Scalable Bayesian Inference

David Dunson

Departments of Statistical Science & Mathematics, Duke University

December 3, 2018

Duke
UNIVERSITY

Outline

Typical approaches to big data

- There is an increasingly immense literature focused on big data

Typical approaches to big data

- There is an increasingly immense literature focused on big data
- Most of the focus has been on optimization methods

Typical approaches to big data

- There is an increasingly immense literature focused on big data
- Most of the focus has been on optimization methods
- Rapidly obtaining a point estimate even when sample size n & overall 'size' of data is immense

Typical approaches to big data

- There is an increasingly immense literature focused on big data
- Most of the focus has been on optimization methods
- Rapidly obtaining a point estimate even when sample size n & overall 'size' of data is immense
- Huge focus on specific settings - e.g., linear regression, labeling images, etc

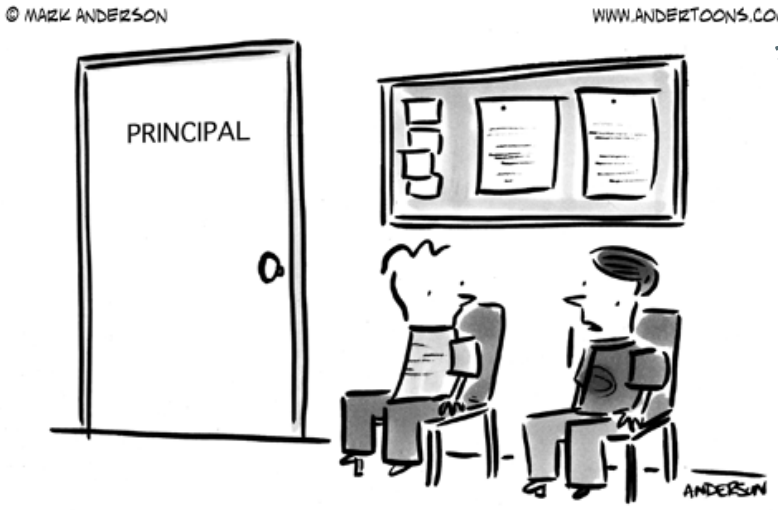
Typical approaches to big data

- There is an increasingly immense literature focused on big data
- Most of the focus has been on optimization methods
- Rapidly obtaining a point estimate even when sample size n & overall 'size' of data is immense
- Huge focus on specific settings - e.g., linear regression, labeling images, etc
- Bandwagons: most people work on very similar problems, while critical open problems remain untouched

My focus - probability models

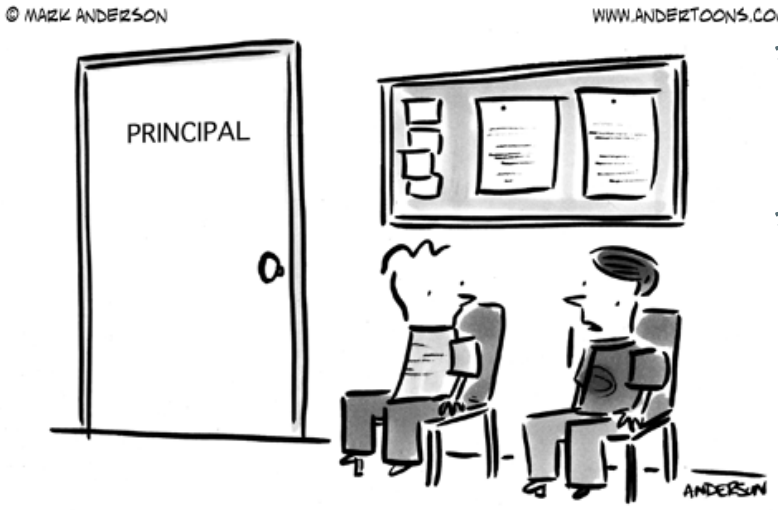
"I wish we hadn't learned probability
'cause I don't think our odds are good."

My focus - probability models



- General probabilistic inference algorithms for complex data

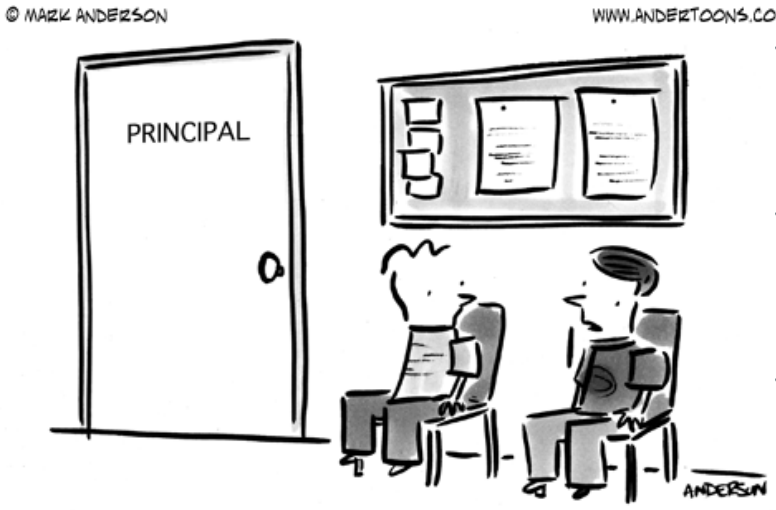
My focus - probability models



"I wish we hadn't learned probability
'cause I don't think our odds are good."

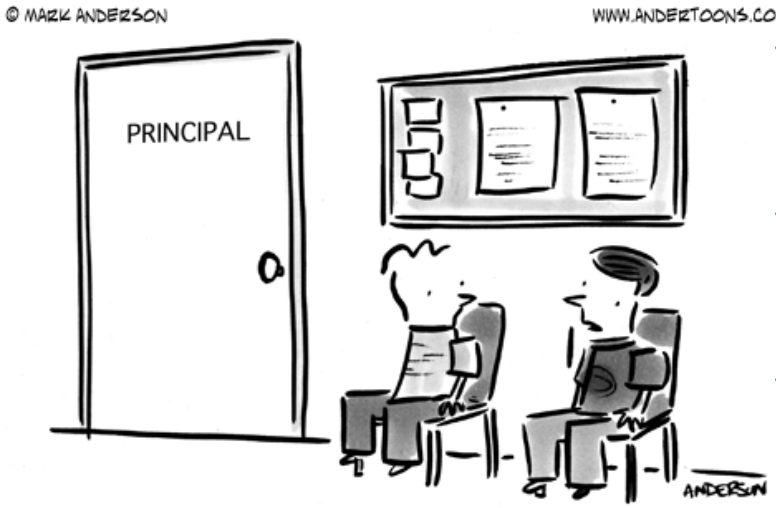
- General probabilistic inference algorithms for complex data
- We would like to handle arbitrarily complex probability models

My focus - probability models



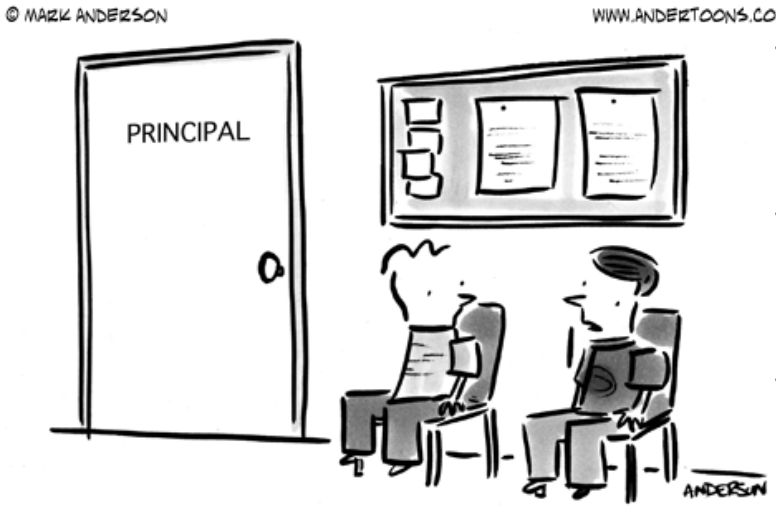
- General probabilistic inference algorithms for complex data
- We would like to handle arbitrarily complex probability models
- Algorithms scalable to huge data - potentially using many computers

My focus - probability models



- General probabilistic inference algorithms for complex data
- We would like to handle arbitrarily complex probability models
- Algorithms scalable to huge data - potentially using many computers

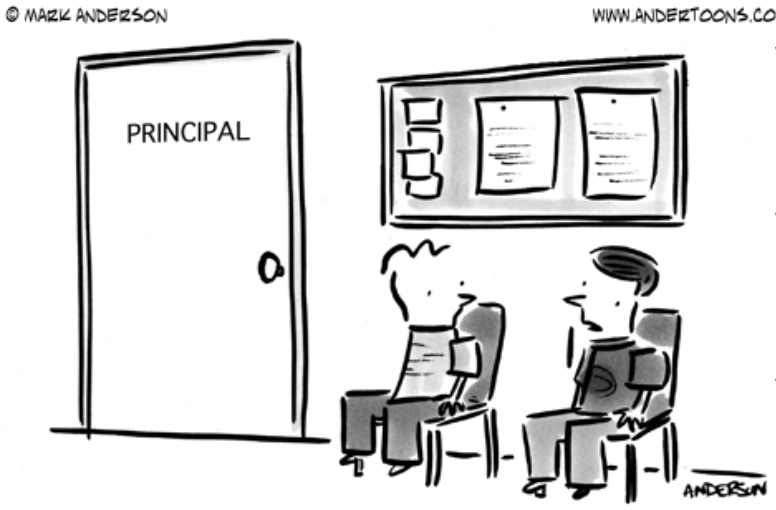
My focus - probability models



- General probabilistic inference algorithms for complex data
- We would like to handle arbitrarily complex probability models
- Algorithms scalable to huge data - potentially using many computers

- Accurate uncertainty quantification (UQ) is a critical issue

My focus - probability models



"I wish we hadn't learned probability
'cause I don't think our odds are good."

- General probabilistic inference algorithms for complex data
- We would like to handle arbitrarily complex probability models
- Algorithms scalable to huge data - potentially using many computers

- Accurate uncertainty quantification (UQ) is a critical issue
- Robustness of inferences also crucial

Bayes approaches

- Bayesian methods offer an attractive general approach for modeling complex data

Bayes approaches

- Bayesian methods offer an attractive general approach for modeling complex data
- Choosing a prior $\pi(\theta)$ & likelihood $L(Y^{(n)}|\theta)$, the posterior is

$$\pi_n(\theta|Y^{(n)}) = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{\int \pi(\theta)L(Y^{(n)}|\theta)d\theta} = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{L(Y^{(n)})}.$$

Bayes approaches

- Bayesian methods offer an attractive general approach for modeling complex data
- Choosing a prior $\pi(\theta)$ & likelihood $L(Y^{(n)}|\theta)$, the posterior is

$$\pi_n(\theta|Y^{(n)}) = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{\int \pi(\theta)L(Y^{(n)}|\theta)d\theta} = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{L(Y^{(n)})}.$$

- The posterior $\pi_n(\theta|Y^{(n)})$ characterizes uncertainty in the parameters, in any functional $f(\theta)$ of interest & in predictive distributions

Bayes approaches

- Bayesian methods offer an attractive general approach for modeling complex data
- Choosing a prior $\pi(\theta)$ & likelihood $L(Y^{(n)}|\theta)$, the posterior is

$$\pi_n(\theta|Y^{(n)}) = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{\int \pi(\theta)L(Y^{(n)}|\theta)d\theta} = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{L(Y^{(n)})}.$$

- The posterior $\pi_n(\theta|Y^{(n)})$ characterizes uncertainty in the parameters, in any functional $f(\theta)$ of interest & in predictive distributions
- Often θ is moderate to high-dimensional & the integral in denominator is intractable

Bayes approaches

- Bayesian methods offer an attractive general approach for modeling complex data
- Choosing a prior $\pi(\theta)$ & likelihood $L(Y^{(n)}|\theta)$, the posterior is

$$\pi_n(\theta|Y^{(n)}) = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{\int \pi(\theta)L(Y^{(n)}|\theta)d\theta} = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{L(Y^{(n)})}.$$

- The posterior $\pi_n(\theta|Y^{(n)})$ characterizes uncertainty in the parameters, in any functional $f(\theta)$ of interest & in predictive distributions
- Often θ is moderate to high-dimensional & the integral in denominator is intractable
- Hence, in interesting models the posterior is not available analytically - what to do??

