A Study of Bootstrap Based Approximations for Posterior Distributions

Alex Zhao

Pennsylvania State University

yazhao@psu.edu

April 16, 2018

Monte Carlo Markov Chains

- MCMC methods are algorithms for sampling from a probability distribution
- By building a Markov chain where the equilibrium distribution is the desired probability distribution, a sample can be drawn simply by running the chain
- Mostly used in Bayesian statistics, generally for cases where there's a need to draw from analytically difficult posterior distributions, usually those with multi-dimensional integrals

Example: Metropolis-Hastings Algorithm

Assume we have $h(x) = c\pi(x), x \in \Omega$ as a function proportional to our posterior, and a proposal $q(x, y) = q(y|x)$ (transition kernel of irreducible Markov Chain)

1. Start with $X_0 = x_0 \in \Omega$. For $n = 0, 1, 2, \ldots$, if $X_n = x$, generate as follows:
2. Propose $y \sim q(.|x)$
3. Accept/reject proposal:
    1. $\alpha(x, y) = \begin{cases} \min\{\frac{h(y)q(x|y)}{h(x)q(y|x)}, 1\} & h(x)g(x, y) > 0 \\ 1 & o/w \end{cases}$
    2. Accept $X_{n+1} = y$ with probability $\alpha(x, y)$ or instead reject and have $X_{n+1} = x$ with probability $1 - \alpha(x, y)$

Issues with MCMC

1. Within the context of certain classes of problems, for example multinomial inverse regression (MNIR), fully Bayesian methods through Monte Carlo marginalization are prohibitively expensive (Taddy 2013)
2. Sometimes the likelihood doesn't have a good conjugate prior as in the case of the negative binomial likelihood model (Pillow and Scott, 2012)
3. Even when MCMC is feasible, sometimes there are simpler or easier ways to get estimates from the posterior distribution
    - Weighted Likelihood Bootstrap (Newton and Raftery 1994)
    - Simple parametric bootstrap (Efron 2011)

Weighted Likelihood Bootstrapping

Regular Likelihood

Given independent data $x_1, \ldots, x_n$, with each x_i having a probability density function of $f_i(x_i; \theta)$, the likelihood function we get is

$$L(\theta) = \prod_{i=1}^{n} f_i(x_i; \theta)$$

Weighted Likelihood

Given independent data $x_1, \ldots, x_n$, with each x_i having a probability density function of $f_i(x_i; \theta)$, the weighted likelihood function we get is

$$\tilde{L}(\theta) = \prod_{i=1}^{n} f_i(x_i; \theta)^{w_{n,i}}$$

Weighted Likelihood Bootstrapping, Cont'd

How are the weights determined?

- "[B]y the statistician." (Newton Raftery 1994)
- Uniform Dirichlet distribution
  1. Generate n samples from $Y_i \sim Exp(\lambda)$
  2. Create $W_{n,i} = Y_i / \bar{Y}$
  3. If need be, get $W_{n,i} \propto Y_i^{\alpha}, \alpha \neq 1$ if needed to be over or underdispersed with respect to Dirichlet
- Many other possible distributions

Raw sample of weighted likelihood bootstrap parameter estimates from repeatedly generating weight vectors and optimizing the weighted likelihood function

WLB Algorithm

1. Start with data $x_1, \ldots, x_n$ with $f_i(x_i; \theta)$
2. For j $= 1$ to total number of iterations N
   1. Generate weight vector $w_n = (w_{n,1}, \ldots, w_{n,n})$
      1. Generate $y_1, \ldots, y_n \sim Exp(\lambda)$
      2. Create $w_{n,i} = y_i / \bar{y}$, with α if necessary
   2. Optimize $\tilde{L}(\theta) = \prod_{i=1}^{n} f_i(x_i; \theta)^{w_{n,i}}$ to find "maximum likelihood" estimates for θ, $\tilde{\theta}^j$
3. Create an importance weight $\mu_j \propto r(\tilde{\theta}^j) = \pi(\tilde{\theta}^j) L_m(\tilde{\theta}^j) / \hat{g}(\tilde{\theta}^j)$
   - $\pi()$ is a prior on the parameter θ
   - $L_m()$ is the marginal likelihood for θ
   - $\hat{g}$ is the estimate of the joint density of $\tilde{\theta}$ with a normal kernel and Terrell's (1990) method of maximal smoothing
4. Sample from the discrete distribution determined by the weights (Sampling-Importance Resampling)

Parametric Bootstrapping

1. We have a Bayesian prior and want to compute its posterior distribution
2. Even without weighting the individual components of the complete likelihood, it's possible to use bootstrapping to achieve the same kind of estimates as MCMC
3. Sometimes offers an easier path towards calculating posterior distributions

Parametric Bootstrap Example

Assuming $y_i \sim N(a_0, \sigma^2), i = 1, \ldots, n$, we want to look at the variability of $\beta = (a_0, \sigma^2)$. We get our bootstrap estimates for β^* from

$$a_0^* \sim N(\hat{a}_0, \frac{\hat{\sigma^2}}{n}), \sigma^{2^*} \sim \hat{\sigma^2} \frac{\chi^2_{n-1}}{n}$$

Bayes Parameter Expected Value

With a prior $\pi(\beta)$, Bayes theorem says given $\hat{\beta}$:

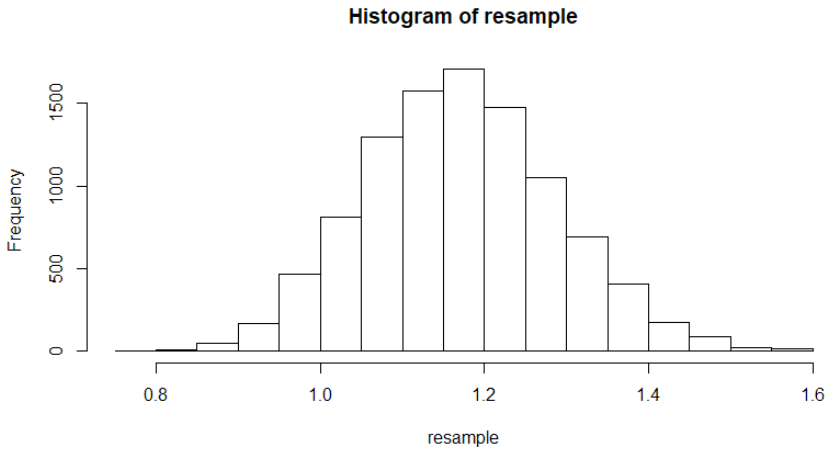
$$E\{\theta|\hat{\beta}\} = \frac{\int_\beta t(\beta)\pi(\beta)g_\beta(\hat{\beta})d\beta}{\int_\beta \pi(\beta)g_\beta(\hat{\beta})d\beta}$$

If we take $R(\beta) = \frac{g_\beta(\hat{\beta})}{g_{\hat{\beta}(\beta^*)}}$, we can replace $g_\beta(\hat{\beta})$ in the expected value with $R(\beta)g_{\hat{\beta}}(\beta^*)$, which allows us to integrate over the bootstrap density.

