos' social networks are estimated to be composed of 73.3% males and 26.7% females.

Given that our surveys include more female-dominated occupations (Childcare Worker, Social Worker, and Cosmetologist) than male-dominated occupations (Police and Electrician), this skew in the kernels has a few possible explanations. From a mathematical perspective, with respect to the negative binomial expectation in Equation 4.20, the consequence is that for each individual, the occurrence of larger normalizing summations C_{kg_j} for female-dominated occupations in the likelihood causes a decrease in the value of $\rho_{g_i g_j}$ for female alters (since an individual's degree d_i is fixed for all $\rho_{g_i g_j}$ and C_{kg_j}).

From the perspective of transmission errors, it may be the case that men disclose their occupations more often than women do, which would cause underreporting in the survey responses, particularly to questions about female-dominated occupations.

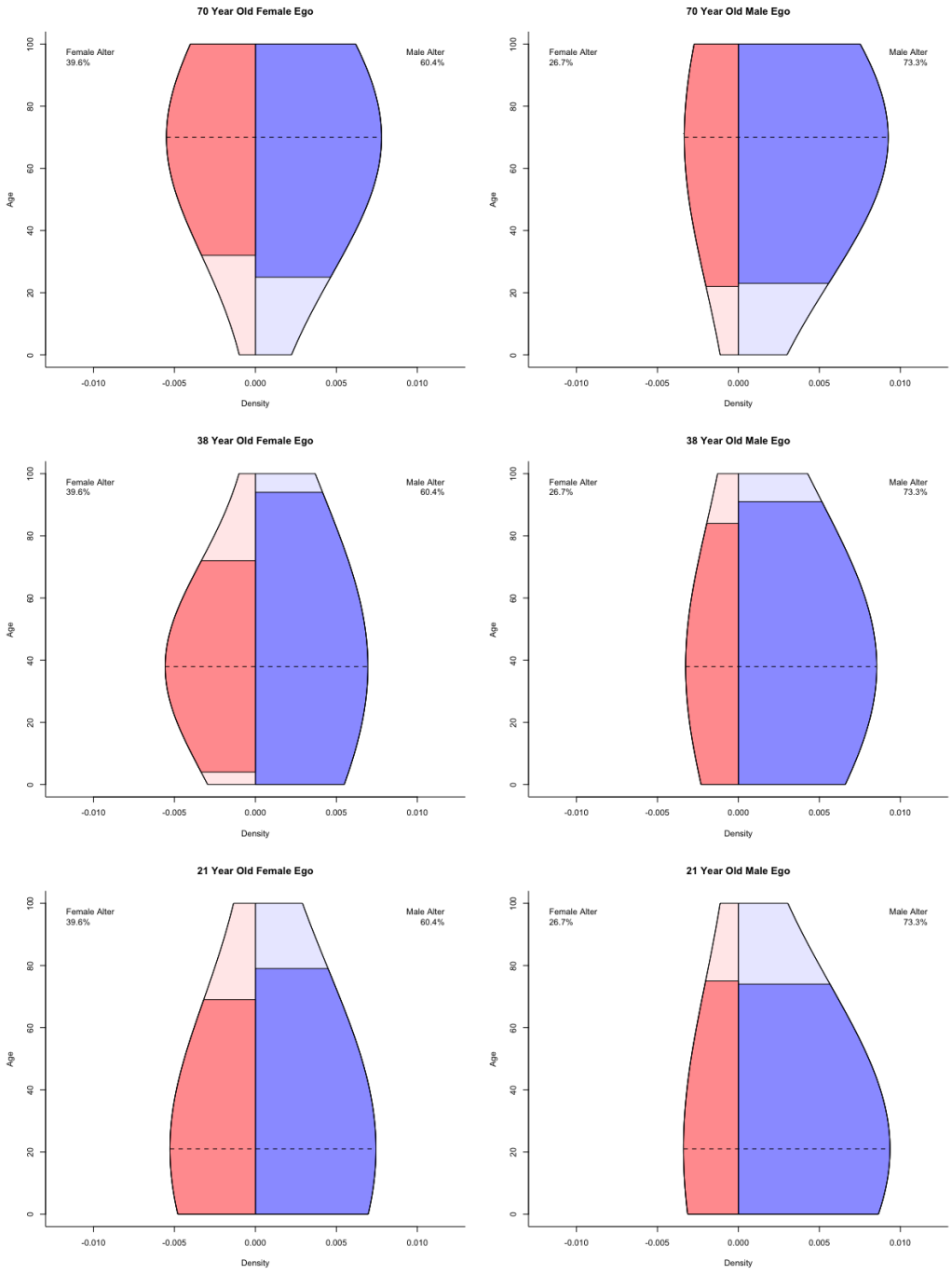

Figure 4.8: Occupation-based kernel estimates for male and female egos of age 21, 38, and 70. The darker regions show one standard deviation of the kernel.

Alternatively, it may also be the case that people with certain occupations disclose their occupations to others at a relatively lower rate (perhaps due to stigma or political reasons), and that in our case those lower-disclosing occupations happen to be female-dominated.

While it is difficult to determine which of the above issues is contributing most to the skewness, we suggest that when including survey questions about subpopulations that contain both males and females (e.g. occupation, alma mater, company), one choose subpopulations $\mathscr{G}_k$ that have roughly the same value of the normalizing summation C_{kg_j} for both genders. Such subpopulations will contribute to the likelihood in a more balanced way, and will be less likely to suffer from transmission effects correlated with gender.

Figure 4.9 displays the kernel bandwidth spline estimates by ego and alter gender. While the male ego splines share similar shapes to those estimated using names, the female ego splines look quite different. The splines also all show a stronger overall negative trend than the names-based estimates. Most saliently, however, the posterior variance of the splines themselves is much larger than that of the names-based estimates. This is because the expected number known μ_{ik} appears in the likelihood only 8 times (due to the 8 occupations) for each individual, whereas in the names-based approach the μ_{ik} appeared 12 times. As a result, the effective sample size is smaller, and the spline estimates are noisier.