Classical Posterior approximations

- In conjugate models, can express the posterior in simple form - e.g, as a multivariate Gaussian

Classical Posterior approximations

- In conjugate models, can express the posterior in simple form - e.g, as a multivariate Gaussian
- In more complex settings, can approximate posterior using some tractable class of distributions

Classical Posterior approximations

- In conjugate models, can express the posterior in simple form - e.g, as a multivariate Gaussian
- In more complex settings, can approximate posterior using some tractable class of distributions
- Large sample Gaussian approximations:

$$\pi_n(\theta|Y^{(n)}) \approx N(\hat{\mu}_n, \Sigma_n)$$

Bayesian central limit theorem (Bernstein von Mises)

Classical Posterior approximations

- In conjugate models, can express the posterior in simple form - e.g, as a multivariate Gaussian
- In more complex settings, can approximate posterior using some tractable class of distributions
- Large sample Gaussian approximations:

$$\pi_n(\theta|Y^{(n)}) \approx N(\hat{\mu}_n, \Sigma_n)$$

  Bayesian central limit theorem (Bernstein von Mises)
- Relies on sample size n large relative to # parameters p, likelihood smooth & differentiable, true value θ_0 in interior of parameter space

Classical Posterior approximations

- In conjugate models, can express the posterior in simple form - e.g, as a multivariate Gaussian
- In more complex settings, can approximate posterior using some tractable class of distributions
- Large sample Gaussian approximations:

$$\pi_n(\theta|Y^{(n)}) \approx N(\hat{\mu}_n, \Sigma_n)$$

  Bayesian central limit theorem (Bernstein von Mises)
- Relies on sample size n large relative to # parameters p, likelihood smooth & differentiable, true value θ_0 in interior of parameter space
- Related class of approximations use a Laplace approximation to $\int \pi(\theta) L(Y^{(n)}|\theta) d\theta$

Alternative analytic approximations

- As an alternative to Laplace/Gaussian approximations, we can define some approximating class $q(\theta)$

Alternative analytic approximations

- As an alternative to Laplace/Gaussian approximations, we can define some approximating class $q(\theta)$
- $q(\theta)$ may be something like a product of exponential family distributions parameterized by ξ

Alternative analytic approximations

- As an alternative to Laplace/Gaussian approximations, we can define some approximating class $q(\theta)$
- $q(\theta)$ may be something like a product of exponential family distributions parameterized by ξ
- We could think to define some discrepancy between $q(\theta)$ and $\pi_n(\theta) = \pi_n(\theta|Y^{(n)})$

Alternative analytic approximations

- As an alternative to Laplace/Gaussian approximations, we can define some approximating class $q(\theta)$
- $q(\theta)$ may be something like a product of exponential family distributions parameterized by ξ
- We could think to define some discrepancy between $q(\theta)$ and $\pi_n(\theta) = \pi_n(\theta|Y^{(n)})$
- If we can *optimize* ξ to minimize discrepancy, resulting $\hat{q}(\theta)$ may give us a decent approximation

Alternative analytic approximations

- As an alternative to Laplace/Gaussian approximations, we can define some approximating class $q(\theta)$
- $q(\theta)$ may be something like a product of exponential family distributions parameterized by ξ
- We could think to define some discrepancy between $q(\theta)$ and $\pi_n(\theta) = \pi_n(\theta|Y^{(n)})$
- If we can *optimize* ξ to minimize discrepancy, resulting $\hat{q}(\theta)$ may give us a decent approximation
- Basis of variational Bayes, expectation-propagation & related methods

Variational Bayes - brief comments

- ICML 2018 tutorial by Tamara Broderick
  <www.tamarabroderick.com>

Variational Bayes - brief comments

- ICML 2018 tutorial by Tamara Broderick <*www.tamarabroderick.com*>
- Based on maximizing a lower bound discarding an intractable term in KL divergence

Variational Bayes - brief comments

- ICML 2018 tutorial by Tamara Broderick <*www.tamarabroderick.com*>
- Based on maximizing a lower bound discarding an intractable term in KL divergence
- In general have no clue how accurate the approximation is

Variational Bayes - brief comments

- ICML 2018 tutorial by Tamara Broderick <*www.tamarabroderick.com*>
- Based on maximizing a lower bound discarding an intractable term in KL divergence
- In general have no clue how accurate the approximation is
- Often posterior uncertainty badly under-estimated, though there are some fix-ups; e.g., Giordano, Broderick & Jordan (2015)

Variational Bayes - brief comments

- ICML 2018 tutorial by Tamara Broderick <*www.tamarabroderick.com*>
- Based on maximizing a lower bound discarding an intractable term in KL divergence
- In general have no clue how accurate the approximation is
- Often posterior uncertainty badly under-estimated, though there are some fix-ups; e.g., Giordano, Broderick & Jordan (2015)
- Fix-ups improve the variance characterization in a local mode

Variational Bayes - brief comments

- ICML 2018 tutorial by Tamara Broderick <*www.tamarabroderick.com*>
- Based on maximizing a lower bound discarding an intractable term in KL divergence
- In general have no clue how accurate the approximation is
- Often posterior uncertainty badly under-estimated, though there are some fix-ups; e.g., Giordano, Broderick & Jordan (2015)
- Fix-ups improve the variance characterization in a local mode
- Recent article: “On statistical optimality of variational Bayes” Pati, Bhattacharya & Yang, arXiv:1712.08983.

Variational Bayes - brief comments

- ICML 2018 tutorial by Tamara Broderick <*www.tamarabroderick.com*>
- Based on maximizing a lower bound discarding an intractable term in KL divergence
- In general have no clue how accurate the approximation is
- Often posterior uncertainty badly under-estimated, though there are some fix-ups; e.g., Giordano, Broderick & Jordan (2015)
- Fix-ups improve the variance characterization in a local mode
- Recent article: “On statistical optimality of variational Bayes” Pati, Bhattacharya & Yang, arXiv:1712.08983.
- No theory on accuracy of UQ

Markov chain Monte Carlo

- Hence, accurate analytic approximations to the posterior have proven elusive outside of narrow settings

Markov chain Monte Carlo

- Hence, accurate analytic approximations to the posterior have proven elusive outside of narrow settings
- Markov chain Monte Carlo (MCMC) & other posterior sampling algorithms provide an alternative

Markov chain Monte Carlo

- Hence, accurate analytic approximations to the posterior have proven elusive outside of narrow settings
- Markov chain Monte Carlo (MCMC) & other posterior sampling algorithms provide an alternative
- MCMC: sequential algorithm to obtain correlated draws from the posterior:

$$\pi_n(\theta|Y^{(n)}) = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{\int \pi(\theta)L(Y^{(n)}|\theta)d\theta} = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{L(Y^{(n)})}.$$

Markov chain Monte Carlo

- Hence, accurate analytic approximations to the posterior have proven elusive outside of narrow settings
- Markov chain Monte Carlo (MCMC) & other posterior sampling algorithms provide an alternative
- MCMC: sequential algorithm to obtain correlated draws from the posterior:

$$\pi_n(\theta|Y^{(n)}) = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{\int \pi(\theta)L(Y^{(n)}|\theta)d\theta} = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{L(Y^{(n)})}.$$

- MCMC bypasses need to approximate the marginal likelihood $L(Y^{(n)})$

Markov chain Monte Carlo

- Hence, accurate analytic approximations to the posterior have proven elusive outside of narrow settings
- Markov chain Monte Carlo (MCMC) & other posterior sampling algorithms provide an alternative
- MCMC: sequential algorithm to obtain correlated draws from the posterior:

$$\pi_n(\theta|Y^{(n)}) = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{\int \pi(\theta)L(Y^{(n)}|\theta)d\theta} = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{L(Y^{(n)})}.$$

- MCMC bypasses need to approximate the marginal likelihood $L(Y^{(n)})$
- Often samples more useful then an analytic form for $\pi_n(\theta)$ anyway

Markov chain Monte Carlo

- Hence, accurate analytic approximations to the posterior have proven elusive outside of narrow settings
- Markov chain Monte Carlo (MCMC) & other posterior sampling algorithms provide an alternative
- MCMC: sequential algorithm to obtain correlated draws from the posterior:

$$\pi_n(\theta|Y^{(n)}) = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{\int \pi(\theta)L(Y^{(n)}|\theta)d\theta} = \frac{\pi(\theta)L(Y^{(n)}|\theta)}{L(Y^{(n)})}.$$

- MCMC bypasses need to approximate the marginal likelihood $L(Y^{(n)})$
- Often samples more useful then an analytic form for $\pi_n(\theta)$ anyway
- Can use samples to calculate a wide variety of posterior & predictive summaries of interest

MCMC

- MCMC-based summaries of the posterior for any functional $f(\theta)$

MCMC

- MCMC-based summaries of the posterior for any functional $f(\theta)$
- As the number of samples T increases, these summaries become more accurate

MCMC

- MCMC-based summaries of the posterior for any functional $f(\theta)$
- As the number of samples T increases, these summaries become more accurate
- MCMC constructs Markov chain with stationary distribution $\pi_n(\theta|Y^{(n)})$

MCMC

- MCMC-based summaries of the posterior for any functional $f(\theta)$
- As the number of samples T increases, these summaries become more accurate
- MCMC constructs Markov chain with stationary distribution $\pi_n(\theta|Y^{(n)})$
- A *transition kernel* is carefully chosen & iterative sampling proceeds

MCMC

- MCMC-based summaries of the posterior for any functional $f(\theta)$
- As the number of samples T increases, these summaries become more accurate
- MCMC constructs Markov chain with stationary distribution $\pi_n(\theta|Y^{(n)})$
- A *transition kernel* is carefully chosen & iterative sampling proceeds
- Most MCMC algorithms types of Metropolis-Hastings (MH):

MCMC

- MCMC-based summaries of the posterior for any functional $f(\theta)$
- As the number of samples T increases, these summaries become more accurate
- MCMC constructs Markov chain with stationary distribution $\pi_n(\theta|Y^{(n)})$
- A *transition kernel* is carefully chosen & iterative sampling proceeds
- Most MCMC algorithms types of Metropolis-Hastings (MH):
  1. $\theta^* \sim g(\theta^{(t-1)})$ = sample a proposal ($\theta^{(t)}$=sample at step t)

MCMC

- MCMC-based summaries of the posterior for any functional $f(\theta)$
- As the number of samples T increases, these summaries become more accurate
- MCMC constructs Markov chain with stationary distribution $\pi_n(\theta|Y^{(n)})$
- A *transition kernel* is carefully chosen & iterative sampling proceeds
- Most MCMC algorithms types of Metropolis-Hastings (MH):
  1. $\theta^* \sim g(\theta^{(t-1)})$ = sample a proposal ($\theta^{(t)}$=sample at step t)
  2. Accept proposal by letting $\theta^{(t)} = \theta^*$ with probability

$$\min\left\{1, \frac{\pi(\theta^*)L(Y^{(n)}|\theta^*)}{\pi(\theta^{(t-1)})L(Y^{(n)}|\theta^{(t-1)})}\frac{g(\theta^{(t-1)})}{g(\theta^*)}\right\}$$

Comments on MCMC & MH in particular

- Design of "efficient" MH algorithms involves choosing good proposals $g(\cdot)$

Comments on MCMC & MH in particular

- Design of "efficient" MH algorithms involves choosing good proposals $g(\cdot)$
- $g(\cdot)$ can depend on the previous value of θ & on the data but not on further back samples - except in adaptive MH

Comments on MCMC & MH in particular

- Design of "efficient" MH algorithms involves choosing good proposals $g(\cdot)$
- $g(\cdot)$ can depend on the previous value of θ & on the data but not on further back samples - except in adaptive MH
- Gibbs sampler: Letting $\theta = (\theta_1, \ldots, \theta_p)'$ we draw subsets of θ from their exact conditional posterior distributions fixing the others

Comments on MCMC & MH in particular

- Design of "efficient" MH algorithms involves choosing good proposals $g(\cdot)$
- $g(\cdot)$ can depend on the previous value of θ & on the data but not on further back samples - except in adaptive MH
- Gibbs sampler: Letting $\theta = (\theta_1, \ldots, \theta_p)'$ we draw subsets of θ from their exact conditional posterior distributions fixing the others
- Random walk: $g(\theta^{(t-1)})$ is a distribution centered on $\theta^{(t-1)}$ with a tunable covariance

Comments on MCMC & MH in particular

- Design of "efficient" MH algorithms involves choosing good proposals $g(\cdot)$
- $g(\cdot)$ can depend on the previous value of θ & on the data but not on further back samples - except in adaptive MH
- Gibbs sampler: Letting $\theta = (\theta_1, \ldots, \theta_p)'$ we draw subsets of θ from their exact conditional posterior distributions fixing the others
- Random walk: $g(\theta^{(t-1)})$ is a distribution centered on $\theta^{(t-1)}$ with a tunable covariance
- HMC/Langevin: Exploit gradient information to generate samples far from $\theta^{(t-1)}$ having high posterior density

MCMC & Computational bottlenecks

- Time per iteration increases with # of parameters/unknowns

MCMC & Computational bottlenecks

- Time per iteration increases with # of parameters/unknowns
- Can also increase with the sample size n

MCMC & Computational bottlenecks

- Time per iteration increases with # of parameters/unknowns
- Can also increase with the sample size n
- Due to the cost of sampling proposal & calculating acceptance probability

MCMC & Computational bottlenecks

- Time per iteration increases with # of parameters/unknowns
- Can also increase with the sample size n
- Due to the cost of sampling proposal & calculating acceptance probability
- Similar costs occur in most optimization algorithms!