Parametric Bootstrap Algorithm

1. For data $y_i \sim N(a_0, \sigma^2)$, get the maximum likelihood estimates for $\hat{\beta} = (\hat{\alpha_0}, \hat{\sigma^2})$
2. For bootstrap samples j in 1 to B
   1. Draw bootstrap samples $\beta_1, \beta_2, \ldots, \beta_B$ according to $a_0^* \sim N(\hat{a}_0, \frac{\hat{\sigma^2}}{n}), \sigma^{2^*} \sim \hat{\sigma^2}\frac{\chi^2_{n-1}}{n}$
3. Calculate $R(\beta_j), \pi(\beta_j), t(\beta_j)$ for each bootstrap sample
4. Calculate $\hat{E}\{\theta|\hat{\beta}\} = \frac{\sum_{j=1}^{B} t(\beta_j)\pi(\beta_j)R(\beta_j)}{\sum_{j=1}^{B} \pi(\beta_j)R(\beta_j)}$

With our example: $n = 100, \sigma^2 = 1.25, a_0 = 1$, looking at just our bootstrap of σ^2:



General Properties of the Bootstrap to Note

- Both the weighted likelihood and Efron's parametric bootstrap approach require an importance weighing step in order to get samples from the posterior that's comparable to MCMC methods
- We are using an importance distribution, not trying to draw from the true posterior (since we don't have the true likelihood). Instead this approach is approximating the likelihood with the MLE
- Replacing the likelihood of $\theta|X$ with likelihood of $\theta|\hat{\beta}$ where $\hat{\beta}$ is a sufficient statistic
- Posterior distribution is different from MLE, hence the weighting
- Using the MLE to approximate the likelihood is adding an extra level of approximation in exchange for computational speed
- Because of the importance weighing, these bootstrap approaches can be considered the importance sampling to approximate Bayesian computing's rejection sampling

Areas of Comparison

1. Does parametric or weighted likelihood bootstrapping produce better estimates of parameters or values compared with MCMC approaches like Metropolis Hastings?
2. What is the difference in computational time and efficiency between these two categories of approaches?
3. How do these evaluations differ when it comes non-simulated data? Are there classes of problems where bootstrap can be effective when MCMC cannot, and vice versa?

Neural Models with Negative-Binomial Spiking (Pillow and Scott 2012)

- Neuroscience requires estimating neural spike responses, generally done through Poisson
- A better model is negative-binomial to account over overdispersion, but this is harder to work with analytically
- Instead of using MCMC or a Poly-Gamma distribution to sidestep this analytically intractable posterior, we instead applied parametric bootstrapping?

Multinomial Inverse Regression (Taddy 2013)

MNIR

Consider the text-sentiment contingency table with collapsed token (word) counts $x_y = \sum_{i:y_i=y} x_i$ for each $y \in Y$. Then the multinomial inverse regression model is

$$x_y \sim MN(q_y, m_y), q_{yj} = \frac{\exp(\alpha_j + y\phi_j)}{\sum_{l=1}^{p} \exp(\alpha_l + y\phi_l)}$$

Each MN is a p-dimensional multinomial distribution with size $m_y = \sum_{i:y_i=y} m_i$ and probabilities $q_y = [q_{y1}, \ldots, q_{yp}]$

Generally, each coefficient ϕ_j is estimated from LaPlace priors, which is difficult to do through Monte Carlo marginalization. Could bootstrap techniques address this?

Study EM and MCMC algorithms for Gaussian mixture model

Dongkuan Xu
College of IST, Penn State University

Gaussian mixture model (GMM)

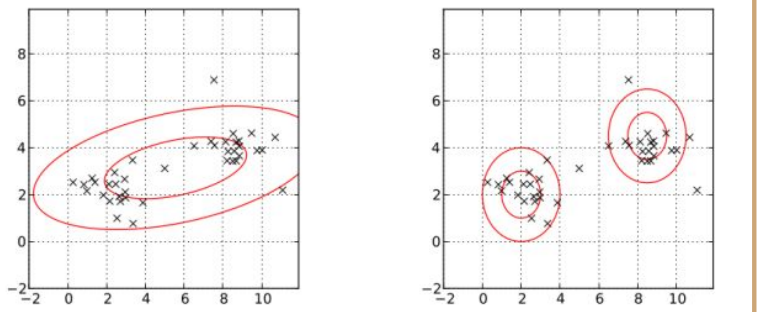
- Definition

$$p(\vec{x}) = \sum_{i=1}^{K} \phi_i \mathcal{N}(\vec{x} \mid \vec{\mu}_i, \Sigma_i)$$

$$\mathcal{N}(\vec{x} \mid \vec{\mu}_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^K |\Sigma_i|}} \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu}_i)^{\mathrm{T}} {\Sigma_i}^{-1} (\vec{x} - \vec{\mu}_i)\right)$$

$$\sum_{i=1}^{K} \phi_i = 1$$

- Related work
  - GMM based on EM algorithm [1-3]
  - Bayesian GMM based on EM or MCMC [4-5]
  - GMM with unknown mixture number [6-7]
- Application
  - Data clustering
  - Image segmentation
  - Time series analysis
  - Genetics



Motivation: data could be multimodal

Problem 1: log-likelihood is hard to calculate when MLE

- Maximum likelihood estimation
  - Likelihood function

$$\mathcal{L}(\theta\,;x)$$

  - Maximum likelihood estimate

$$\hat{\theta} \in \{\arg\max_{\theta\in\Theta} \mathcal{L}(\theta\,;x)\}$$

- Log-likelihood is hard to calculate by direct derivative: summation operations in the logarithmic operations

$$\sum_{i=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k) \right\}$$

Solution: Expectation-maximization (EM) for GMM

- Log-likelihood function

$$\sum_{i=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k) \right\}$$

- Estimation step

$$\gamma(i,k) = \frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j)}$$

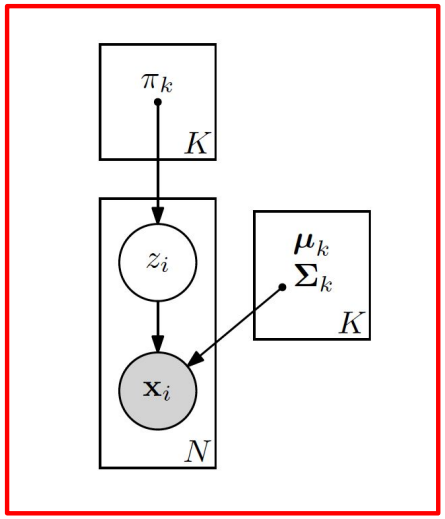
- Maximization step

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma(i,k) x_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma(i,k)(x_i - \mu_k)(x_i - \mu_k)^T$$