4.5.3 Combined

We finally let $\mathscr{G}_k$ take on all of the names and occupations that were used in Sections 4.5.1 and 4.5.2. Thus, each survey respondent appears 20 times in the likelihood, once for a response to each of the 12 names and once for a response to each of the 8 occupations.

Figure 4.10 displays the degree estimates obtained from fitting the kernel mixing model to both names and occupations responses. The relative trends between the age

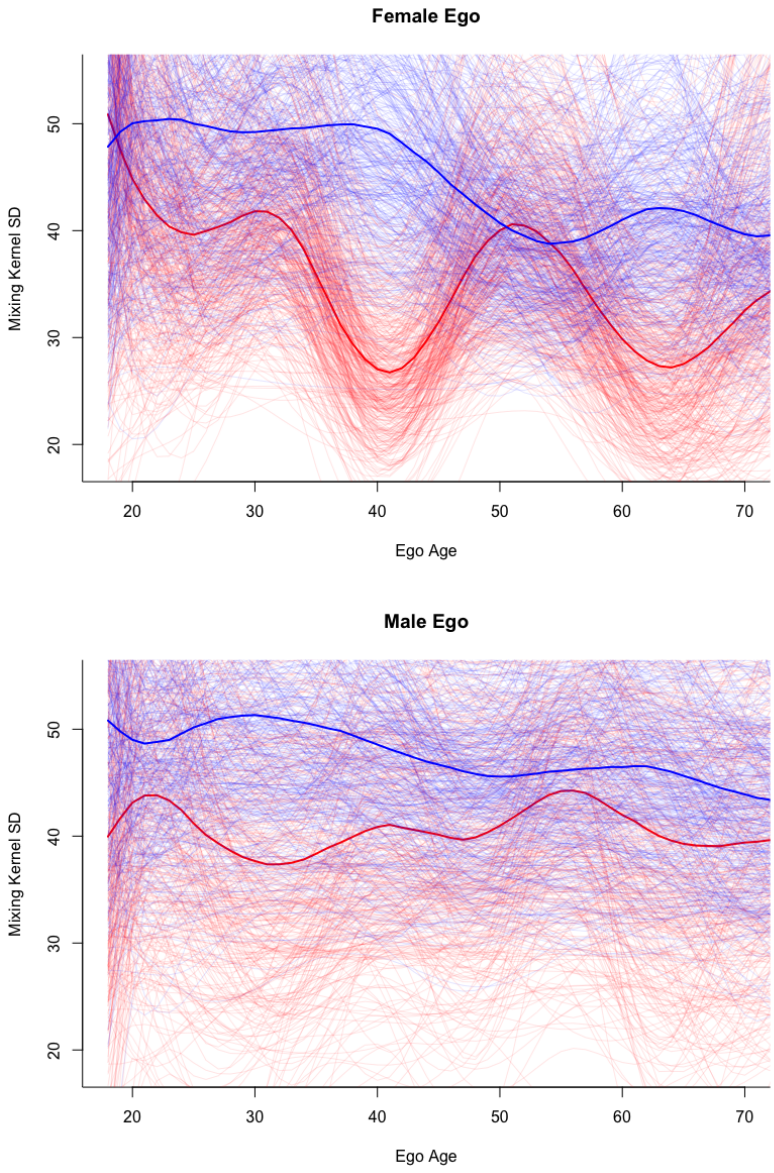

Figure 4.9: Posterior draws of occupation-based kernel bandwidth splines for male and female egos. Blue draws correspond to male alters while red draws correspond to female alters. Individual draws are shown with 0.1 alpha while medians are shown with 1.0 alpha

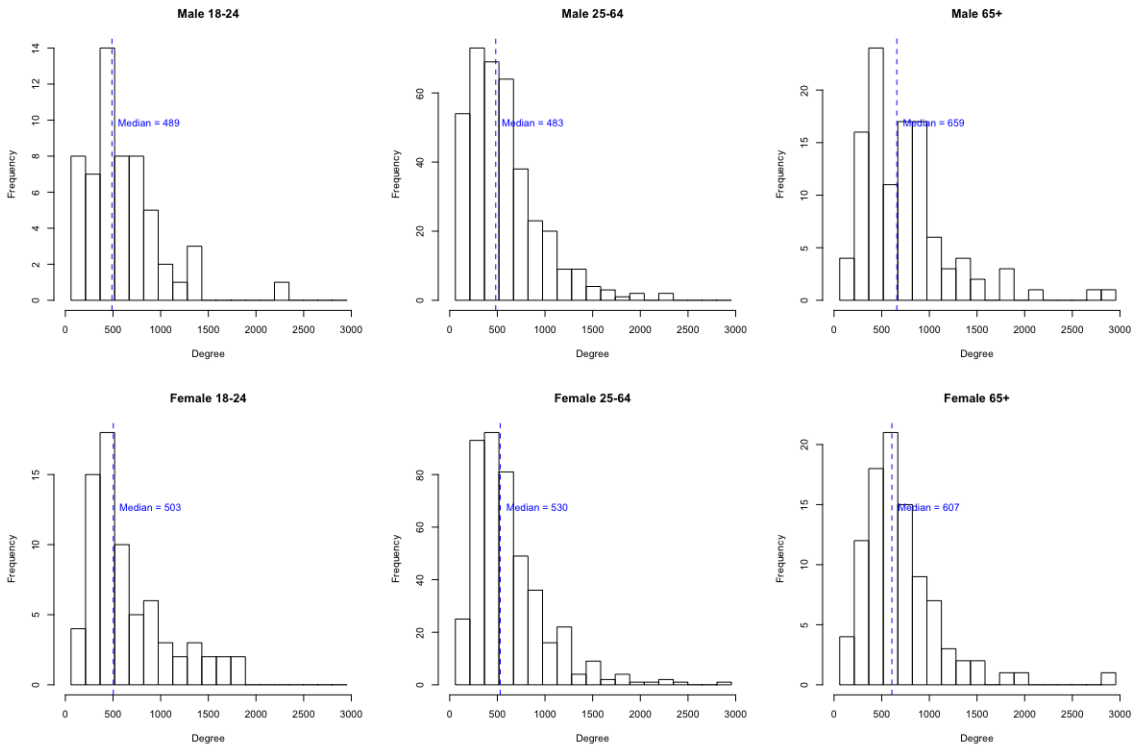

Figure 4.10: Name and occupation based degree estimates for both genders across three age groups.

groups are similar to the names-based estimates in Figure 4.4 and the occupations-based estimates in Figure 4.7. However, these degree estimates are in the 500-700 range, lying in between the names-only and occupations-only estimates' ranges. This middle ground is expected as the names and occupations responses both appear with d_i in the likelihood. Additionally, because the names responses appear 50% more often (there are 12 names and 8 occupations), the degree estimates are more similar to those obtained by the names-based approach in Section 4.5.1.