MCMC & Computational bottlenecks

- Time per iteration increases with # of parameters/unknowns
- Can also increase with the sample size n
- Due to the cost of sampling proposal & calculating acceptance probability
- Similar costs occur in most optimization algorithms!
- For example, the computational bottleneck may be attributable to gradient evaluations

MCMC - A potential 2nd bottleneck

- MCMC does not produce independent samples from $\pi_n(\theta)$

MCMC - A potential 2nd bottleneck

- MCMC does not produce independent samples from $\pi_n(\theta)$
- Draws are auto-correlated - as level of correlation increases, information provided by each sample decreases

MCMC - A potential 2nd bottleneck

- MCMC does not produce independent samples from $\pi_n(\theta)$
- Draws are auto-correlated - as level of correlation increases, information provided by each sample decreases
- "Slowly mixing" Markov chains have highly autocorrelated draws

MCMC - A potential 2nd bottleneck

- MCMC does not produce independent samples from $\pi_n(\theta)$
- Draws are auto-correlated - as level of correlation increases, information provided by each sample decreases
- “Slowly mixing” Markov chains have highly autocorrelated draws
- A well designed MCMC algorithm with a good proposal should ideally exhibit rapid convergence & mixing

MCMC - A potential 2nd bottleneck

- MCMC does not produce independent samples from $\pi_n(\theta)$
- Draws are auto-correlated - as level of correlation increases, information provided by each sample decreases
- “Slowly mixing” Markov chains have highly autocorrelated draws
- A well designed MCMC algorithm with a good proposal should ideally exhibit rapid convergence & mixing
- Otherwise the Monte Carlo (MC) error in posterior summaries may be high

MCMC: Causes of scalability problems

- Often mixing gets worse as problem size grows (e.g. data dimension)

MCMC: Causes of scalability problems

- Often mixing gets worse as problem size grows (e.g. data dimension)
- Hence, in some cases we have a double bottleneck - worsening mixing & time/iteration

MCMC: Causes of scalability problems

- Often mixing gets worse as problem size grows (e.g. data dimension)
- Hence, in some cases we have a double bottleneck - worsening mixing & time/iteration
- Also MCMC is an inherently serial algorithm, so naive implementation may require storing & processing all data on one machine

MCMC: Causes of scalability problems

- Often mixing gets worse as problem size grows (e.g. data dimension)
- Hence, in some cases we have a double bottleneck - worsening mixing & time/iteration
- Also MCMC is an inherently serial algorithm, so naive implementation may require storing & processing all data on one machine
- Limits ease at which divide-and-conquer strategies can be applied.

MCMC: Causes of scalability problems

- Often mixing gets worse as problem size grows (e.g. data dimension)
- Hence, in some cases we have a double bottleneck - worsening mixing & time/iteration
- Also MCMC is an inherently serial algorithm, so naive implementation may require storing & processing all data on one machine
- Limits ease at which divide-and-conquer strategies can be applied.
- For the above reasons, it is common to simply state that MCMC is not scalable

MCMC: A bright future

- Each of the above problems can be addressed & there is an emerging rich literature!

MCMC: A bright future

- Each of the above problems can be addressed & there is an emerging rich literature!
- This is even given that orders of magnitude more researchers work on developing scalable optimization algorithms

MCMC: A bright future

- Each of the above problems can be addressed & there is an emerging rich literature!
- This is even given that orders of magnitude more researchers work on developing scalable optimization algorithms
- For an MCMC algorithm to be scalable, MC error in posterior summaries based on running for time τ should not explode with dimensionality

MCMC: A bright future

- Each of the above problems can be addressed & there is an emerging rich literature!
- This is even given that orders of magnitude more researchers work on developing scalable optimization algorithms
- For an MCMC algorithm to be scalable, MC error in posterior summaries based on running for time τ should not explode with dimensionality
- Some popular algorithms have been shown to not be scalable while others can be made scalable

MCMC: A bright future

- Each of the above problems can be addressed & there is an emerging rich literature!
- This is even given that orders of magnitude more researchers work on developing scalable optimization algorithms
- For an MCMC algorithm to be scalable, MC error in posterior summaries based on running for time τ should not explode with dimensionality
- Some popular algorithms have been shown to not be scalable while others can be made scalable
- I'm going to highlight some relevant relevant work starting by focusing on big n problems & then transitioning to big p

Outline

Some Solutions

- **Embarrassingly parallel (EP) MCMC**: run MCMC in parallel for different subsets of data & combine.

Some Solutions

- **Embarrassingly parallel (EP) MCMC**: run MCMC in parallel for different subsets of data & combine.
- **Approximate MCMC**: Approximate expensive to evaluate transition kernels.

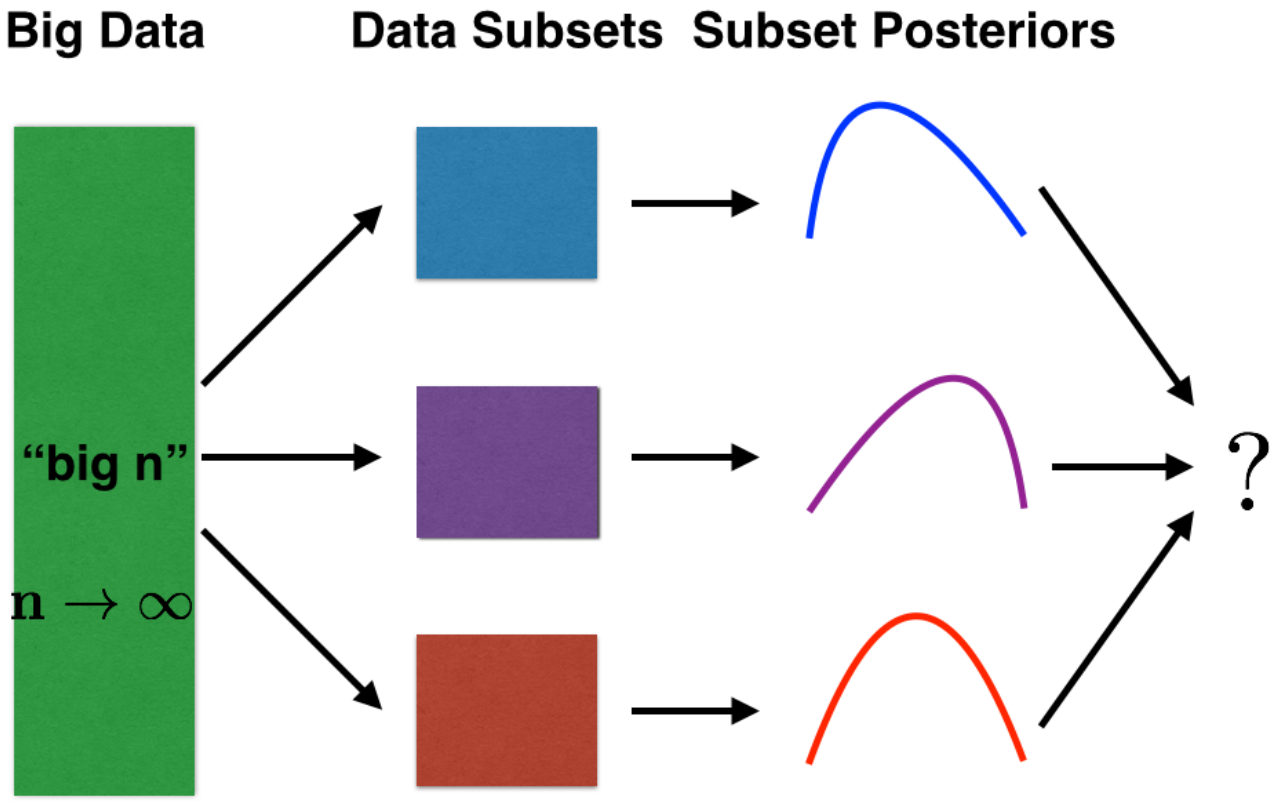
Some Solutions

- **Embarrassingly parallel (EP) MCMC**: run MCMC in parallel for different subsets of data & combine.
- **Approximate MCMC**: Approximate expensive to evaluate transition kernels.
- **C-Bayes**: Condition on observed data being in small neighborhood of data drawn from assumed model [*ROBUST*]

Some Solutions

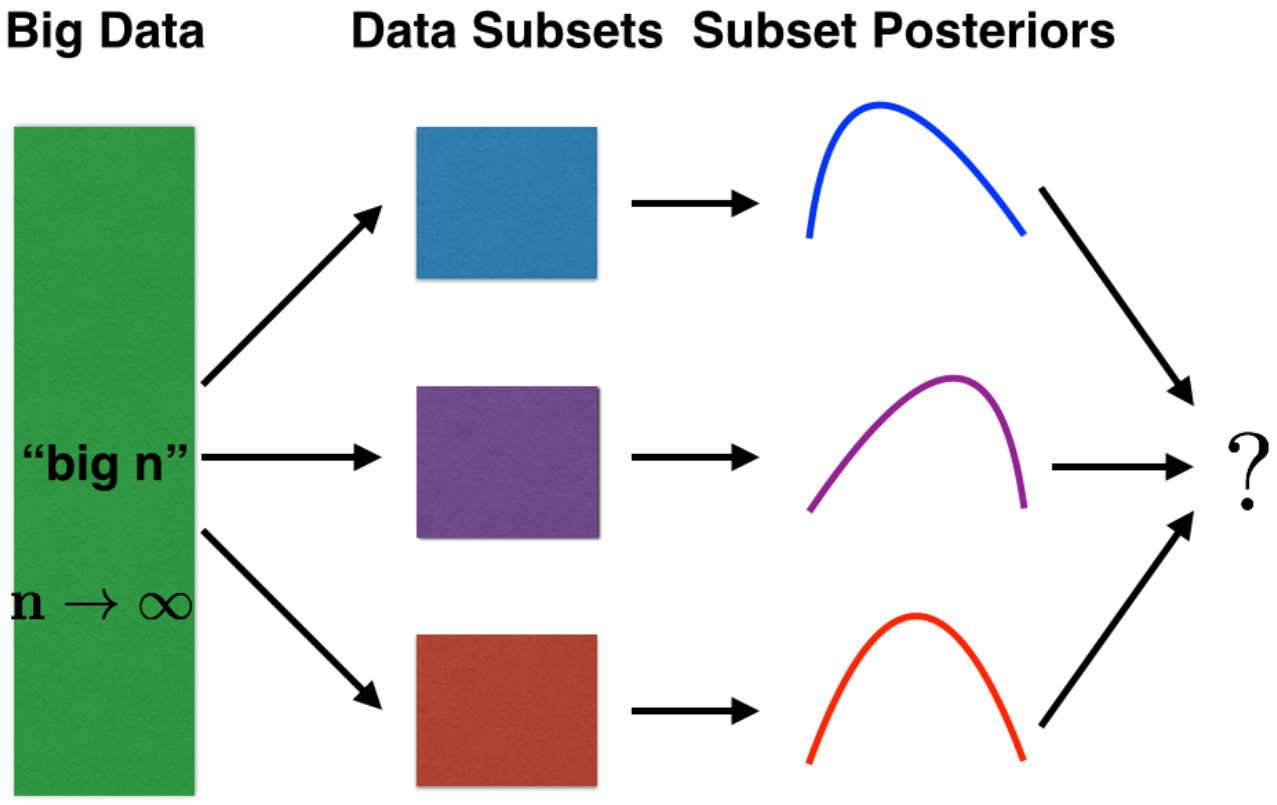
- **Embarrassingly parallel (EP) MCMC**: run MCMC in parallel for different subsets of data & combine.
- **Approximate MCMC**: Approximate expensive to evaluate transition kernels.
- **C-Bayes**: Condition on observed data being in small neighborhood of data drawn from assumed model [*ROBUST*]
- **Hybrid algorithms**: run MCMC for a subset of the parameters & use a fast estimate for the others.