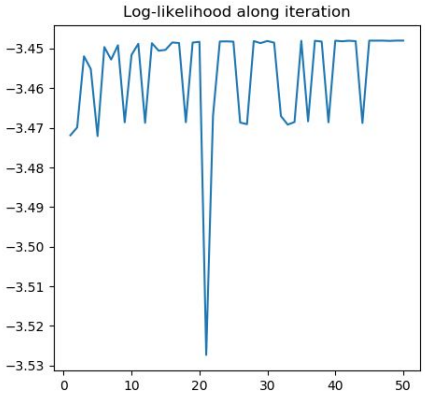
$$\pi_k = N_k / N$$

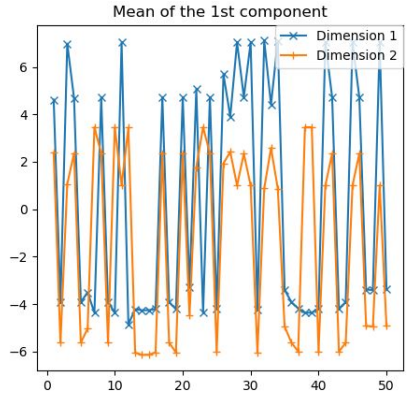
$$N_k = \sum_{i=1}^{N} \gamma(i,k)$$

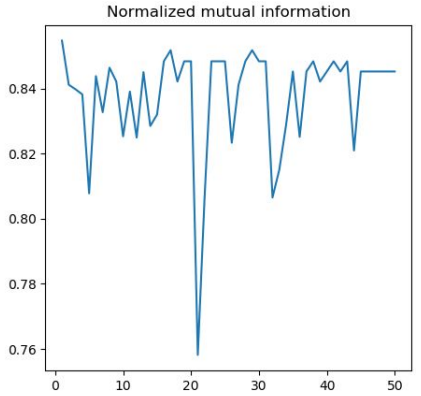


GMM with EM

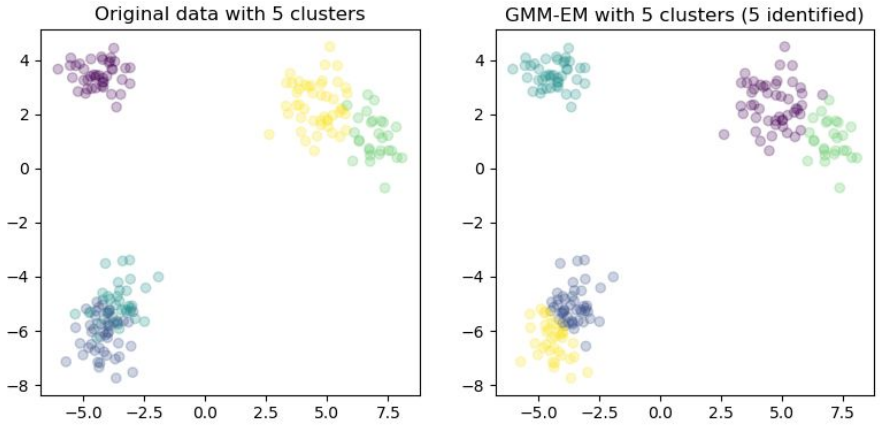
- Stopping criteria
  - (1) Maximum iteration number of EM (2) A threshold about the gain on log-likelihood
- Running time per iteration: 0.00326 s
- Iteration: (1) Log-likelihood (2) Mean of the 1st component (3) Normalized mutual information (NMI)







- Clustering results:



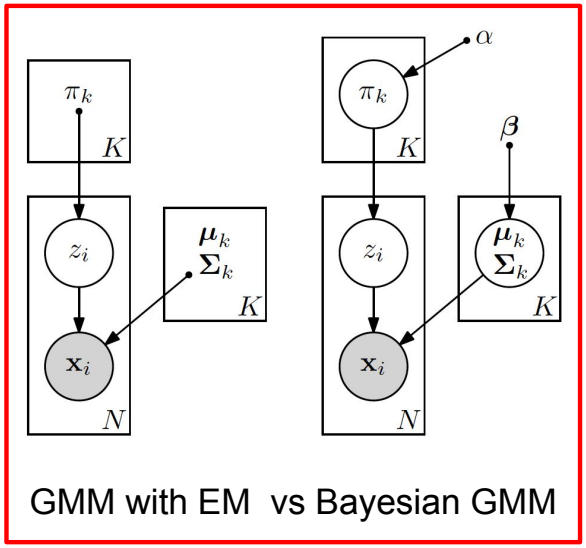
Problem 2: Too many free parameters for EM when high-dimensional

- Parameters of GMM

$$\mu_k = \frac{1}{N_k}\sum_{i=1}^{N}\gamma(i,k)x_i \qquad \Sigma_k = \frac{1}{N_k}\sum_{i=1}^{N}\gamma(i,k)(x_i-\mu_k)(x_i-\mu_k)^T \qquad \pi_k = N_k/N$$

- MLE approach based on EM may fail due to singularities or degeneracies
- A Bayesian approach alleviate these by treating $\Theta = (\boldsymbol{\pi}, \{\boldsymbol{\mu}_k\}, \{\boldsymbol{\Sigma}_k\})$ as random variables and working with distributions over Θ rather than point estimates
- Choose conjugate priors to marginalize over the parameters
  - Symmetric Dirichlet prior for mixture weights
  - Normal-inverse-Wishart (NIW) prior for component parameters

$$\begin{aligned}
\boldsymbol{\pi} &\sim \mathrm{Dir}(\alpha/K\mathbf{1}) \\
z_i &\sim \boldsymbol{\pi} \\
\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k &\sim \mathrm{NIW}(\mathbf{m}_0, \kappa_0, \nu_0, \mathbf{S}_0) \\
\mathbf{x}_i &\sim \mathcal{N}(\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i})
\end{aligned}$$