Figure 4.11 displays the kernel estimates for male and female egos of three different ages, each. The gender mixing rates are in between those estimated by the names-based approach in Section 4.5.1 and those estimated by the occupations-based approach in Section 4.5.2. Namely, female egos' social networks are estimated to be composed of 52.9% males and 47.1% females, while male egos' social networks are estimated to be composed of 58.2% males and 41.8% females. Overall, these proportions are more similar to the names-based estimates because there are more names than occupations in the likelihood. While these estimates seem more consistent with reality than the occupations estimates, it is not clear if this combined approach is preferable to simply using responses to names alone.

Figure 4.12 displays the kernel bandwidth spline estimates by ego and alter gender. The overall shapes of the splines are quite similar to those estimated by the names-based approach in Figure 4.6, though the increase in the bandwidth for older female egos is more pronounced here. Furthermore, the posterior variance of the splines is smaller due to the fact that occupations responses are also included in the likelihood.

4.6 Discussion

This paper presents a latent kernel framework for social mixing using ARD. We show that, through the choice of a particular family of models, we can produce an interpretable representation of latent mixing patterns in survey data.

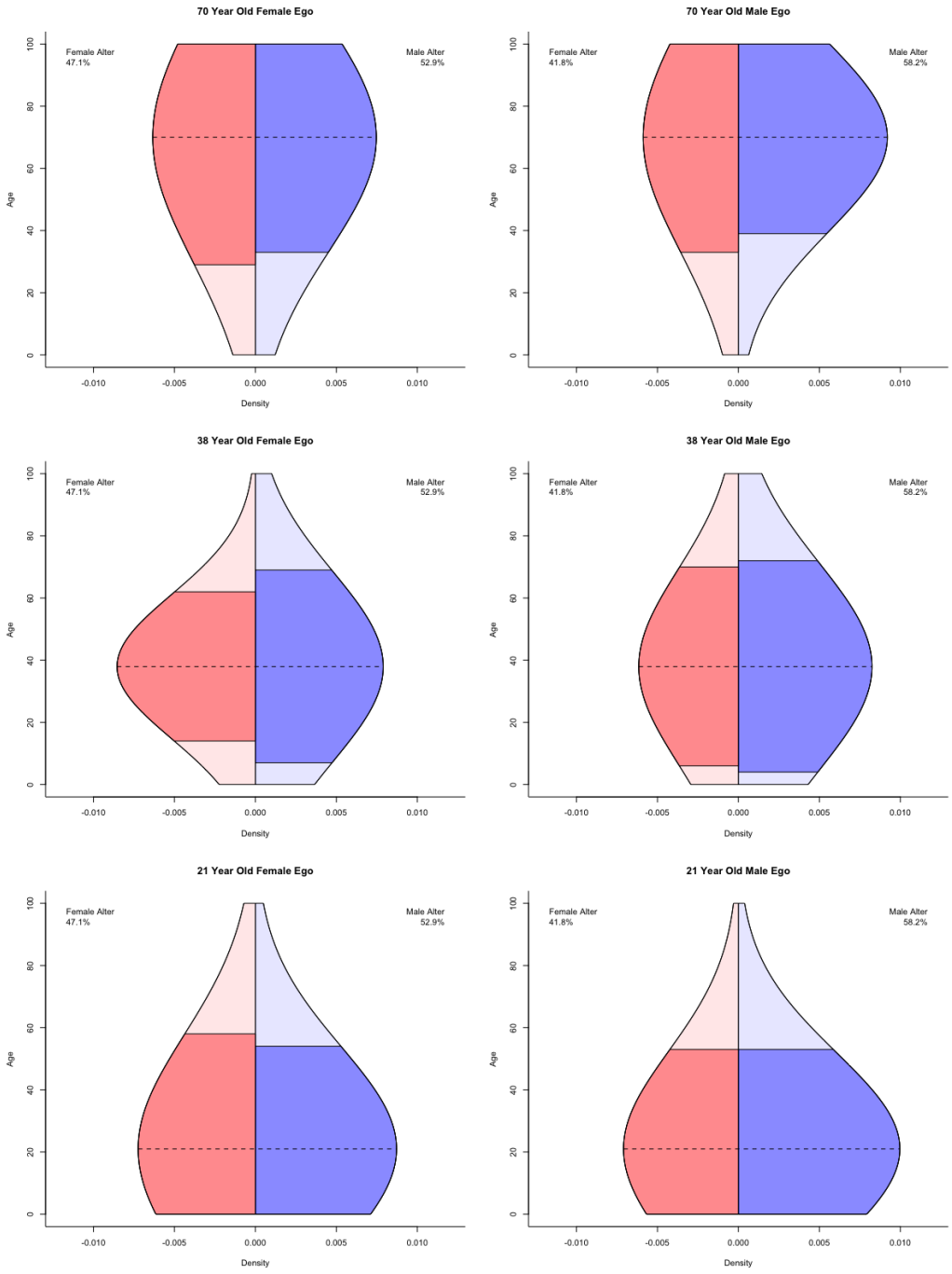

Figure 4.11: Name and occupation based kernel estimates for male and female egos of age 21, 38, and 70. The darker regions show one standard deviation of the kernel.