Embarrassingly parallel MCMC



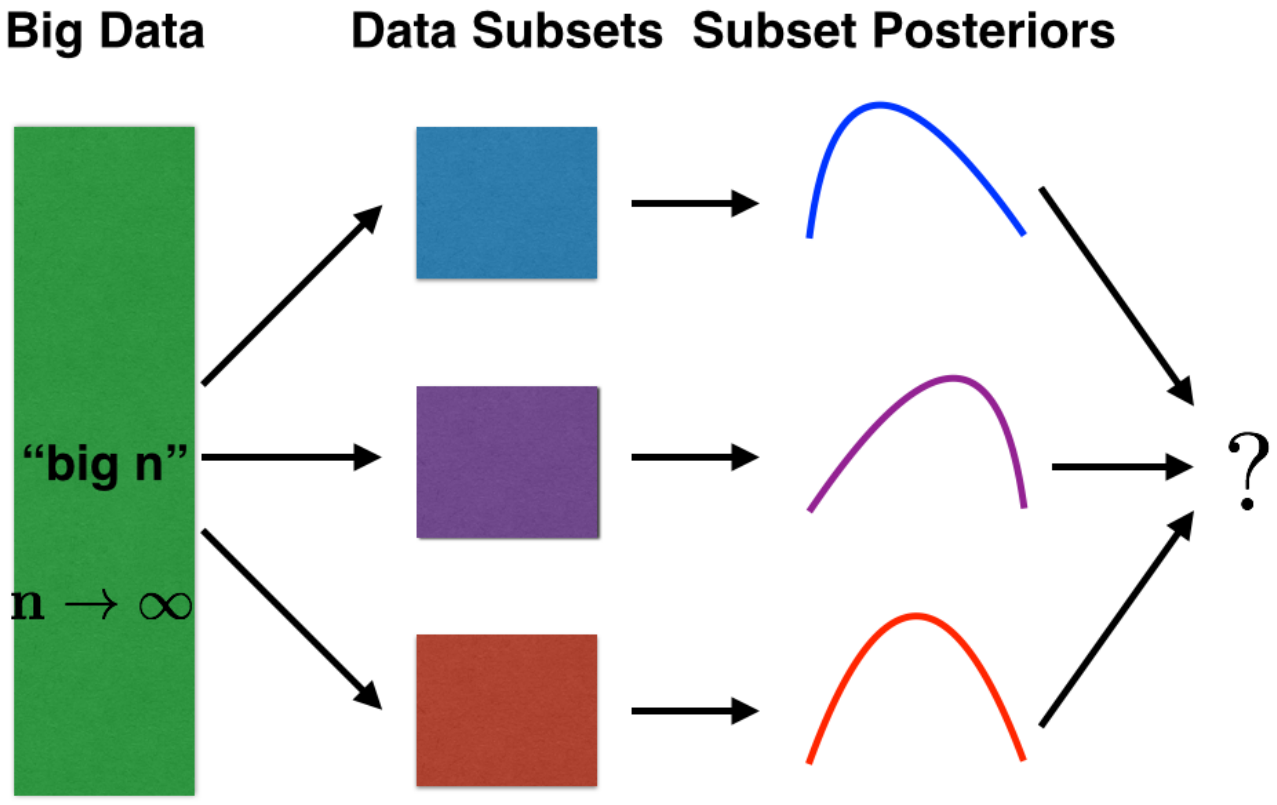
- Divide large sample size n data set into many smaller data sets stored on different machines

Embarrassingly parallel MCMC



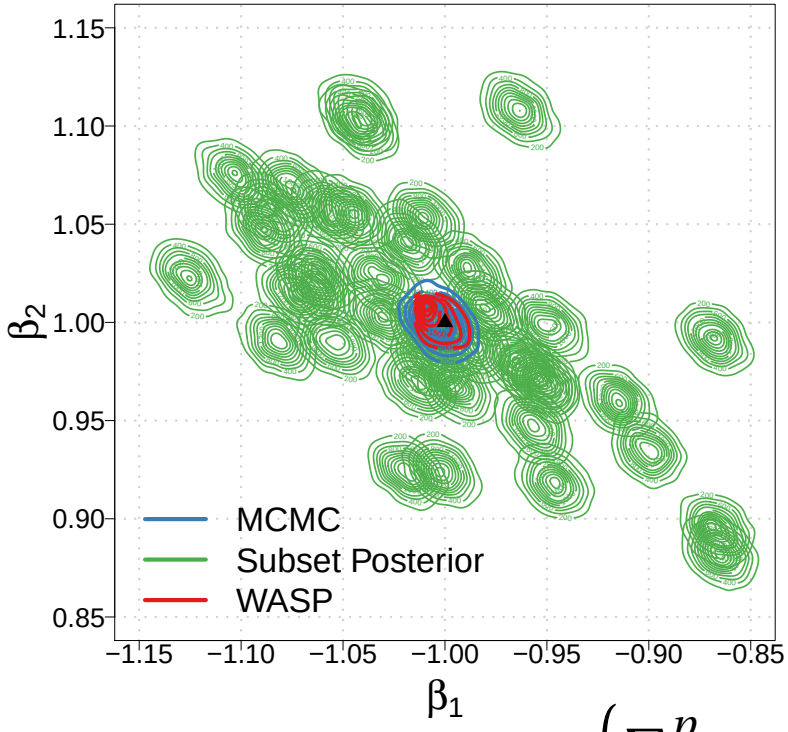
- Divide large sample size n data set into many smaller data sets stored on different machines
- Draw posterior samples for each subset posterior in parallel

Embarrassingly parallel MCMC



- Divide large sample size n data set into many smaller data sets stored on different machines
- Draw posterior samples for each subset posterior in parallel
- 'Magically' combine the results quickly & simply

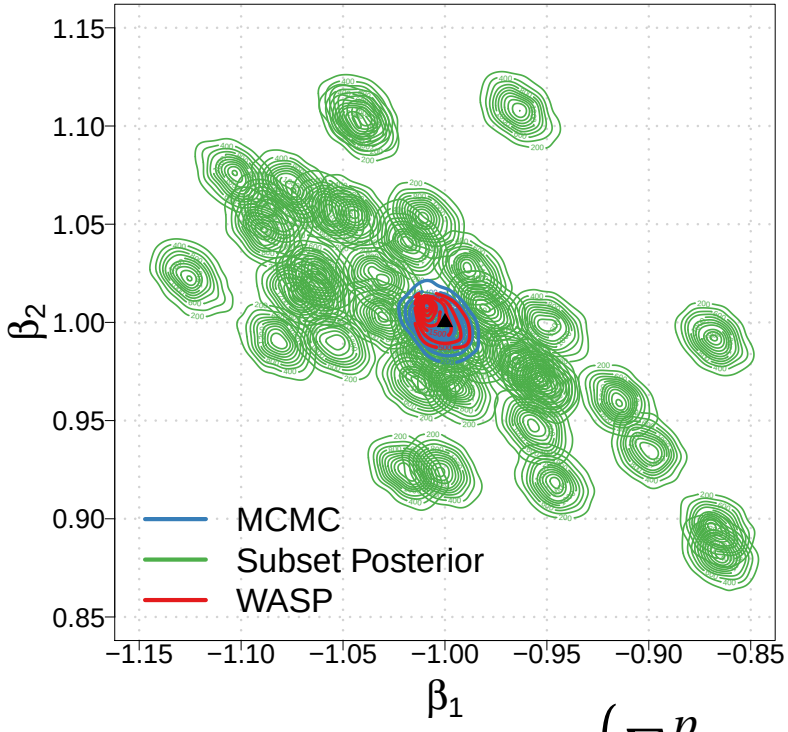
Toy Example: Logistic Regression



$$\text{pr}(y_i = 1 | x_{i1}, \ldots, x_{ip}, \theta) = \frac{\exp\left(\sum_{j=1}^{p} x_{ij}\beta_j\right)}{1 + \exp\left(\sum_{j=1}^{p} x_{ij}\beta_j\right)}.$$

- Subset posteriors: ‘noisy’ approximations of full data posterior.

Toy Example: Logistic Regression



$$\text{pr}(y_i = 1 | x_{i1}, \ldots, x_{ip}, \theta) = \frac{\exp\left(\sum_{j=1}^{p} x_{ij}\beta_j\right)}{1 + \exp\left(\sum_{j=1}^{p} x_{ij}\beta_j\right)}.$$

- Subset posteriors: 'noisy' approximations of full data posterior.
- 'Averaging' of subset posteriors reduces this 'noise' & leads to an accurate posterior approximation.

Stochastic Approximation

- Full data posterior density of *inid* data $Y^{(n)}$

$$\pi_n(\theta \mid Y^{(n)}) = \frac{\prod_{i=1}^n p_i(y_i \mid \theta)\pi(\theta)}{\int_\Theta \prod_{i=1}^n p_i(y_i \mid \theta)\pi(\theta)\, d\theta}.$$

Stochastic Approximation

- Full data posterior density of *inid* data $Y^{(n)}$

$$\pi_n(\theta \mid Y^{(n)}) = \frac{\prod_{i=1}^n p_i(y_i \mid \theta)\pi(\theta)}{\int_\Theta \prod_{i=1}^n p_i(y_i \mid \theta)\pi(\theta)d\theta}.$$

- Divide full data $Y^{(n)}$ into k subsets of size m:
  $Y^{(n)} = (Y_{[1]}, \ldots, Y_{[j]}, \ldots, Y_{[k]})$.