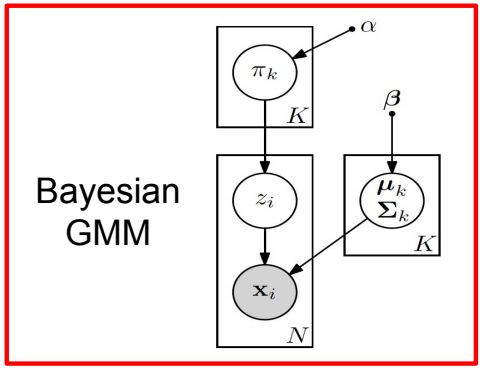


GMM with EM vs Bayesian GMM

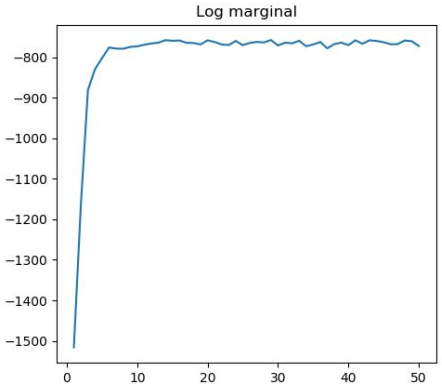
Solution: Bayesian approach for GMM

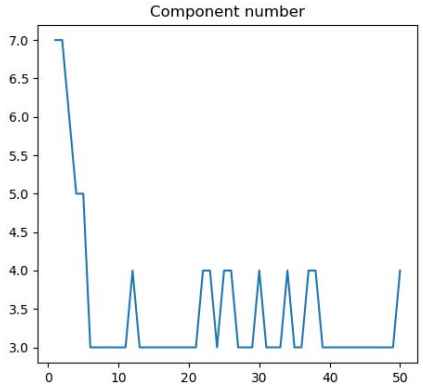
- Algorithm of Collapsed Gibbs sampler for GMM:

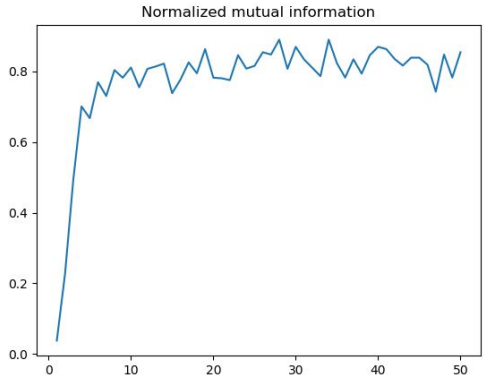
Choose an initial $\mathbf{z}$.
for T iterations **do** ▷ Gibbs sampling iterations
  for $i = 1$ to N **do**
    Remove $\mathbf{x}_i$'s statistics from component z_i. ▷ Old assignment for $\mathbf{x}_i$
    for $k = 1$ to K **do** ▷ Every possible component
      Calculate $P(z_i = k|\mathbf{z}_{\setminus i}, \mathcal{X}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \propto P(z_i = k|\mathbf{z}_{\setminus i}, \boldsymbol{\alpha})\, p(\mathbf{x}_i|\mathcal{X}_{k\setminus i}, \boldsymbol{\beta})$.
    end for
    Sample k_{new} from $P(z_i|\mathbf{z}_{\setminus i}, \mathcal{X}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ after normalizing.
    Add $\mathbf{x}_i$'s statistics to the component $z_i = k_{\text{new}}$. ▷ New assignment for $\mathbf{x}_i$
  end for
end for



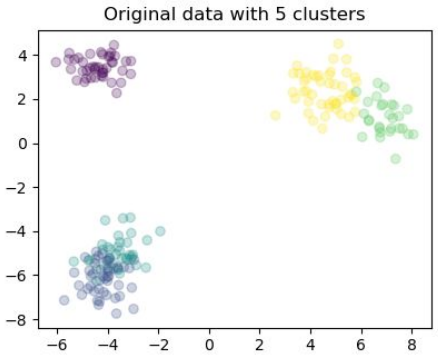
- Stopping criteria:
  - (1) Maximum iteration number of EM
  - (2) A threshold about the gain of log marginal of data and component assignments: p(X, z) (**possible**)
- Running time per iteration: 0.00492 (**higher than the one of GMM-EM**)
- Iteration: (1) Log marginal of p(X, z) (2) Normalized mutual information (NMI) (3) Component number

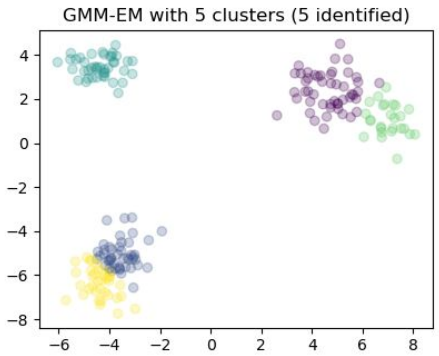


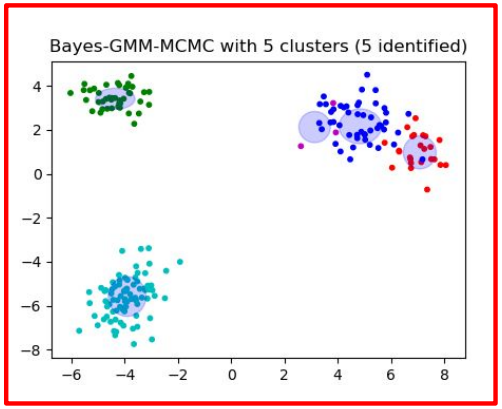




- Clustering result: **a little worse than GMM-EM**

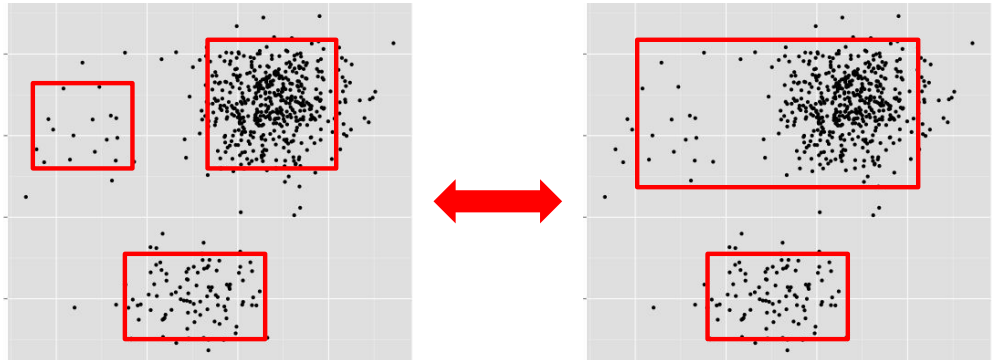






Problem 3: GMM with unknown mixture number

- Unknown mixture number



- MLE, EM do not work

$$\sum_{i=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k) \right\} \qquad \gamma(i,k) = \frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i|\mu_j, \Sigma_j)}$$

log-likelihood log-likelihood

- A number of possible solutions
  - reversible jump MCMC (Richardson and Green 1997, Gruet et al. 1999)
  - Bayes factors (Kass and Raftery 1995, Richardson and Green 1997)
  - entropy distance or K-L divergence (Mengersen and Robert 1996, Sahu and Cheng 2003)
  - birth-and-death processes (Stephens 2000a, Cappé et al. 2002)