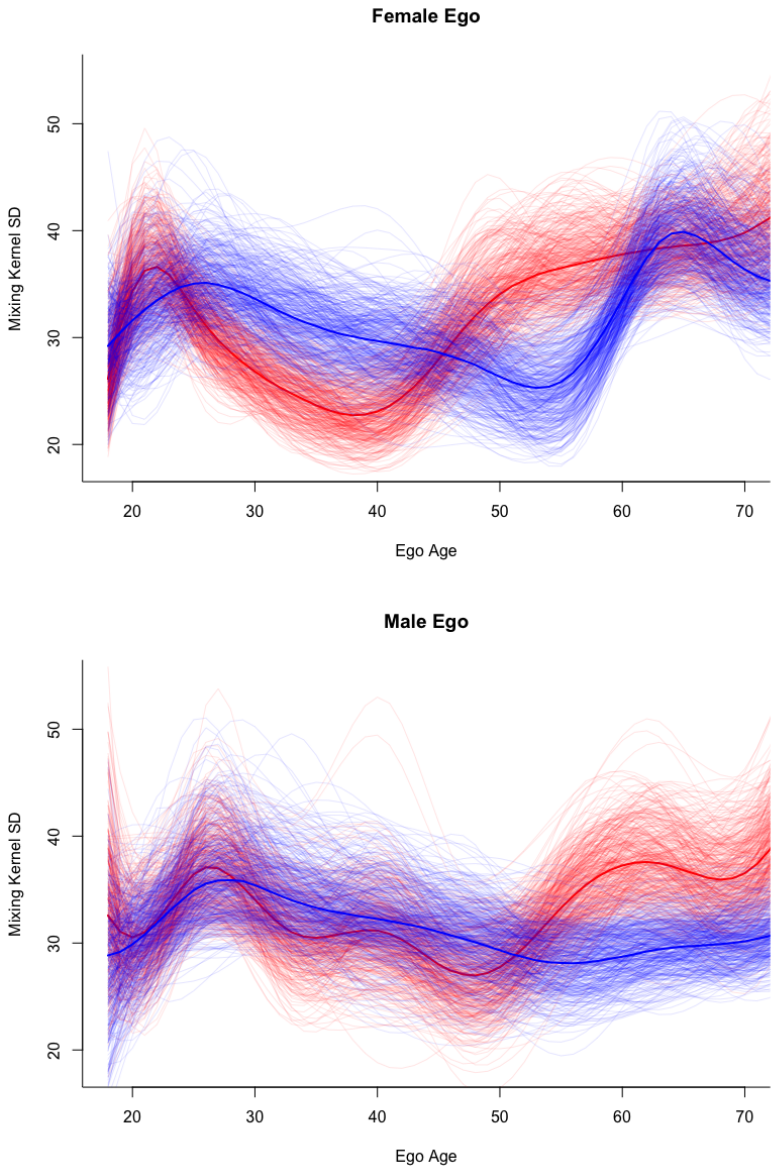

Figure 4.12: Posterior draws of name-and-occupation-based kernel bandwidth splines for male and female egos. Blue draws correspond to male alters while red draws correspond to female alters. Individual draws are shown with 0.1 alpha while medians are shown with 1.0 alpha

In scaling up the ARD to the full network we make assumptions about respondents' abilities to recall their network. First, we assume accurate recall from respondents' complete networks, which is typically not valid for moderate to large subpopulations (e.g. people named Michael). We also assume that the respondent has accurate information about the group membership of each of their alters. This issue, known in sociology literature as transmission errors, is more common with some groups than others (e.g. acquaintances of a woman who has had an abortion may not know the woman's status). In some cases it is possible to select subpopulations that minimize transmission errors (e.g. first names), yet this remains an open problem in cases where groups of interest are prone to transmission errors (e.g. stigmatized occupations).

Recent work demonstrates that features of social mixing, such as homophily (tendency for actors to form ties with similar others), are distinguishable after aggregation. In particular, McCormick et al. [2010] estimate mixing patterns using an unstructured mixing matrix that requires age to be binned into categories. The latent kernel model, in contrast, provides a structured, yet flexible, framework to estimate social mixing between individuals of any age and does not suffer from identifiability issues. The addition of the kernel bandwidth spline provides further insights by estimating social mixing variability granularly over time.

In using a continuous framework for the age of the agents in a social network, we also make assumptions about various age distributions. Firstly, we assume that the normalized (with respect to discrete age) frequency of the subpopulations amongst individuals of a particular age and gender can be approximated by a normal distribution. Secondly, we assume that the Gaussian kernel need not be truncated to account for the fact that individuals cannot have a negative or very large (100+) age. If these assumptions were relaxed, the degree and social mixing estimates would likely align more closely with reality. However, the computational costs of evaluating a non-closed form negative binomial expectation could be immense, particularly in an MCMC framework.

The Gaussian kernel, however, still provides room for future flexibility despite its unimodality and symmetry. Mixtures of Gaussian distributions, for example, would provide a more flexible representation of latent features and could provide additional insights into inter-generational mixing patterns, while still maintaining computational tractability. The bandwidth spline framework could also be applied to the gender mixing rates to provide insight into how gender mixing changes by age. At the same time, additional categorial variables could be used in place of gender. For example, a spline framework applied to race could be used to understand social mixing rates between races over time.

Chapter 5

Bibliography

Karsten Ahnert and Mario Mulansky. Odeint–solving ordinary differential equations in c++. In *AIP Conference Proceedings*, volume 1389, pages 1586–1589. AIP, 2011.

Allen H Barton. Survey research and macro-methodology. *American Behavioral Scientist*, 12(2):1–9, 1968.

M Betoule, R Kessler, J Guy, J Mosher, D Hardin, R Biswas, P Astier, P El-Hage, M Konig, S Kuhlmann, et al. Improved cosmological constraints from a joint analysis of the sdss-ii and snls supernova samples. *Astronomy & Astrophysics*, 568:A22, 2014.

T. Bodnar, A. K. Gupta, and N. Parolya. Optimal linear shrinkage estimator for large dimensional precision matrix. *arXiv preprint arXiv:1308.0931*, 2014.

Yeojin Chung, Andrew Gelman, Sophia Rabe-Hesketh, Jingchen Liu, and Vincent Dorie. Weakly informative prior for point estimation of covariance matrices in hierarchical models. *Journal of Educational and Behavioral Statistics*, 40(2):136–157, 2015.