Stochastic Approximation

- Full data posterior density of *inid* data $Y^{(n)}$

$$\pi_n(\theta \mid Y^{(n)}) = \frac{\prod_{i=1}^n p_i(y_i \mid \theta)\pi(\theta)}{\int_\Theta \prod_{i=1}^n p_i(y_i \mid \theta)\pi(\theta)d\theta}.$$

- Divide full data $Y^{(n)}$ into k subsets of size m:
  $Y^{(n)} = (Y_{[1]}, \ldots, Y_{[j]}, \ldots, Y_{[k]})$.
- Subset posterior density for jth data subset

$$\pi_m^\gamma(\theta \mid Y_{[j]}) = \frac{\prod_{i\in[j]}(p_i(y_i \mid \theta))^\gamma \pi(\theta)}{\int_\Theta \prod_{i\in[j]}(p_i(y_i \mid \theta))^\gamma \pi(\theta)d\theta}.$$

Stochastic Approximation

- Full data posterior density of *inid* data $Y^{(n)}$

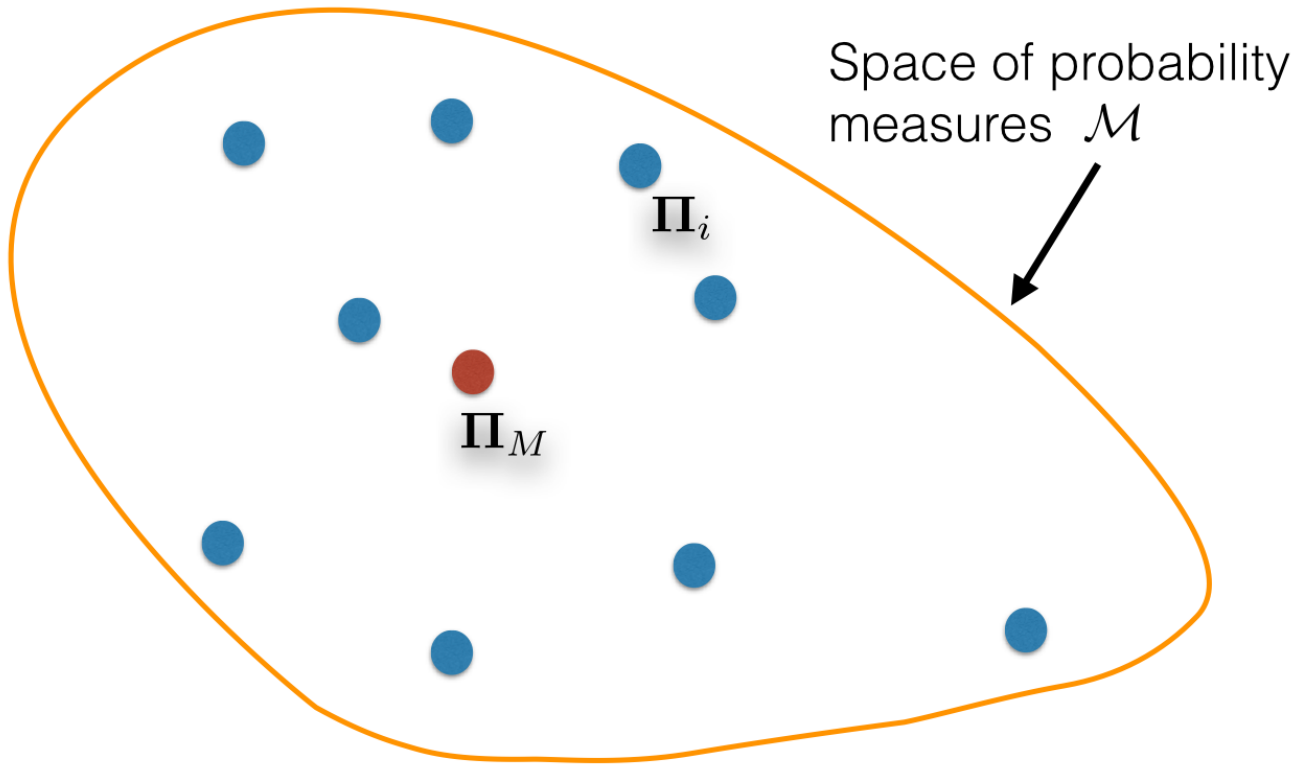
$$\pi_n(\theta \mid Y^{(n)}) = \frac{\prod_{i=1}^n p_i(y_i \mid \theta)\pi(\theta)}{\int_\Theta \prod_{i=1}^n p_i(y_i \mid \theta)\pi(\theta)d\theta}.$$

- Divide full data $Y^{(n)}$ into k subsets of size m:
  $Y^{(n)} = (Y_{[1]}, \ldots, Y_{[j]}, \ldots, Y_{[k]})$.
- Subset posterior density for jth data subset

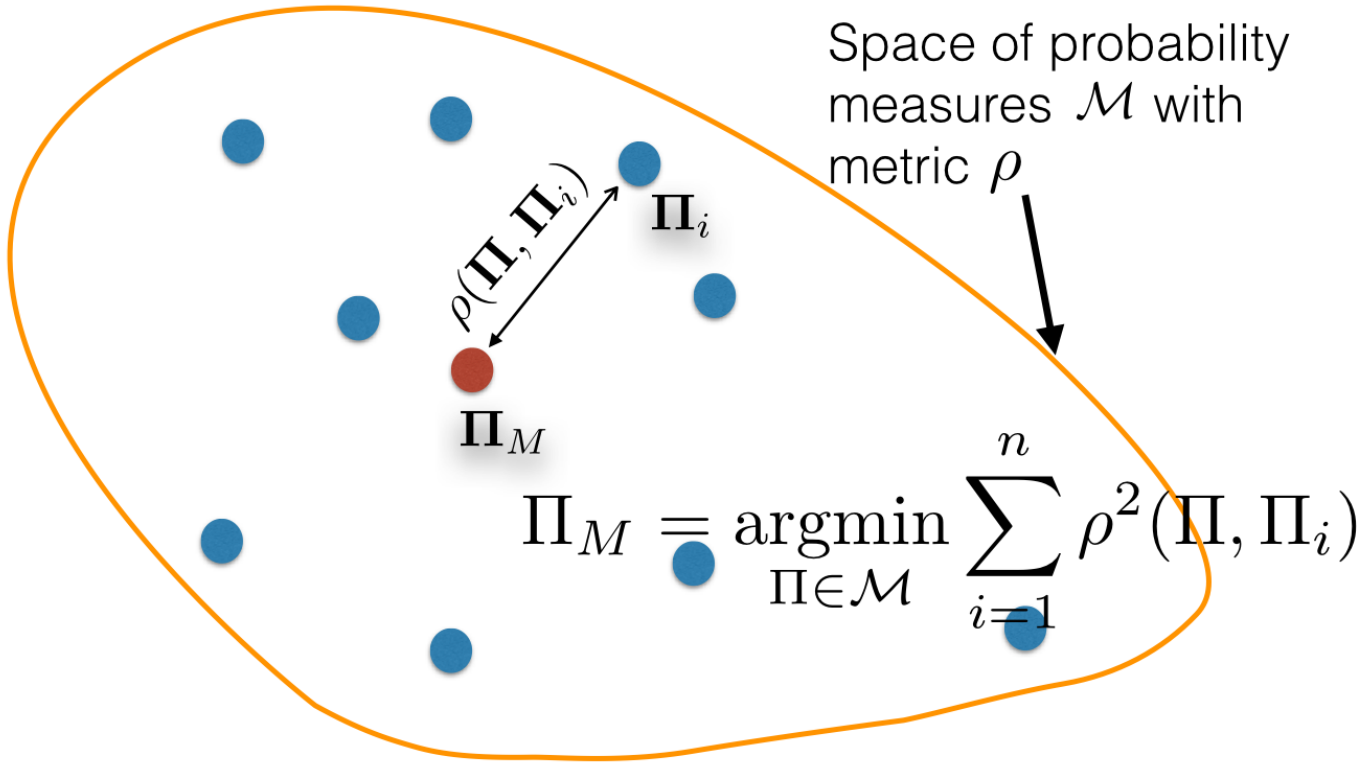
$$\pi_m^\gamma(\theta \mid Y_{[j]}) = \frac{\prod_{i\in[j]}(p_i(y_i \mid \theta))^\gamma \pi(\theta)}{\int_\Theta \prod_{i\in[j]}(p_i(y_i \mid \theta))^\gamma \pi(\theta)d\theta}.$$

- $\gamma = O(k)$ - chosen to minimize approximation error

Barycenter in Metric Spaces



Barycenter in Metric Spaces



WAsserstein barycenter of Subset Posteriors (WASP)

Srivastava, Li & Dunson (2015)

- 2-Wasserstein distance between $\mu, \nu \in \mathscr{P}_2(\Theta)$

$$W_2(\mu,\nu) = \inf\left\{\left(\mathbb{E}[d^2(X,Y)]\right)^{\frac{1}{2}} : \mathsf{law}(X) = \mu, \mathsf{law}(Y) = \nu\right\}.$$

WAsserstein barycenter of Subset Posteriors (WASP)

Srivastava, Li & Dunson (2015)

- 2-Wasserstein distance between $\mu, \nu \in \mathscr{P}_2(\Theta)$

$$W_2(\mu,\nu) = \inf\left\{\left(\mathbb{E}[d^2(X,Y)]\right)^{\frac{1}{2}} : \text{law}(X) = \mu, \text{law}(Y) = \nu\right\}.$$

- $\Pi_m^\gamma(\cdot \mid Y_{[j]})$ for $j = 1, \ldots, k$ are combined through WASP

$$\overline{\Pi}_n^\gamma(\cdot \mid Y^{(n)}) = \underset{\Pi \in \mathscr{P}_2(\Theta)}{\text{argmin}} \frac{1}{k} \sum_{j=1}^{k} W_2^2(\Pi, \Pi_m^\gamma(\cdot \mid Y_{[j]})).$$ [Agueh & Carlier (2011)]

WAsserstein barycenter of Subset Posteriors (WASP)

Srivastava, Li & Dunson (2015)

- 2-Wasserstein distance between $\mu, \nu \in \mathscr{P}_2(\Theta)$

$$W_2(\mu,\nu) = \inf\left\{\left(\mathbb{E}[d^2(X,Y)]\right)^{\frac{1}{2}} : \text{law}(X) = \mu, \text{law}(Y) = \nu\right\}.$$

- $\Pi_m^{\gamma}(\cdot \mid Y_{[j]})$ for $j = 1, \ldots, k$ are combined through WASP

$$\overline{\Pi}_n^{\gamma}(\cdot \mid Y^{(n)}) = \underset{\Pi \in \mathscr{P}_2(\Theta)}{\operatorname{argmin}} \frac{1}{k} \sum_{j=1}^{k} W_2^2(\Pi, \Pi_m^{\gamma}(\cdot \mid Y_{[j]})).$$ [Agueh & Carlier (2011)]

- Plugging in $\widehat{\Pi}_m^{\gamma}(\cdot \mid Y_{[j]})$ for $j = 1, \ldots, k$, a linear program (LP) can be used for fast estimation of an atomic approximation!

LP Estimation of WASP

- Minimizing Wasserstein is solution to a discrete optimal transport problem

LP Estimation of WASP

- Minimizing Wasserstein is solution to a discrete optimal transport problem
- Let $\mu = \sum_{j=1}^{J_1} a_j \delta_{\theta_{1j}}$, $\nu = \sum_{l=1}^{J_2} b_l \delta_{\theta_{2l}}$ & $\mathbf{M}_{12} \in \Re^{J_1 \times J_2}$ = matrix of square differences in atoms $\{\theta_{1j}\}, \{\theta_{2l}\}$.