Solution: Reversible jump MCMC for GMM with unknown mixture number

- Idea: birth and death moves
  - Add a new normal component in the mixture generated from the prior, or remove one component, according to the acceptance probability
- Assumption:
  - Competing models can be enumerable and represented as: $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots\}$
  - Current state of Markov chain is: (k, θ_k)
- GMM with RJMCMC:
  - Propose a visit from current model (k, θ_k) to next model $(\theta_{k'}, k')$
  - Accept the visit or not
  - Repeat proposing model visit until stopping criteria are met
  - Clustering based on standard GMM

Reversible jump MCMC for GMM with unknown mixture number

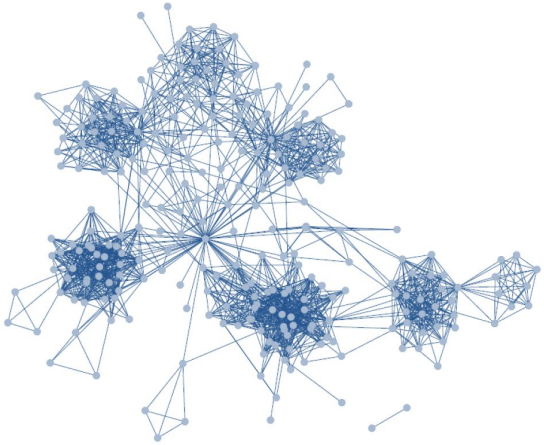
- Algorithm of GMM with RJMCMC:
  - (1) Initialize current model indicator k
  - (2) Propose a visit from mode M_k to model M_k ' with probability J(k → k')
  - (3) Sample parameter u of GMM from a proposal density $q(u|\theta_k, k, k')$
  - (4) Set $(\theta_{k'}, u') = g_{k,k'}(\theta_k, u)$, where $g_{k,k'}(\cdot)$is a bijection, where u and u ' play the role of matching the dimensions of both vectors
  - (5) The acceptance probability of the new model $(\theta_{k'}, k')$ the minimum between 1 and :

$$\underbrace{\frac{p(y|\theta_{k'}, k')p(\theta_{k'})p(k')}{p(y|\theta_k, k)p(\theta_k)p(k)}}_{\text{model ratio}} \underbrace{\frac{J(k' \to k)q(u'|\theta_{k'}, k', k)}{J(k \to k')q(u|\theta_k, k, k')} \left|\frac{\partial g_{k,k'}(\theta_k, u)}{\partial(\theta_k, u)}\right|}_{\text{proposal ratio}}$$

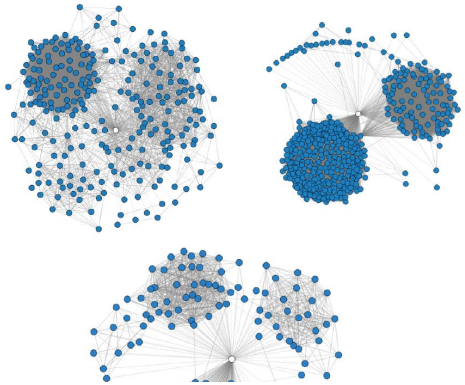
  - (6) Calculate M
  - (7) ite = ite + 1
  - (8) if ite < max or MCMC standard error > φ, then go to (2)
  - (9) output cluster assignment for all data samples

Network-structure data clustering

- Network data is ubiquitous
  - Web network
  - Social network
  - Biological network, etc.

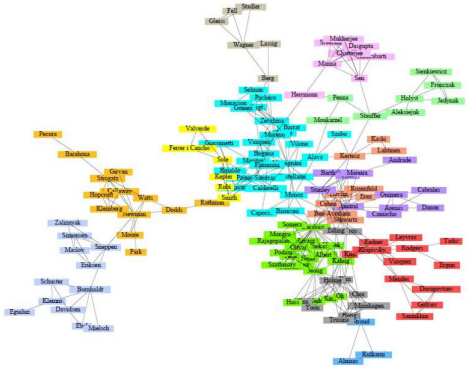


Web network

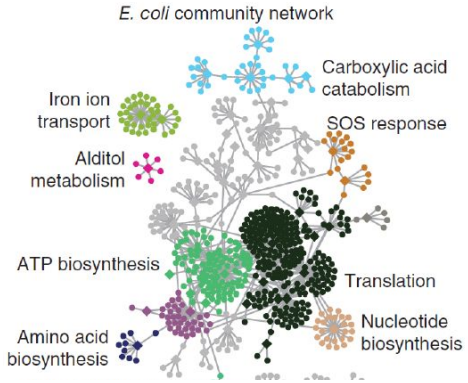


Social network

- Network clustering
  - Detect sub-networks that satisfy certain properties
  - Many connections within clusters and few connections across clusters



Co-author network



Gene network

Comparison between EM-GMM and Bayesian-GMM applied to network data clustering

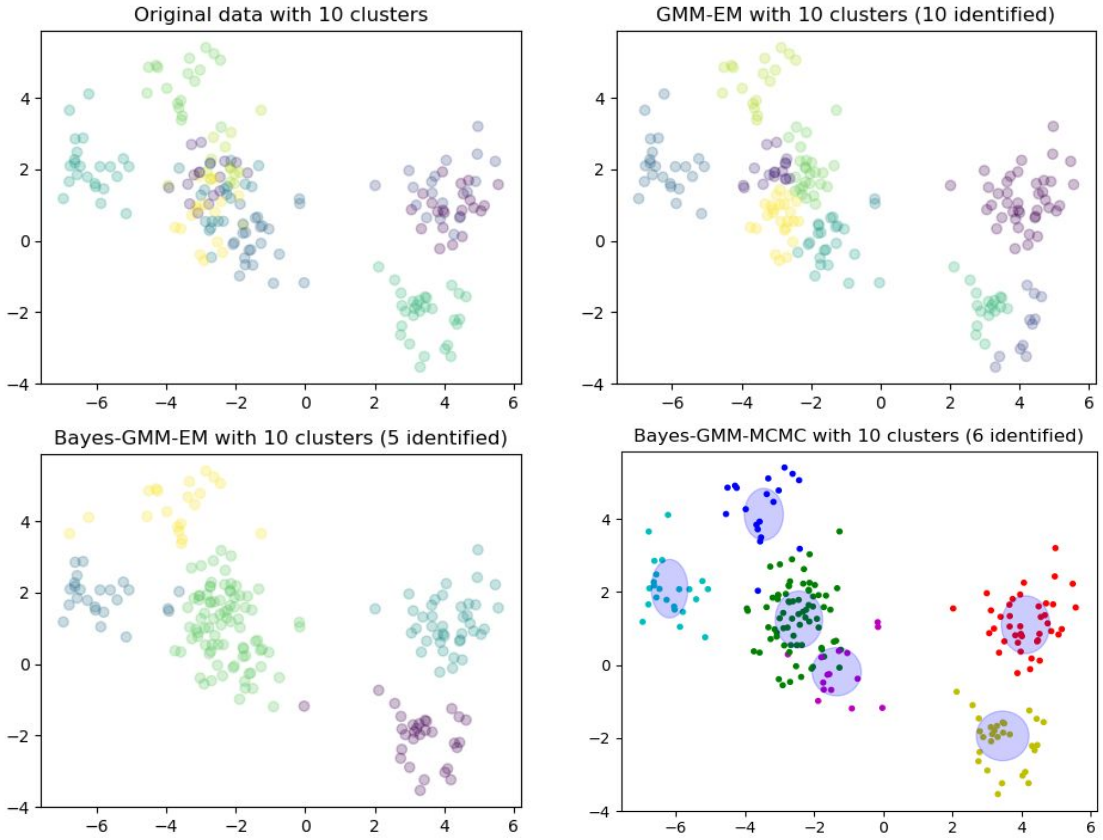
- Methods:
  - GMM based on EM
  - Bayesian GMM based on EM
  - Bayesian GMM based on MCMC