Carlos Contreras, Mario Hamuy, MM Phillips, Gastón Folatelli, Nicholas B Suntzeff, SE Persson, Maximilian Stritzinger, Luis Boldt, Sergio González, Wojtek Krzeminski,

et al. The carnegie supernova project: first photometry data release of low-redshift type ia supernovae. *The Astronomical Journal*, 139(2):519, 2010.

Carl De Boor. *A practical guide to splines*, volume 27. Springer-Verlag New York, 1978.

John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Andrew Gelman. Prior distributions for variance parameters in hierarchical models (comment on article by browne and draper). *Bayesian analysis*, 1(3):515–534, 2006.

Andrew Gelman, Aki Vehtari, Pasi Jylänki, Tuomas Sivula, Dustin Tran, Swupnil Sahai, Paul Blomstedt, John P. Cunningham, David Schiminovich, and Christian Robert. Expectation propagation as a way of life: A framework for bayesian inference on partitioned data. 2017.

Julien Guy, P Astier, S Nobili, N Regnault, and R Pain. Salt: a spectral adaptive light curve template for type ia supernovae. *Astronomy & Astrophysics*, 443(3): 781–791, 2005.

Erika T. Hamden, David Schiminovich, and Mark Seibert. The diffuse galactic far-ultraviolet sky. *The Astrophysical Journal*, 779(180):15, December 2013.

Matthew D Hoffman and Andrew Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

Richard Kessler, Andrew Becker, David Cinabro, Jake Vanderplas, Joshua A Frieman, John Marriner, Tamara M Davis, Benjamin Dilday, Jon Holtzman, Saurabh Jha,

et al. First-year sloan digital sky survey-ii (sdss-ii) supernova results: Hubble diagram and cosmological parameters. *arXiv preprint arXiv:0908.4274*, 2009.

Peter D Killworth, Eugene C Johnsen, Christopher McCarty, Gene Ann Shelley, and H Russell Bernard. A social network approach to estimating seroprevalence in the united states. *Social networks*, 20(1):23–50, 1998.

Peter D Killworth, Christopher McCarty, Eugene C Johnsen, H Russell Bernard, and Gene A Shelley. Investigating the variation of personal network size under unknown error conditions. *Sociological Methods & Research*, 35(1):84–112, 2006.

Marek Kowalski, David Rubin, Greg Aldering, RJ Agostinho, A Amadon, R Amanullah, C Balland, K Barbary, G Blanc, PJ Challis, et al. Improved cosmological constraints from new, old, and combined supernova data sets. *The Astrophysical Journal*, 686 (2):749, 2008.

MC March, R Trotta, P Berkes, GD Starkman, and PM Vaudrevange. Improved constraints on cosmological parameters from type ia supernova data. *Monthly Notices of the Royal Astronomical Society*, 418(4):2308–2329, 2011.

Christopher McCarty, Peter D Killworth, H Russell Bernard, Eugene C Johnsen, and Gene A Shelley. Comparing two methods for estimating network size. *Human organization*, 60(1):28–39, 2001.

Tyler H McCormick, Matthew J Salganik, and Tian Zheng. How many people do you know?: Efficiently estimating personal network size. *Journal of the American Statistical Association*, 105(489):59–70, 2010.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

Thomas P. Minka. *A family of algorithms for approximate Bayesian inference.* PhD thesis, Cambridge, MA, USA, 2001a.

Thomas P. Minka. Expectation propagation for approximate Bayesian inference. In J. Breese and D. Koller, editors, *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence (UAI-2001)*, pages 362–369. Morgan Kaufmann, San Francisco, Clif., 2001b.

Radford M Neal. Probabilistic inference using markov chain monte carlo methods. 1993.

Radford M Neal. Slice sampling. *Annals of statistics*, pages 705–741, 2003.

Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11), 2011.

Manfred Opper and Ole Winther. Gaussian processes for classification: Mean-field algorithms. *Neural Computation*, 12(11):2655–2684, 2000.

Saul Perlmutter, G Aldering, G Goldhaber, RA Knop, P Nugent, PG Castro, S Deustua, S Fabbro, A Goobar, DE Groom, et al. Measurements of ω and λ from 42 high-redshift supernovae. *The Astrophysical Journal*, 517(2):565, 1999.

Adam G Riess, Alexei V Filippenko, Peter Challis, Alejandro Clocchiatti, Alan Diercks, Peter M Garnavich, Ron L Gilliland, Craig J Hogan, Saurabh Jha, Robert P Kirshner, et al. Observational evidence from supernovae for an accelerating universe and a cosmological constant. *The Astronomical Journal*, 116(3):1009, 1998.

Hikmatali Shariff, Xiyun Jiao, Roberto Trotta, and David A van Dyk. Bahamas: New analysis of type ia supernovae reveals inconsistencies with standard cosmology. *The Astrophysical Journal*, 827(1):1, 2016.

Stan Development Team. *Stan modeling language: User's guide and reference manual*, 2016. Version 2.14.0, http://mc-stan.org/.

N Suzuki, D Rubin, C Lidman, G Aldering, R Amanullah, K Barbary, LF Barrientos, J Botyanszki, M Brodwin, N Connolly, et al. The hubble space telescope cluster

supernova survey. v. improving the dark-energy constraints above z> 1 and building an early-type-hosted supernova sample. *The Astrophysical Journal*, 746(1):85, 2012.

W Michael Wood-Vasey, Gajus Miknaitis, CW Stubbs, Saurabh Jha, AG Riess, Peter M Garnavich, Robert P Kirshner, Claudio Aguilera, Andrew C Becker, JW Blackman, et al. Observational constraints on the nature of dark energy: first cosmological results from the essence supernova survey. *The Astrophysical Journal*, 666(2):694, 2007.

Tian Zheng, Matthew J Salganik, and Andrew Gelman. How many people do you know in prison? using overdispersion in count data to estimate social structure in networks. *Journal of the American Statistical Association*, 101(474):409–423, 2006.