LP Estimation of WASP

- Minimizing Wasserstein is solution to a discrete optimal transport problem
- Let $\mu = \sum_{j=1}^{J_1} a_j \delta_{\theta_{1j}}$, $\nu = \sum_{l=1}^{J_2} b_l \delta_{\theta_{2l}}$ & $\mathbf{M}_{12} \in \Re^{J_1 \times J_2}$ = matrix of square differences in atoms $\{\theta_{1j}\}, \{\theta_{2l}\}$.
- Optimal transport polytope: $\mathscr{T}(\mathbf{a}, \mathbf{b})$ = set of doubly stochastic matrices w/ row sums $\mathbf{a}$ & column sums $\mathbf{b}$

LP Estimation of WASP

- Minimizing Wasserstein is solution to a discrete optimal transport problem
- Let $\mu = \sum_{j=1}^{J_1} a_j \delta_{\theta_{1j}}$, $\nu = \sum_{l=1}^{J_2} b_l \delta_{\theta_{2l}}$ & $\mathbf{M}_{12} \in \Re^{J_1 \times J_2}$ = matrix of square differences in atoms $\{\theta_{1j}\}, \{\theta_{2l}\}$.
- Optimal transport polytope: $\mathcal{T}(\mathbf{a}, \mathbf{b})$ = set of doubly stochastic matrices w/ row sums $\mathbf{a}$ & column sums $\mathbf{b}$
- Objective is to find $\mathbf{T} \in \mathcal{T}(\mathbf{a}, \mathbf{b})$ minimizing $\text{tr}(\mathbf{T}^T \mathbf{M}_{12})$

LP Estimation of WASP

- Minimizing Wasserstein is solution to a discrete optimal transport problem
- Let $\mu = \sum_{j=1}^{J_1} a_j \delta_{\theta_{1j}}$, $\nu = \sum_{l=1}^{J_2} b_l \delta_{\theta_{2l}}$ & $\mathbf{M}_{12} \in \Re^{J_1 \times J_2}$ = matrix of square differences in atoms $\{\theta_{1j}\}, \{\theta_{2l}\}$.
- Optimal transport polytope: $\mathscr{T}(\mathbf{a}, \mathbf{b})$ = set of doubly stochastic matrices w/ row sums $\mathbf{a}$ & column sums $\mathbf{b}$
- Objective is to find $\mathbf{T} \in \mathscr{T}(\mathbf{a}, \mathbf{b})$ minimizing $\text{tr}(\mathbf{T}^T \mathbf{M}_{12})$
- For WASP, generalize to multimargin optimal transport problem - entropy smoothing has been used previously

LP Estimation of WASP

- Minimizing Wasserstein is solution to a discrete optimal transport problem
- Let $\mu = \sum_{j=1}^{J_1} a_j \delta_{\theta_{1j}}$, $\nu = \sum_{l=1}^{J_2} b_l \delta_{\theta_{2l}}$ & $\mathbf{M}_{12} \in \Re^{J_1 \times J_2}$ = matrix of square differences in atoms $\{\theta_{1j}\}, \{\theta_{2l}\}$.
- Optimal transport polytope: $\mathcal{T}(\mathbf{a}, \mathbf{b})$ = set of doubly stochastic matrices w/ row sums $\mathbf{a}$ & column sums $\mathbf{b}$
- Objective is to find $\mathbf{T} \in \mathcal{T}(\mathbf{a}, \mathbf{b})$ minimizing $\text{tr}(\mathbf{T}^T \mathbf{M}_{12})$
- For WASP, generalize to multimargin optimal transport problem - entropy smoothing has been used previously
- We can avoid such smoothing & use sparse LP solvers - neglible computation cost compared to sampling

WASP: Theorems

Theorem (Subset Posteriors)

Under "usual" regularity conditions, there exists a constant C_1 independent of subset posteriors, such that for large m,

$$\mathbb{E}_{P_{\theta_0}^{[j]}} W_2^2 \left\{ \Pi_m^{\gamma}(\cdot \mid Y_{[j]}), \delta_{\theta_0}(\cdot) \right\} \leq C_1 \left(\frac{\log^2 m}{m} \right)^{\frac{1}{\alpha}} \qquad j = 1, \ldots, k,$$

WASP: Theorems

Theorem (Subset Posteriors)

Under "usual" regularity conditions, there exists a constant C_1 independent of subset posteriors, such that for large m,

$$\mathbb{E}_{P_{\theta_0}^{[j]}} W_2^2 \left\{ \Pi_m^{\gamma}(\cdot \mid Y_{[j]}), \delta_{\theta_0}(\cdot) \right\} \leq C_1 \left(\frac{\log^2 m}{m} \right)^{\frac{1}{\alpha}} \qquad j = 1, \ldots, k,$$

Theorem (WASP)

Under "usual" regularity conditions and for large m,

$$W_2 \left\{ \overline{\Pi}_n^{\gamma}(\cdot \mid Y^{(n)}), \delta_{\theta_0}(\cdot) \right\} = O_{P_{\theta_0}^{(n)}} \left(\sqrt{\frac{\log^{2/\alpha} m}{k m^{1/\alpha}}} \right).$$

Simple & Fast Posterior Interval Estimation (PIE)

Li, Srivastava & Dunson (2017)

- Usually report point & interval estimates for different 1-d functionals - *multidimensional posterior difficult to interpret*

Simple & Fast Posterior Interval Estimation (PIE)

Li, Srivastava & Dunson (2017)

- Usually report point & interval estimates for different 1-d functionals - *multidimensional posterior difficult to interpret*
- WASP has explicit relationship with subset posteriors in 1-d

Simple & Fast Posterior Interval Estimation (PIE)

Li, Srivastava & Dunson (2017)

- Usually report point & interval estimates for different 1-d functionals - *multidimensional posterior difficult to interpret*
- WASP has explicit relationship with subset posteriors in 1-d
- Quantiles of WASP are simple averages of quantiles of subset posteriors

Simple & Fast Posterior Interval Estimation (PIE)

Li, Srivastava & Dunson (2017)

- Usually report point & interval estimates for different 1-d functionals - *multidimensional posterior difficult to interpret*
- WASP has explicit relationship with subset posteriors in 1-d
- Quantiles of WASP are simple averages of quantiles of subset posteriors
- Leads to a super trivial algorithm - run MCMC for each subset & average quantiles - *reminiscent of bag of little bootstraps*

Simple & Fast Posterior Interval Estimation (PIE)

Li, Srivastava & Dunson (2017)

- Usually report point & interval estimates for different 1-d functionals - *multidimensional posterior difficult to interpret*
- WASP has explicit relationship with subset posteriors in 1-d
- Quantiles of WASP are simple averages of quantiles of subset posteriors
- Leads to a super trivial algorithm - run MCMC for each subset & average quantiles - *reminiscent of bag of little bootstraps*
- Strong theory showing accuracy of the resulting approximation

Simple & Fast Posterior Interval Estimation (PIE)

Li, Srivastava & Dunson (2017)

- Usually report point & interval estimates for different 1-d functionals - *multidimensional posterior difficult to interpret*
- WASP has explicit relationship with subset posteriors in 1-d
- Quantiles of WASP are simple averages of quantiles of subset posteriors
- Leads to a super trivial algorithm - run MCMC for each subset & average quantiles - *reminiscent of bag of little bootstraps*
- Strong theory showing accuracy of the resulting approximation
- Can implement in *STAN*, which allows powered likelihoods

Theory on PIE/1-d WASP

- We show 1-d WASP $\overline{\Pi}_n(\xi|Y^{(n)})$ is highly accurate approximation to exact posterior $\Pi_n(\xi|Y^{(n)})$

Theory on PIE/1-d WASP

- We show 1-d WASP $\overline{\Pi}_n(\xi|Y^{(n)})$ is highly accurate approximation to exact posterior $\Pi_n(\xi|Y^{(n)})$
- As subset sample size m increases, W_2 distance between them decreases at faster than parametric rate $o_p(n^{-1/2})$

Theory on PIE/1-d WASP

- We show 1-d WASP $\overline{\Pi}_n(\xi|Y^{(n)})$ is highly accurate approximation to exact posterior $\Pi_n(\xi|Y^{(n)})$
- As subset sample size m increases, W_2 distance between them decreases at faster than parametric rate $o_p(n^{-1/2})$
- Theorem allows $k = O(n^c)$ and $m = O(n^{1-c})$ for any $c \in (0,1)$, so m can increase very slowly relative to k (*recall* $n = mk$)

Theory on PIE/1-d WASP

- We show 1-d WASP $\overline{\Pi}_n(\xi|Y^{(n)})$ is highly accurate approximation to exact posterior $\Pi_n(\xi|Y^{(n)})$
- As subset sample size m increases, W_2 distance between them decreases at faster than parametric rate $o_p(n^{-1/2})$
- Theorem allows $k = O(n^c)$ and $m = O(n^{1-c})$ for any $c \in (0,1)$, so m can increase very slowly relative to k (*recall* $n = mk$)
- Their biases, variances, quantiles only differ in high orders of the total sample size

Theory on PIE/1-d WASP

- We show 1-d WASP $\overline{\Pi}_n(\xi|Y^{(n)})$ is highly accurate approximation to exact posterior $\Pi_n(\xi|Y^{(n)})$
- As subset sample size m increases, W_2 distance between them decreases at faster than parametric rate $o_p(n^{-1/2})$
- Theorem allows $k = O(n^c)$ and $m = O(n^{1-c})$ for any $c \in (0,1)$, so m can increase very slowly relative to k (*recall* $n = mk$)
- Their biases, variances, quantiles only differ in high orders of the total sample size
- Conditions: standard, mild conditions on likelihood + prior finite 2nd moment & uniform integrabiity of subset posteriors

Results

- We have implemented for rich variety of data & models

Results

- We have implemented for rich variety of data & models
- Logistic & linear random effects models, mixture models, matrix & tensor factorizations, Gaussian process regression

Results

- We have implemented for rich variety of data & models
- Logistic & linear random effects models, mixture models, matrix & tensor factorizations, Gaussian process regression
- Nonparametric models, dependence, hierarchical models, etc.

Results

- We have implemented for rich variety of data & models
- Logistic & linear random effects models, mixture models, matrix & tensor factorizations, Gaussian process regression
- Nonparametric models, dependence, hierarchical models, etc.
- We compare to long runs of MCMC (when feasible) & VB

Results

- We have implemented for rich variety of data & models
- Logistic & linear random effects models, mixture models, matrix & tensor factorizations, Gaussian process regression
- Nonparametric models, dependence, hierarchical models, etc.
- We compare to long runs of MCMC (when feasible) & VB
- WASP/PIE is much faster than MCMC & highly accurate

Results

- We have implemented for rich variety of data & models
- Logistic & linear random effects models, mixture models, matrix & tensor factorizations, Gaussian process regression
- Nonparametric models, dependence, hierarchical models, etc.
- We compare to long runs of MCMC (when feasible) & VB
- WASP/PIE is much faster than MCMC & highly accurate
- Carefully designed VB implementations often do very well

aMCMC *Johndrow, Mattingly, Mukherjee & Dunson (2015)*

- Different way to speed up MCMC - replace expensive transition kernels with approximations

aMCMC *Johndrow, Mattingly, Mukherjee & Dunson (2015)*

- Different way to speed up MCMC - replace expensive transition kernels with approximations
- For example, approximate a conditional distribution in Gibbs sampler with a Gaussian or using a subsample of data

aMCMC *Johndrow, Mattingly, Mukherjee & Dunson (2015)*

- Different way to speed up MCMC - replace expensive transition kernels with approximations
- For example, approximate a conditional distribution in Gibbs sampler with a Gaussian or using a subsample of data
- Can potentially vastly speed up MCMC sampling in high-dimensional settings

aMCMC *Johndrow, Mattingly, Mukherjee & Dunson (2015)*

- Different way to speed up MCMC - replace expensive transition kernels with approximations
- For example, approximate a conditional distribution in Gibbs sampler with a Gaussian or using a subsample of data
- Can potentially vastly speed up MCMC sampling in high-dimensional settings
- Original MCMC sampler converges to a stationary distribution corresponding to the exact posterior

aMCMC *Johndrow, Mattingly, Mukherjee & Dunson (2015)*

- Different way to speed up MCMC - replace expensive transition kernels with approximations
- For example, approximate a conditional distribution in Gibbs sampler with a Gaussian or using a subsample of data
- Can potentially vastly speed up MCMC sampling in high-dimensional settings
- Original MCMC sampler converges to a stationary distribution corresponding to the exact posterior
- Not clear what happens when we start substituting in approximations - may diverge etc

aMCMC Overview

- aMCMC is used routinely - there is an increasing rich literature on algorithms

aMCMC Overview

- aMCMC is used routinely - there is an increasing rich literature on algorithms
- Theory: guarantees can be used to target design of algorithms

aMCMC Overview

- aMCMC is used routinely - there is an increasing rich literature on algorithms
- Theory: guarantees can be used to target design of algorithms
- Define ‘exact’ MCMC algorithm, which is computationally intractable but has good mixing