- Data set 1: Newsgroup20 (600 instances, 6 clusters, 600-dimension, sparse)

Method	GMM-EM	Bayes-GMM-EM	Bayes-GMM-MCMC
NMI	0.1902517	0.1325872	0.0450131
Identified components	6	6	2

Comparison between EM-GMM and Bayesian-GMM applied to network data clustering

- Data set 2: Synthetic (200 instances, 10 cluster, 2-dimension)



Choosing Summary Statistics for Approximate Bayesian Compution (ABC)

Nathan Wikle

STAT 540

Project Presentation, 17 April 2018

What is ABC?

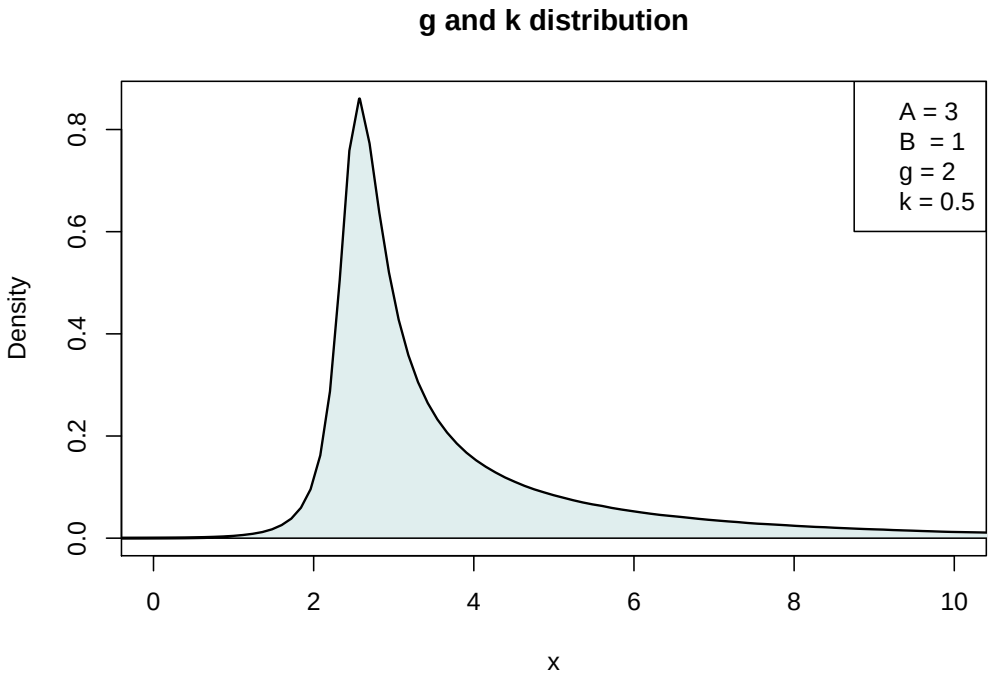
Motivating Problem: How do we perform Bayesian inference when the likelihood function $\ell(\boldsymbol{y}|\boldsymbol{\theta})$ is unavailable?

- e.g., likelihood is given as an intractable integral
  - coalescent models in population genetics
- e.g., intractable normalizing constant
  - Gibbs random fields, point process models, etc

Many Bayesian approaches to inference can no longer be applied!

However, if we can easily simulate from the likelihood,
ABC methods provide an attractive solution.

A Motivating Example



The g and k distribution:

- extension of Normal distribution that accounts for skewness and kurtosis
- CDF and pdf are unavailable in closed form, but the quantile function is given by

$$Q_{gk}(z; A, B, g, k) = A + B\left(1 + 0.8 \tanh\left(\frac{gz}{2}\right)\right) z(1 + z^2)^k, \quad z \sim N(0, 1).$$

Good candidate for ABC: 1) likelihood unavailable, 2) easy to simulate

Rejection-ABC

Some notation:

$\boldsymbol{y}$, the observed data

$\eta(\boldsymbol{y})$, summary statistics of $\boldsymbol{y}$

$\rho > 0$, a distance on η

$\epsilon > 0$, a tolerance level

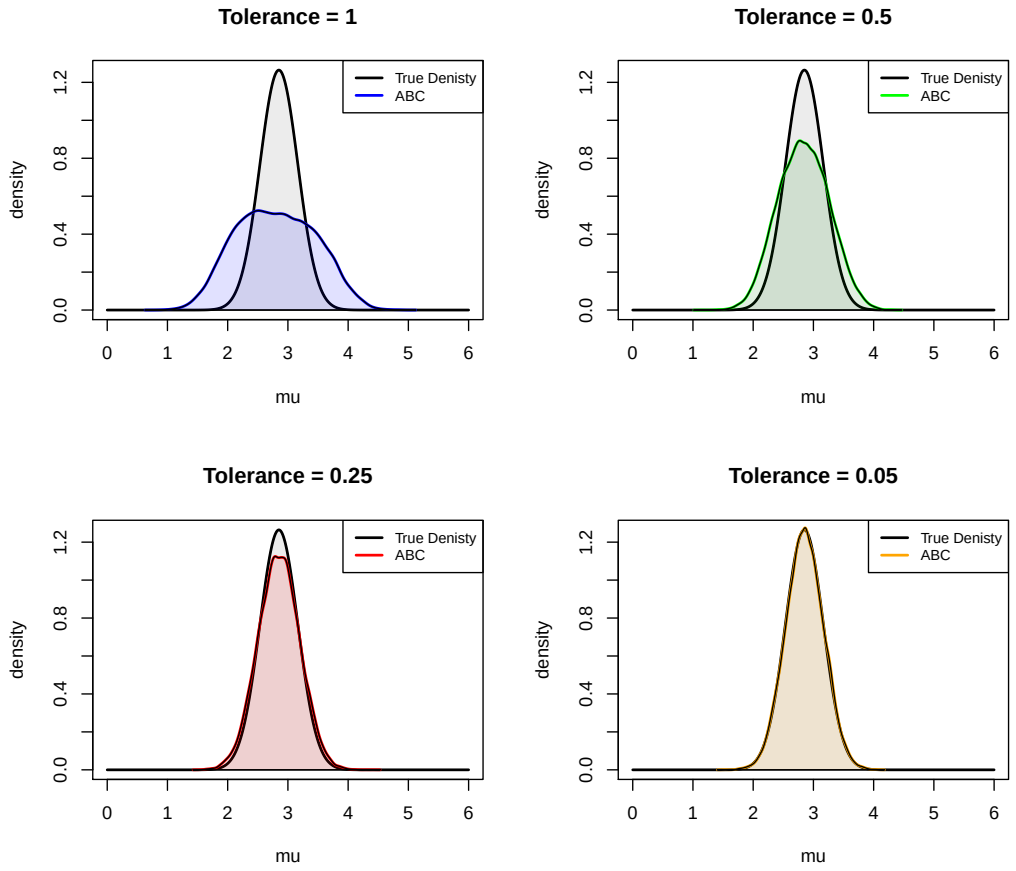
Rejection Algorithm:

- for $i = 1, 2, \ldots, N$:
  - repeat until $\rho\{\eta(\boldsymbol{z}), \eta(\boldsymbol{y})\} \leq \epsilon$
    1. Sample $\boldsymbol{\theta'}$ from $\pi(\cdot)$
    2. Simulate $\boldsymbol{z}$ from $\ell(\cdot|\boldsymbol{\theta'})$
  - set $\boldsymbol{\theta}_i = \boldsymbol{\theta'}$

- The algorithm samples from $\pi_\epsilon(\boldsymbol{\theta}, \boldsymbol{z}|\boldsymbol{y})$, a joint posterior distribution of θ and $\boldsymbol{z}$, where $\boldsymbol{z}$ is ϵ-close to $\boldsymbol{y}$.
- **The basic idea of ABC:**

$$\pi_\epsilon(\boldsymbol{\theta}|\boldsymbol{y}) = \int \pi_\epsilon(\boldsymbol{\theta}, \boldsymbol{z}|\boldsymbol{y}) d\boldsymbol{z} \approx \pi(\boldsymbol{\theta}|\boldsymbol{y}).$$

Challenges of ABC



- The success of ABC is dependent on the choice of calibration parameters.
- Optimal if η is sufficient and $\epsilon \to 0$.
- In practice, η is not sufficient, and small $\epsilon =$ larger computational time.

In general, the ABC literature is focused on:

1) choice of appropriate calibration parameters

2) efficient sampling algorithms

e.g., ABC-MCMC, ABC-SMC, etc.

Choosing Appropriate Summary Statistics

Choosing η is challenging

- problem specific
- sufficient statistics are the gold standard
- want dim(η) as close to dim($\boldsymbol{\theta}$) as possible

Three common classes of methods (Blum et al., 2013):

1) best subset selection (Joyce and Marjoram, 2008)

2) post-processing (Beaumont et al., 2002; Blum and Francois, 2010)

3) semi-automatic ABC (Fearnhead and Prangle, 2012)

Semi-Automatic ABC (Fearnhead and Prangle, 2012)

Idea: Assume interest is in point estimates of model parameters

- If $\boldsymbol{\theta}_0$ is the true parameter value, and $\hat{\boldsymbol{\theta}}$ is an estimate, choose η that minimizes
$$L(\boldsymbol{\theta}_0, \hat{\boldsymbol{\theta}}) = (\boldsymbol{\theta}_0 - \hat{\boldsymbol{\theta}})' A (\boldsymbol{\theta}_0 - \hat{\boldsymbol{\theta}})$$
- $L(\boldsymbol{\theta}_0, \hat{\boldsymbol{\theta}})$ is minimized if $\eta(\boldsymbol{y}) = E(\boldsymbol{\theta}|\boldsymbol{y})$
- Resulting $\eta(\cdot)$ is low-dimensional
- Turned our problem into finding $\eta(\boldsymbol{y}) \approx E(\boldsymbol{\theta}|\boldsymbol{y})$
- Advantage: can be applied to any ABC algorithm

Semi-automatic ABC:

1) Simulate many $(\boldsymbol{\theta}, \boldsymbol{z})$ "pilot" values

    - Simulate $\boldsymbol{\theta} \sim \pi(\cdot)$ and $\boldsymbol{z} \sim \ell(\cdot, \boldsymbol{\theta})$

2) Estimate $\eta(\boldsymbol{z}) \approx E(\boldsymbol{\theta}|\boldsymbol{z})$

3) Use $\eta(\boldsymbol{z})$ as summary statistic for ABC

More on Semi-Automatic ABC

The authors use **linear regression** on the simulated $\{(\boldsymbol{\theta}, \boldsymbol{z})\}$ to estimate $E(\boldsymbol{\theta}|\boldsymbol{z})$.

- $\theta_i = E(\theta_i|\boldsymbol{z}) + \epsilon_i = \beta_0^{(i)} + \boldsymbol{\beta}^{(i)} f(\boldsymbol{z}) + \epsilon_i$, for each θ_i.

My idea: What if we use regularization methods and nonlinear models to estimate $E(\boldsymbol{\theta}|\boldsymbol{z})$? In particular, I considered:

- **LASSO:** minimize $RSS + \lambda \sum_{j=1}^{p} |\beta_j|$
- **Ridge:** minimize $RSS + \lambda \sum_{j=1}^{p} \beta_j^2$
- **Random Forests:** bootstrap aggregation of regression trees

Note: Although the authors don't discuss these extensions, regularization and nonlinear models have been used in post-processing of ABC data for some time (e.g., Beaumont et al. (2002), Blum and Francois (2010), Blum et al. (2013)).

My Work

Motivation:

- Methods are easy to implement in R.
- May lead to better predictions than OLS solution
- May avoid overfitting the initial pilot run.
- In some cases, can help deal with collinearity in $f(\boldsymbol{z})$.
- Can handle large number of covariates.