Onno Zoeter and Tom Heskes. Gaussian quadrature based expectation propagation. In Robert Cowell and Zoubin Ghahramani, editors, *International Workshop on Artificial Intelligence and Statistics (AISTATS)*, volume 10, 2005.

Appendix A

Stan Code for Distributed Expectation Propagation

We present here the Stan code for distributed expectation propagation.

```
EP = function(fit = NULL,
        data = NULL,
        J = 360,
        K = 6,
        prior_Mu = log(c(200, 250, 5, .5, 50, 7, 1, 50, .5)),
              prior_Sigma_inv = diag(9) * 6/10,
              S = 10,
              SMOOTH = 0.9,
              randomSites = FALSE,
              parallel = TRUE,
              mc_iter = 100){

  # Extract the data
  x <- data$x;
  y <- data$y;
  bin <- data$bin;
  P <- length(prior_Mu);
  prior_Sigma_inv_Mu <- prior_Sigma_inv %*% prior_Mu;

  # Suffle groups among sites if needed
  if (randomSites) {
```

```
  bin_order <- bin_samp <- sample(unique(bin), replace = FALSE);
} else {
  bin_order <- bin_samp <- 1:max(bin);
}

# Initialize global and local natural parameters
Sigma_k_inv_Mu <- Sigma_k_inv <- Sigma_inv <- Sigma_inv_Mu <- Sigma_k_inv_tilt <-
    Post_Sigma <- list();
Post_Mu <- matrix(0, ncol = P, nrow = S + 1);
eta_j <- a_j <- tilt_fits <- list();
for(s in 1:S) { eta_j[[s]] <- a_j[[s]] <- matrix(0, nrow = P/2, ncol = length(bin_
    samp)); }

for(k in 1:K) Sigma_k_inv_Mu[[k]] <- rep(0, P);
for(k in 1:K) Sigma_k_inv[[k]] <- diag(P) * 0;
Sigma_inv[[1]] <- prior_Sigma_inv;
Sigma_inv_Mu[[1]] <- prior_Sigma_inv_Mu;

Post_Sigma[[1]] <- solve(Sigma_inv[[1]]);
Post_Mu[1,] <- prior_Mu;
init_data <- list();

# timers
init_time <- proc.time();
k_times <- matrix(0, nrow = K, ncol = S);

# The actual algorithm...
for(s in 1:S){
  # Update natural parameters from previous iteration
  if (parallel) {
    Sigma_inv[[s+1]] <- prior_Sigma_inv;
    Sigma_inv_Mu[[s+1]] <- prior_Sigma_inv_Mu;
  } else if (s > 1) {
    Sigma_inv[[s]] <- Sigma_inv[[s-1]];
    Sigma_inv_Mu[[s]] <- Sigma_inv_Mu[[s-1]];
  }

  for(k in 1:K){
    k_times[k,s] <- proc.time()[3];
    cat(" Current Iteration Status: [", s, " out of ", S, "] \n");
    cat(" Current Partition Status: [", k, " out of ", K, "] \n");

    # 1. Update the Cavity Distribution...
```

```
Sigma_inv_cav <- Sigma_inv[[s]] - Sigma_k_inv[[k]];
Sigma_inv_Mu_cav <- Sigma_inv_Mu[[s]] - Sigma_k_inv_Mu[[k]];
Sigma_cav <- solve(Sigma_inv_cav);
Sigma_cav <- (Sigma_cav + t(Sigma_cav))/2; #preserve symmetry
Mu_cav <- as.vector(Sigma_cav %*% Sigma_inv_Mu_cav);

# 2. Find tilted distribution in Stan...
# Extract current partition of the data...
if (randomSites) {
  bin_cur <- bin_samp[((k-1)*J/K+1) : (k*J/K)];
  subset <- which(bin %in% bin_cur);
  bin_k <- bin[subset];

  bin_order_k <- order(bin_cur);
  for(b in 1:length(bin_k)) {
    bin_k[b] <- bin_order_k[which(bin_cur == bin_k[b])];
  }
} else {
  subset <- which(bin <= k*J/K & bin > (k-1)*J/K);
  bin_k <- (ceiling(bin[subset] - 1))%%(J/K) + 1;
  bin_order_k <- 1:(J/K);
}
y_k <- y[subset];
x_k <- x[subset];
B <- length(unique(bin_k));
tilt_data <- list(N = length(y_k), M = P, B = B, x = x_k, y = y_k, bin = bin_k,
     Mu_Cav = Mu_cav, Sig_Cav = Sigma_cav);

# Fit tilted distribution in Stan....
for(i in 1:4) { init_data[[i]] <- list(eta = matrix(0, nrow = P/2, ncol = J/K),
     phi = Mu_cav);}
tilt_fit <- sampling(fit, data = tilt_data, iter = mc_iter, chains = 4, init =
    init_data);