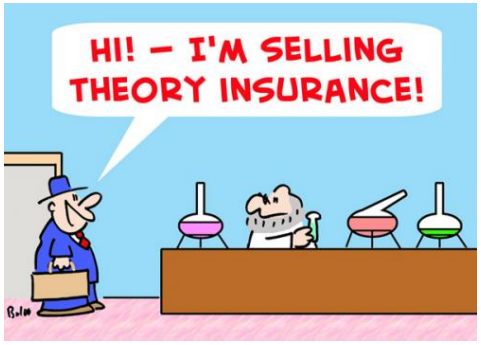
aMCMC Overview

- aMCMC is used routinely - there is an increasing rich literature on algorithms
- Theory: guarantees can be used to target design of algorithms
- Define ‘exact’ MCMC algorithm, which is computationally intractable but has good mixing
- ‘exact’ chain converges to stationary distribution corresponding to exact posterior

aMCMC Overview

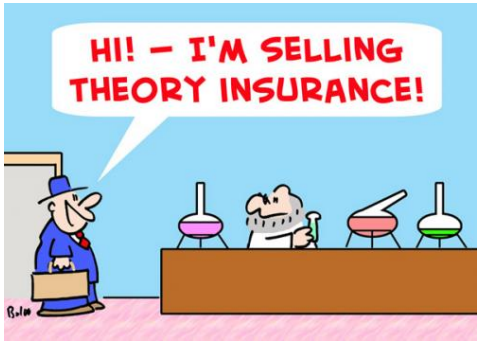
- aMCMC is used routinely - there is an increasing rich literature on algorithms
- Theory: guarantees can be used to target design of algorithms
- Define 'exact' MCMC algorithm, which is computationally intractable but has good mixing
- 'exact' chain converges to stationary distribution corresponding to exact posterior
- Approximate kernel in exact chain with more computationally tractable alternative

Sketch of theory



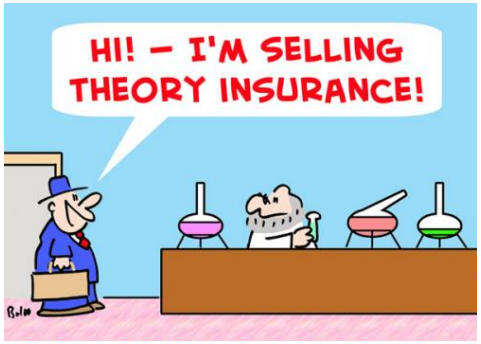
- Define $s_\epsilon = \tau_1(\mathscr{P})/\tau_1(\mathscr{P}_\epsilon)$ = *computational speed-up*, $\tau_1(\mathscr{P})$ = time for one step with transition kernel $\mathscr{P}$

Sketch of theory



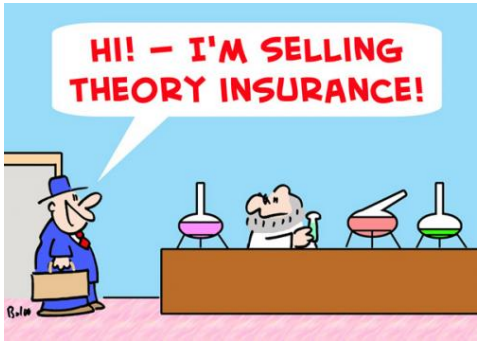
- Define $s_\epsilon = \tau_1(\mathscr{P})/\tau_1(\mathscr{P}_\epsilon)$ = *computational speed-up*, $\tau_1(\mathscr{P})$ = time for one step with transition kernel $\mathscr{P}$
- Interest: optimizing computational time-accuracy tradeoff for estimators of $\Pi f = \int_\Theta f(\theta)\Pi(d\theta|x)$

Sketch of theory



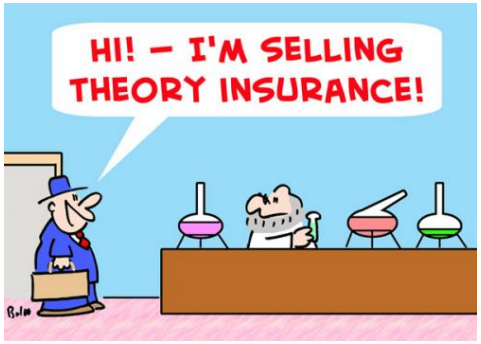
- Define $s_\epsilon = \tau_1(\mathscr{P})/\tau_1(\mathscr{P}_\epsilon)$ = *computational speed-up*, $\tau_1(\mathscr{P})$ = time for one step with transition kernel $\mathscr{P}$
- Interest: optimizing computational time-accuracy tradeoff for estimators of $\Pi f = \int_\Theta f(\theta)\Pi(d\theta|x)$
- We provide *tight, finite sample* bounds on L_2 error

Sketch of theory



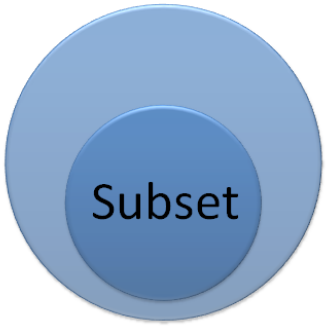
- Define $s_\epsilon = \tau_1(\mathscr{P})/\tau_1(\mathscr{P}_\epsilon)$ = *computational speed-up*, $\tau_1(\mathscr{P})$ = time for one step with transition kernel $\mathscr{P}$
- Interest: optimizing computational time-accuracy tradeoff for estimators of $\Pi f = \int_\Theta f(\theta)\Pi(d\theta|x)$
- We provide *tight, finite sample* bounds on L_2 error
- aMCMC estimators win for low computational budgets but have asymptotic bias

Sketch of theory



- Define $s_\epsilon = \tau_1(\mathscr{P})/\tau_1(\mathscr{P}_\epsilon)$ = *computational speed-up*, $\tau_1(\mathscr{P})$ = time for one step with transition kernel $\mathscr{P}$
- Interest: optimizing computational time-accuracy tradeoff for estimators of $\Pi f = \int_\Theta f(\theta)\Pi(d\theta|x)$
- We provide *tight, finite sample* bounds on L_2 error
- aMCMC estimators win for low computational budgets but have asymptotic bias
- Often larger approximation error → larger s_ϵ & rougher approximations are better when speed super important

Ex 1: Approximations using subsets

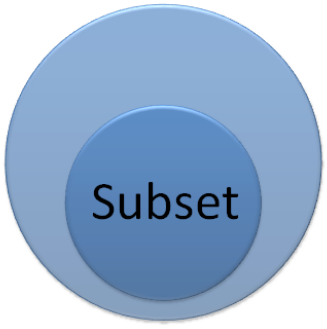


- Replace the full data likelihood with

$$L_\epsilon(x \mid \theta) = \left(\prod_{i \in V} L(x_i \mid \theta) \right)^{N/|V|},$$

for randomly chosen subset $V \subset \{1, \ldots, n\}$.

Ex 1: Approximations using subsets



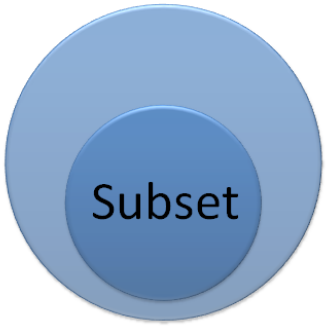
- Replace the full data likelihood with

$$L_\epsilon(x \mid \theta) = \left(\prod_{i \in V} L(x_i \mid \theta) \right)^{N/|V|},$$

  for randomly chosen subset $V \subset \{1, \ldots, n\}$.

- Applied to Pólya-Gamma data augmentation for logistic regression

Ex 1: Approximations using subsets

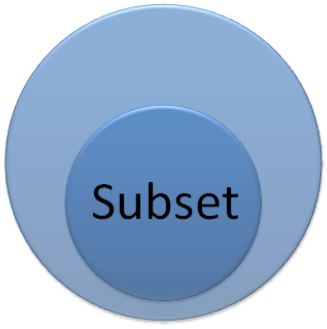


- Replace the full data likelihood with

$$L_\epsilon(x \mid \theta) = \left(\prod_{i \in V} L(x_i \mid \theta) \right)^{N/|V|},$$

  for randomly chosen subset $V \subset \{1, \ldots, n\}$.
- Applied to Pólya-Gamma data augmentation for logistic regression
- Different V at each iteration – trivial modification to Gibbs

Ex 1: Approximations using subsets

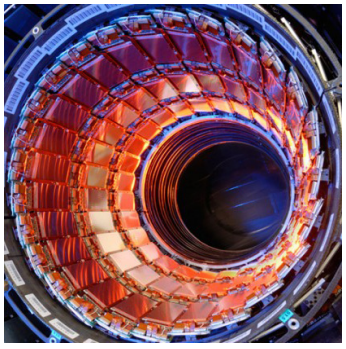


- Replace the full data likelihood with

$$L_\epsilon(x \mid \theta) = \left(\prod_{i \in V} L(x_i \mid \theta) \right)^{N/|V|},$$

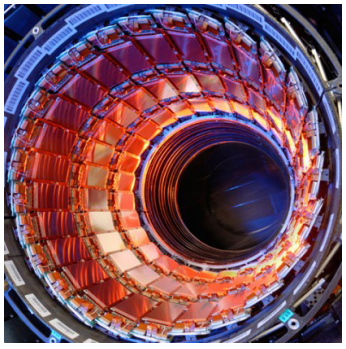
  for randomly chosen subset $V \subset \{1, \ldots, n\}$.
- Applied to Pólya-Gamma data augmentation for logistic regression
- Different V at each iteration – trivial modification to Gibbs
- Assumptions hold with high probability for subsets > minimal size (wrt distribution of subsets, data & kernel).

Application to SUSY dataset



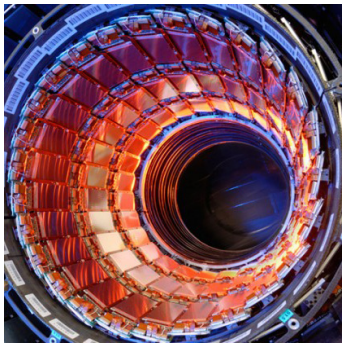
- $n = 5,000,000$ (0.5 million test), binary outcome & 18 continuous covariates

Application to SUSY dataset



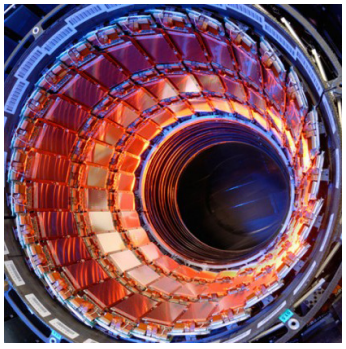
- $n = 5,000,000$ (0.5 million test), binary outcome & 18 continuous covariates
- Considered subsets sizes ranging from $|V| = 1,000$ to $4,500,000$

Application to SUSY dataset



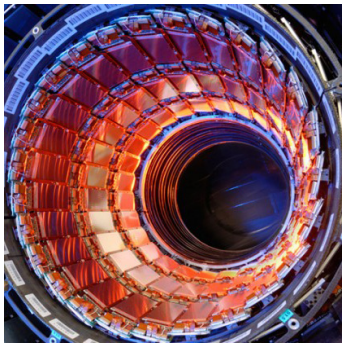
- $n = 5,000,000$ (0.5 million test), binary outcome & 18 continuous covariates
- Considered subsets sizes ranging from $|V| = 1,000$ to $4,500,000$
- Considered different losses as function of $|V|$

Application to SUSY dataset



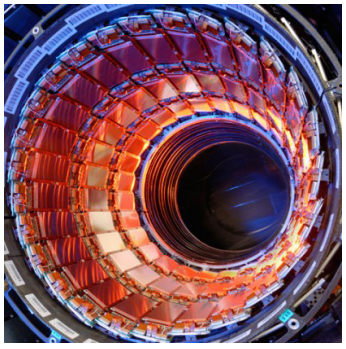
- $n = 5,000,000$ (0.5 million test), binary outcome & 18 continuous covariates
- Considered subsets sizes ranging from $|V| = 1,000$ to $4,500,000$
- Considered different losses as function of $|V|$
- Rate at which loss $\to 0$ with ϵ heavily dependent on loss

Application to SUSY dataset



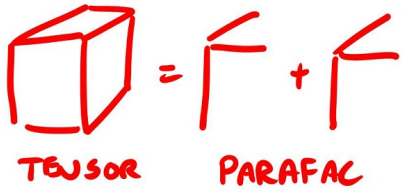
- $n = 5,000,000$ (0.5 million test), binary outcome & 18 continuous covariates
- Considered subsets sizes ranging from $|V| = 1,000$ to $4,500,000$
- Considered different losses as function of $|V|$
- Rate at which loss $\to 0$ with ϵ heavily dependent on loss
- For small computational budget & focus on posterior mean estimation, small subsets preferred