I compared the performance using two examples:

- A toy example: Normal likelihood with conjugate priors
  - possible to compare performance to the true posterior
  - less computational cost, used Rejection-ABC
- The g and k distribution:
  - true posterior is not available; compare to results in paper
  - computing cost is noticeable, used ABC-MCMC

Normal Toy Example

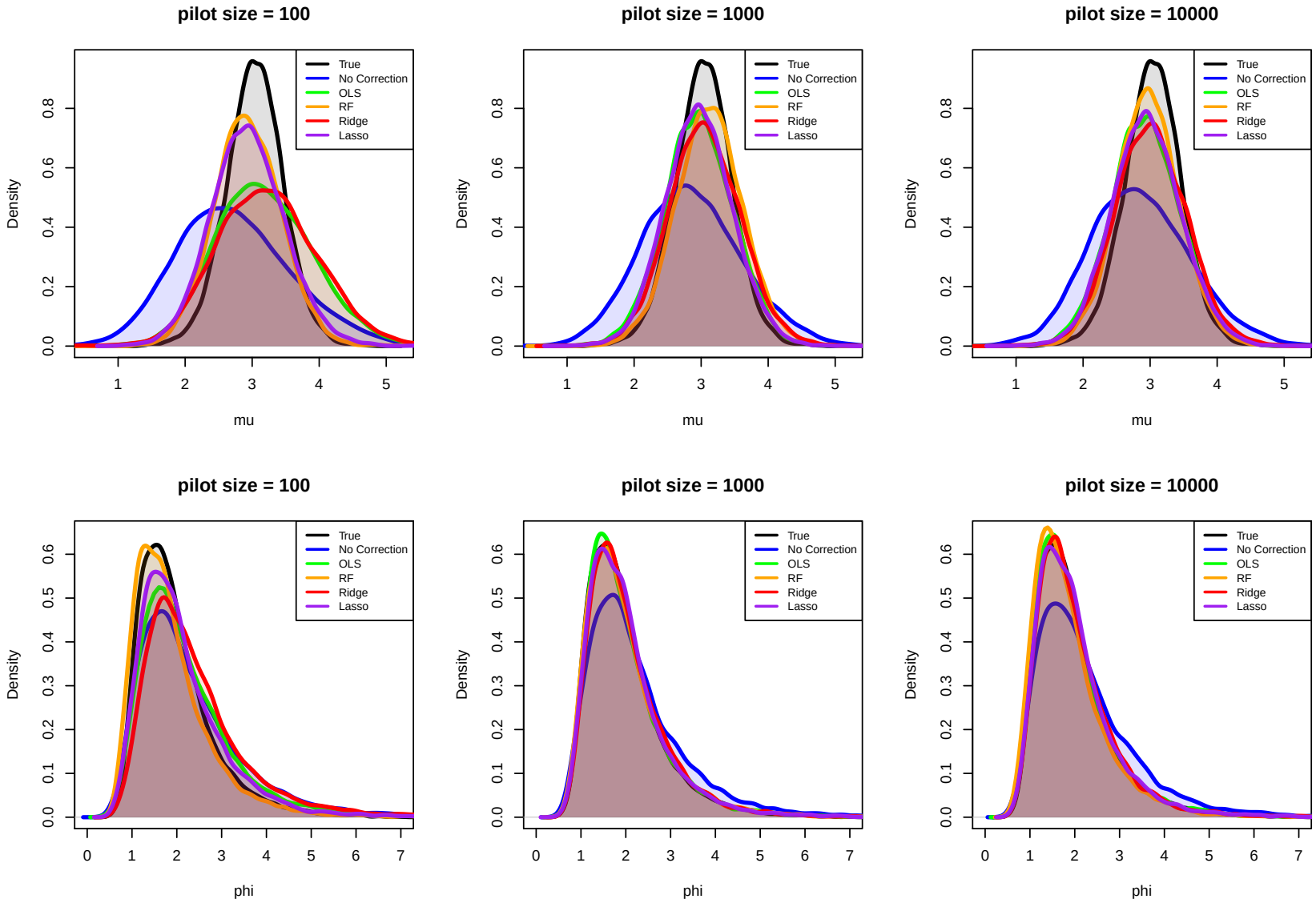
$$\boldsymbol{y} \overset{iid}{\sim} N(\mu, \sigma^2), \quad \mu|\sigma^2 \sim N(\mu_0, \frac{\sigma^2}{p_0}), \quad \frac{1}{\sigma^2} \sim \Gamma(\nu_0/2, s_0/2)$$

- Note: this is a conjugate prior, and we can sample directly from the posterior for comparison.
- $f(\boldsymbol{z})$ contains the true sufficient statistics $[\bar{\boldsymbol{z}}, \text{SS}(\boldsymbol{z})]$ and the median.
  - Also included: transformations of these values, and 15 $N(0, 1)$ covariates.

Questions to consider:

- Approximation to the true posterior?
- Computational costs?
- How many pilot samples are needed?

Results: Normal Toy Example



g and k distribution

$$\boldsymbol{y} \overset{iid}{\sim} Q_{gk}(\cdot, A, B, g, k), \quad (A, B, g, k) \sim (0, 10)^4$$

- $\boldsymbol{y}_{obs}$ consists of 10,000 observations from $Q_{gk}(\cdot, 3, 1, 2, 0.5)$
- $f(\boldsymbol{z})$ is a vector of 60 equally spaced order statistics and their powers (up to the 4th power)
- ABC - MCMC was used to facilitate sampling

Some notes about implementation:

- each pilot run consisted of 10,000 samples
- semi-automatic ABC-MCMC was implemented for OLS, lasso, ridge, and random forest, 25 times each
- MSE (with respect to the posterior mean) was used to compare methods

Results: g and k distribution

Posterior Mean MSE:

	A	B	g	k
OLS Reg.	**0.000103**	**0.000797**	**0.062392**	0.135695
Lasso	0.001441	0.004164	0.248754	0.558078
Ridge Reg.	0.001451	0.004888	1.156954	0.871882
Random Forests*	0.001405	0.021403	11.666854	**0.008647**

Some observations:

- The penalty term for both lasso and ridge regression was chosen (using CV) to be very small.
- Random forests takes longer to predict than the regression models - significantly increased computational burden of ABC-MCMC.
- ABC-MCMC required a lot of tuning to run well. Perhaps a different ABC algorithm would be preferred.

Conclusions and Future Work

Conclusions:

- Semi-automatic ABC provides a general approach for finding summary statistics for ABC.
- Lasso, ridge regression, and random forests are competitive with OLS regression for choosing η, and should be considered as alternative methods, especially when simulation is costly.

Future Work:

- I hope to compare the success of these approaches on a class of repulsive point processes, presented by Shirota and Gelfand (2017).
- Additional questions to investigate:

  1) How well does semi-automatic ABC perform in ABC-SMC?
  2) Would other nonlinear models (e.g., neural nets) outperform regression and random forests?