tilt_fits[[k]] <- tilt_fit;

# Extract local parameter means....
current_cols = ((J/K)*(k-1) + 1):(J/K*k)
eta_k <- apply(extract(tilt_fit)$eta, c(2,3), mean);
eta_j[[s]][, current_cols] <- eta_k[, bin_order_k];
a_k <- apply(extract(tilt_fit)$a, c(2,3), mean);
a_j[[s]][, current_cols] <- a_k[, bin_order_k];
bin_order[current_cols] <- bin_order_k;
```

```
# Extract global parameter mean and covariance matrix....
Mu_tilt <- colMeans(extract(tilt_fit)$phi);
Sigma_tilt <- matrix(0, nrow = P, ncol = P);
for(i in 1:P)
  for(j in 1:P)
    Sigma_tilt[i,j] <- cov(extract(tilt_fit)$phi[,i], extract(tilt_fit)$phi[,j
        ]);

print(diag(Sigma_cav) - diag(Sigma_tilt));

# Bias correction on covariance matrix....
n <- length(extract(tilt_fit)$phi[,1]);
Sigma_inv_tilt <- solve(Sigma_tilt) * (n-P-2)/(n-1);
Sigma_inv_Mu_tilt <- Sigma_inv_tilt %*% Mu_tilt;
Sigma_k_inv_tilt[[k]] <- Sigma_inv_tilt;

# Smooth the site update until the posterior precision is positive definite....
delta = 1;
repeat {
  # 3. Attempt to update the site distribution....
  Sigma_k_inv[[k]] <- (Sigma_inv_tilt - Sigma_inv_cav) * delta;
  Sigma_k_inv_Mu[[k]] <- (Sigma_inv_Mu_tilt - Sigma_inv_Mu_cav) * delta;

  # 4. Attempt to update g(phi) in parallel/serial....
  if (parallel) {
    Sigma_inv_proposal <- Sigma_inv[[s+1]] + Sigma_inv_cav * (1 - delta) +
        Sigma_k_inv[[k]];
    Sigma_inv_Mu_proposal <- Sigma_inv_Mu[[s+1]] + Sigma_inv_Mu_cav * (1 -
        delta) + Sigma_k_inv_Mu[[k]];

    n_negative <- length(which(diag(solve(Sigma_inv_proposal)) < 0));

    if((n_negative > 0) && (delta > 0.000001)){
      # If the update is not positiv definite, dampen the tilted contribution
          and try again....
      cat("Failure (Invalid Covariance): ...", delta, "... ");
      delta = delta * SMOOTH;
    } else if((n_negative > 0) && (delta < 0.000001)) {
      # If we've tried to dampen too many times, just discard this site's
          contribution....
      cat("\n\nSite was discarded!");
      break;
    } else {
```

```
            # Otherwise, update the site approximation for real....
            cat("\nSite was succesfully added!\n\n");
            Sigma_inv[[s+1]] <- Sigma_inv_proposal;
            Sigma_inv_Mu[[s+1]] <- Sigma_inv_Mu_proposal;
            break;
          }
        } else {
          Sigma_inv[[s]] <- Sigma_k_inv[[k]] + (1-delta) * Sigma_inv_cav;
          Sigma_inv_Mu[[s]] <- Sigma_k_inv_Mu[[k]] + (1-delta) * Sigma_inv_Mu_cav;
        }
      }

      k_times[k,s] <- proc.time()[3] - k_times[k,s];
    }

    eta_j[[s]] <- t(eta_j[[s]]);
    a_j[[s]] <- t(a_j[[s]]);

    # Convert natural parameters back into usual framework....
    if (parallel) {
      Post_Sigma[[s+1]] = solve(Sigma_inv[[s+1]]);
      Post_Mu[s+1,] = as.vector(Post_Sigma[[s+1]] %*% Sigma_inv_Mu[[s+1]]);
    } else if (s > 1) {
      Post_Sigma[[s+1]] = solve(Sigma_inv[[s]]);
      Post_Mu[s+1,] = as.vector(Post_Sigma[[s+1]] %*% Sigma_inv_Mu[[s]]);
    }
  }

  # Compute the total time elapsed....
  final_time <- proc.time();

  return(list(Post_Sigma = Post_Sigma,
              Post_Mu = Post_Mu,
              Sigma_k_inv = Sigma_k_inv,
              Sigma_k_inv_tilt = Sigma_k_inv_tilt,
              eta_j = eta_j,
              a_j = a_j,
              tilt_fits = tilt_fits,
              time = final_time - init_time,
              k_times = k_times,
              bin_samp = bin_samp,
              bin_order = bin_order));
}
```

Appendix B

Stan Code for Supernovae ODE Integration

We present here the Stan code for the supernovae analysis, that includes luminosity integration achieved by the fourth and fifth order Runge-Kutta method.

```
functions {
  // Flat luminosity integral.
  real[] luminosity_flat(real t, real[] y, real[] theta, real[] x_r, int[] x_i) {
    // theta[1] = Omega_m ; theta[2] = w
    real dydt[1];
    dydt[1] = (
                theta[1] * (1 + t)^3 +
                (1 - theta[1]) * (1 + t)^(3 * (1 + theta[2]))
              )^(-0.5);
    return dydt;
  }

  // Curved luminosity integral.
  real[] luminosity_curved(real t, real[] y, real[] theta, real[] x_r, int[] x_i) {
    // theta[1] = Omega_m ; theta[2] = Omega_L
    real dydt[1];
    dydt[1] = (
                theta[1] * (1 + t)^3 +
                theta[2] +
                (1 - theta[1] - theta[2]) * (1 + t)^2
```

```
            )^(-0.5);
    return dydt;
  }

  // Sinn function.
  real sinn(real Omega_k_, real x_) {
    real sqrt_Omega_k_;
    sqrt_Omega_k_ = sqrt(fabs(Omega_k_));
    if(Omega_k_ == 0) {
      return(x_);
    }
    else if(Omega_k_ > 0) {
      return(sinh(sqrt_Omega_k_ * x_)/sqrt_Omega_k_);
    }
    else {
      return(sin(sqrt_Omega_k_ * x_)/sqrt_Omega_k_);
    }
  }

  // Luminosity Distance.
  real lum_dist(real c_, real H_0_, real z_, real Omega_k_, real int_) {
    real dL_;
    dL_ = c_/H_0_ * (1 + z_) * sinn(Omega_k_, int_);
    return dL_;
  }

  // Distance modulus.
  real dist_mod(real c_, real H_0_, real z_, real Omega_k_, real int_) {
    real mu_;
    mu_ = 25 + 5 * log10(lum_dist(c_, H_0_, z_, Omega_k_, int_));
    return mu_;
  }
}

data {
  int<lower=1> N;                    # number of data points
  int<lower=1> N_sub;                # number of data points for the third variable

  // Data
  vector[N] z_cmb;                   # observed CMB frame redshift
  vector[N] z_hel;                   # observed heliocntric redshift
  vector[N] mB_hat;                  # observed B-band peak magnitude
  vector[N] x1_hat;                  # observed stretch correction
```

```
  vector[N] cl_hat;                    # observed color correction
  matrix[3,3] C_hat[N];                # observed covariance for mag, stretch, color
  vector[N_sub] meth_hat;              # observed metallicity
  vector[N_sub] rate_hat;              # observed log star formation rate
  vector[N_sub] age_hat;               # observed log age

  // Constants
  real c;                              # speed of light in km/s
  real H_0;                            # dimensionless Hubble parameter today

  // Universe Assumption
  int<lower=0, upper=1> is_curved;     # whether or not the universe is curved
  int<lower=0, upper=4> third_var;     # 2 = metallicity; 3 = formation rate; 4 = age

  // Prior Centers and Scales
  vector[12] pri_mu;                   # means for the priors
  vector[12] pri_sd;                   # sds for the priors
}

transformed data {
  real x_r[0];                         # real inputs to the luminosity integral
  int x_i[0];                          # integer inputs to the luminosity integral
  real ts[N,2];                        # limits of the luminosity integral
  real t0;                             # initial limit value for the ode integration
  real y0[1];                          # value of integral at initial limit value t0
  for(n in 1:N) {
    ts[n,1] = 0;
    ts[n,2] = z_cmb[n];
  }
  t0 = -0.5;
  y0[1] = 0;
}

parameters {
  // Cosmological Parameters
  real<lower=0,upper=2> Omega_m;           # total matter density (baryonic + dark)
  real<lower=0,upper=2> Omega_L;           # dark energy density
  real<lower=-2,upper=0> w;                # dark energy constant

  // Covariates
  real<lower=0> alpha;                     # coefficient of stretch covariate
  real<lower=0> beta_0;                    # coefficient of color covariate
```

```
  // Covariates for Subset
  real<lower=0> alpha_1;                  # coefficient of stretch covariate
  real<lower=0> beta_1;                   # coefficient of color covariate
  real gamma;                             # coefficient of third covariate

  // Population-Level
  real M_0;                               # mean absolute magnitude (M_0^epsilon)
  real<lower=0> sigma_res;                # sd of absolute magnitude
  real x1_0;                              # mean stretch correction (x1*)
  real log_R_x1;                          # log sd of stretch correction (R_x1)
  real cl_0;                              # mean color correction (c*)
  real log_R_cl;                          # log sd of color correction (R_c)

  // Local-Level
  vector[N] M;                            # true absolute magnitude
  vector[N] x1;                           # true stretch correction
  vector[N] cl;                           # true color correction
}

transformed parameters {
  real theta[2];                          # vectorized cosmological parameters
  real<lower=0> R_x1;                     # sd of stretch correction (R_x1)
  real<lower=0> R_cl;                     # sd of color correction (R_c)
  // Cosmological Parameters
  theta[1] = Omega_m;
  if(is_curved) {
    theta[2] = Omega_L;
  }
  else {
    theta[2] = w;
  }
  // Population Level Parameters
  R_x1 = exp(log_R_x1);
  R_cl = exp(log_R_cl);
}

model {
  real lum_limits[2,1];    # integrated luminosity limits
  real int_lum;            # integrated luminosity
  vector[N] mu;            # distance modulus
  vector[N] mB;            # true B-band peak magnitude
  vector[3] salt2_hat[N];  # observed salt2 (script D_hat)
  vector[3] salt2[N];      # true salt2 (script D)
```

```
## True B-band Peak Magnitude
// Curved luminosity integration
if(is_curved) {
  for(n in 1:N) {
    lum_limits = integrate_ode_rk45(luminosity_curved, y0, t0, ts[n,], theta, x_r,
        x_i);
    int_lum = lum_limits[2,1] - lum_limits[1,1];
    mu[n] = dist_mod(c, H_0, z_hel[n], 1 - Omega_m - Omega_L, int_lum);
  }
}
// Flat luminosity integration
else {
  for(n in 1:N) {
    lum_limits = integrate_ode_rk45(luminosity_flat, y0, t0, ts[n,], theta, x_r, x_
        i);
    int_lum = lum_limits[2,1] - lum_limits[1,1];
    mu[n] = dist_mod(c, H_0, z_hel[n], 0, int_lum);
  }
}

// Baseline Model
for(n in 1:N) {
  if (n > N_sub) {
    mB[n] = mu[n] - alpha * x1[n] + beta_0 * cl[n] + (M[n] - 19.3);
  }
  else {
    mB[n] = mu[n] - alpha_1 * x1[n] + beta_1 * cl[n] + (M[n] - 19.3);
    if (third_var == 2) {
      mB[n] = mB[n] + gamma * meth_hat[n];
    }
    else if (third_var == 3) {
      mB[n] = mB[n] + gamma * rate_hat[n];
    }
    else if (third_var == 4) {
      mB[n] = mB[n] + gamma * age_hat[n];
    }
  }
}

## Priors
// Cosmological Parameters
Omega_m   ~ normal(pri_mu[1], pri_sd[1]);
```

```
  Omega_L   ~ normal(pri_mu[2], pri_sd[2]);
  w         ~ normal(pri_mu[3], pri_sd[3]);
  // Covariates
  alpha     ~ normal(pri_mu[4], pri_sd[4]);
  beta_0    ~ normal(pri_mu[5], pri_sd[5]);
  alpha_1   ~ normal(pri_mu[4], pri_sd[4]);
  beta_1    ~ normal(pri_mu[5], pri_sd[5]);
  gamma     ~ normal(pri_mu[6], pri_sd[6]);
  // Population-Level
  M_0       ~ normal(pri_mu[7], pri_sd[7]);
  sigma_res ~ normal(pri_mu[8], pri_sd[8]);
  x1_0      ~ normal(pri_mu[9], pri_sd[9]);
  log_R_x1  ~ normal(pri_mu[10], pri_sd[10]);
  cl_0      ~ normal(pri_mu[11], pri_sd[11]);
  log_R_cl  ~ normal(pri_mu[12], pri_sd[12]);

  // Local-Level
  M ~ normal(M_0, sigma_res);
  x1 ~ normal(x1_0, R_x1);
  cl ~ normal(cl_0, R_cl);

  ## Likelihood
  for(n in 1:N) {
    salt2_hat[n,1] = mB_hat[n];
    salt2_hat[n,2] = x1_hat[n];
    salt2_hat[n,3] = cl_hat[n];
    salt2[n,1] = mB[n];
    salt2[n,2] = x1[n];
    salt2[n,3] = cl[n];
    salt2_hat[n] ~ multi_normal(salt2[n], C_hat[n]);
  }
}
```