Application to SUSY dataset



- $n = 5,000,000$ (0.5 million test), binary outcome & 18 continuous covariates
- Considered subsets sizes ranging from $|V| = 1,000$ to $4,500,000$
- Considered different losses as function of $|V|$
- Rate at which loss $\to 0$ with ϵ heavily dependent on loss
- For small computational budget & focus on posterior mean estimation, small subsets preferred
- As budget increases & loss focused more on tails (e.g., for interval estimation), optimal $|V|$ increases

Application 2: Mixture models & tensor factorizations

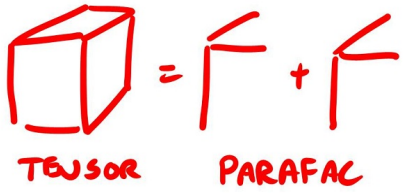


- We also considered a nonparametric Bayes model:

$$\text{pr}(y_{i1} = c_1, \ldots, y_{ip} = c_p) = \sum_{h=1}^{k} \lambda_h \prod_{j=1}^{p} \psi_{hc_j}^{(j)},$$

a very useful model for multivariate categorical data

Application 2: Mixture models & tensor factorizations

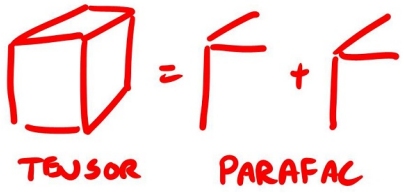


- We also considered a nonparametric Bayes model:

$$\text{pr}(y_{i1} = c_1, \ldots, y_{ip} = c_p) = \sum_{h=1}^{k} \lambda_h \prod_{j=1}^{p} \psi_{hc_j}^{(j)},$$

  a very useful model for multivariate categorical data
- Dunson & Xing (2009) - a data augmentation Gibbs sampler

Application 2: Mixture models & tensor factorizations

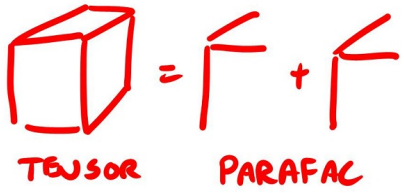


- We also considered a nonparametric Bayes model:

$$\text{pr}(y_{i1} = c_1, \ldots, y_{ip} = c_p) = \sum_{h=1}^{k} \lambda_h \prod_{j=1}^{p} \psi_{hc_j}^{(j)},$$

  a very useful model for multivariate categorical data
- Dunson & Xing (2009) - a data augmentation Gibbs sampler
- Sampling latent classes computationally prohibitive for huge n

Application 2: Mixture models & tensor factorizations

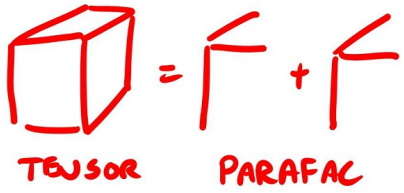


- We also considered a nonparametric Bayes model:

$$\text{pr}(y_{i1} = c_1, \ldots, y_{ip} = c_p) = \sum_{h=1}^{k} \lambda_h \prod_{j=1}^{p} \psi_{hc_j}^{(j)},$$

  a very useful model for multivariate categorical data
- Dunson & Xing (2009) - a data augmentation Gibbs sampler
- Sampling latent classes computationally prohibitive for huge n
- Use adaptive Gaussian approximation - avoid sampling individual latent classes

Application 2: Mixture models & tensor factorizations

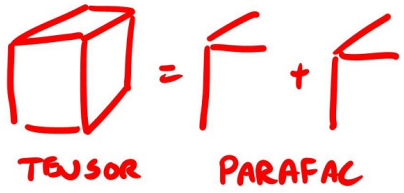


- We also considered a nonparametric Bayes model:

$$\text{pr}(y_{i1} = c_1, \ldots, y_{ip} = c_p) = \sum_{h=1}^{k} \lambda_h \prod_{j=1}^{p} \psi_{hc_j}^{(j)},$$

  a very useful model for multivariate categorical data
- Dunson & Xing (2009) - a data augmentation Gibbs sampler
- Sampling latent classes computationally prohibitive for huge n
- Use adaptive Gaussian approximation - avoid sampling individual latent classes
- We have shown Assumptions 1-2, Assumption 2 result more general than this setting

Application 2: Mixture models & tensor factorizations

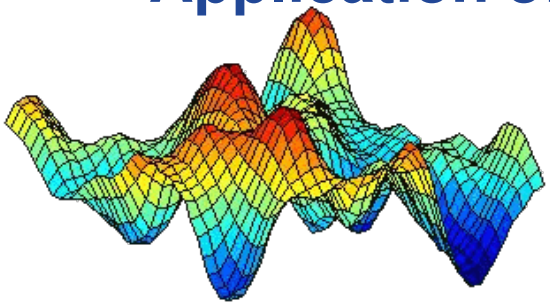


- We also considered a nonparametric Bayes model:

$$\text{pr}(y_{i1} = c_1, \ldots, y_{ip} = c_p) = \sum_{h=1}^{k} \lambda_h \prod_{j=1}^{p} \psi_{hc_j}^{(j)},$$

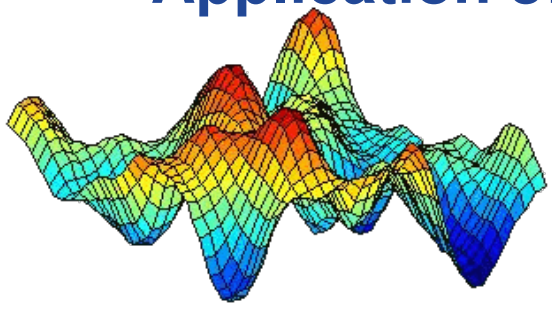
  a very useful model for multivariate categorical data
- Dunson & Xing (2009) - a data augmentation Gibbs sampler
- Sampling latent classes computationally prohibitive for huge n
- Use adaptive Gaussian approximation - avoid sampling individual latent classes
- We have shown Assumptions 1-2, Assumption 2 result more general than this setting
- Improved computation performance for large n

Application 3: Low rank approximation to GP



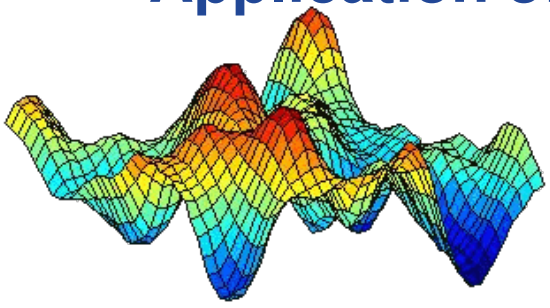
- Gaussian process regression, $y_i = f(x_i) + \eta_i$, $\eta_i \sim N(0, \sigma^2)$

Application 3: Low rank approximation to GP



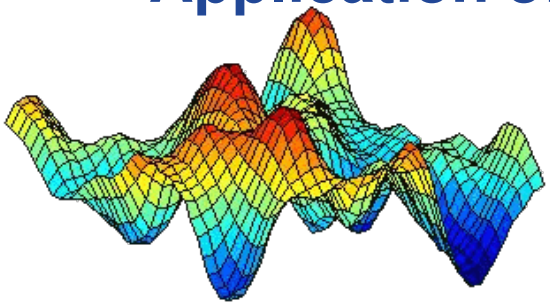
- Gaussian process regression, $y_i = f(x_i) + \eta_i$, $\eta_i \sim N(0, \sigma^2)$
- $f \sim GP$ prior with covariance $\tau^2 \exp(-\phi ||x_1 - x_2||^2)$

Application 3: Low rank approximation to GP



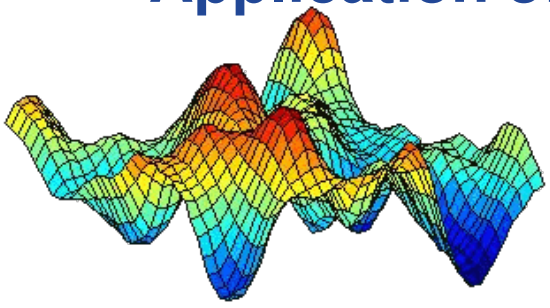
- Gaussian process regression, $y_i = f(x_i) + \eta_i$, $\eta_i \sim N(0, \sigma^2)$
- $f \sim GP$ prior with covariance $\tau^2 \exp(-\phi||x_1 - x_2||^2)$
- Discrete-uniform on ϕ & gamma priors on τ^{-2}, σ^{-2}

Application 3: Low rank approximation to GP



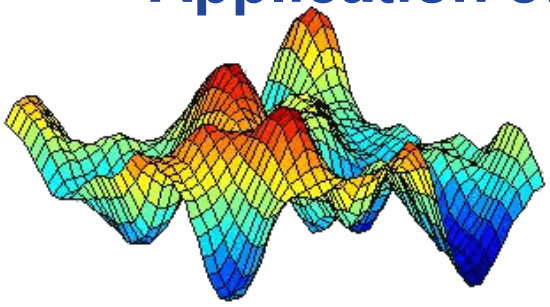
- Gaussian process regression, $y_i = f(x_i) + \eta_i$, $\eta_i \sim N(0, \sigma^2)$
- $f \sim GP$ prior with covariance $\tau^2 \exp(-\phi||x_1 - x_2||^2)$
- Discrete-uniform on ϕ & gamma priors on τ^{-2}, σ^{-2}
- Marginal MCMC sampler updates $\phi, \tau^{-2}, \sigma^{-2}$

Application 3: Low rank approximation to GP



- Gaussian process regression, $y_i = f(x_i) + \eta_i$, $\eta_i \sim N(0, \sigma^2)$
- $f \sim GP$ prior with covariance $\tau^2 \exp(-\phi||x_1 - x_2||^2)$
- Discrete-uniform on ϕ & gamma priors on τ^{-2}, σ^{-2}
- Marginal MCMC sampler updates $\phi, \tau^{-2}, \sigma^{-2}$
- We show Assumption 1 holds under mild regularity conditions on "truth", Assumption 2 holds for partial rank-r eigen approximation to Σ

Application 3: Low rank approximation to GP



- Gaussian process regression, $y_i = f(x_i) + \eta_i$, $\eta_i \sim N(0, \sigma^2)$
- $f \sim GP$ prior with covariance $\tau^2 \exp(-\phi||x_1 - x_2||^2)$
- Discrete-uniform on ϕ & gamma priors on τ^{-2}, σ^{-2}
- Marginal MCMC sampler updates $\phi, \tau^{-2}, \sigma^{-2}$
- We show Assumption 1 holds under mild regularity conditions on "truth", Assumption 2 holds for partial rank-r eigen approximation to Σ
- Less accurate approximations clearly superior in practice for small computational budget

Some interim comments

- EP-MCMC & aMCMC can be used in many-many settings to vastly speed up computation for big n

Some interim comments

- EP-MCMC & aMCMC can be used in many-many settings to vastly speed up computation for big n
- Here, I just illustrated some of the possible algorithms - there is an increasingly huge literature on many other approaches

Some interim comments

- EP-MCMC & aMCMC can be used in many-many settings to vastly speed up computation for big n
- Here, I just illustrated some of the possible algorithms - there is an increasingly huge literature on many other approaches
- aMCMC can just as easily be used in high-dimensional (large p) problems

Some interim comments

- EP-MCMC & aMCMC can be used in many-many settings to vastly speed up computation for big n
- Here, I just illustrated some of the possible algorithms - there is an increasingly huge literature on many other approaches
- aMCMC can just as easily be used in high-dimensional (large p) problems
- It is also certainly possible to combine EP-MCMC + aMCMC

Some interim comments

- EP-MCMC & aMCMC can be used in many-many settings to vastly speed up computation for big n
- Here, I just illustrated some of the possible algorithms - there is an increasingly huge literature on many other approaches
- aMCMC can just as easily be used in high-dimensional (large p) problems
- It is also certainly possible to combine EP-MCMC + aMCMC
- Robustness: one topic we haven't discussed yet is robustness

Robustness in big data

- In standard Bayesian inference, it is assumed that the model is correct.

Robustness in big data

- In standard Bayesian inference, it is assumed that the model is correct.
- Small violations of this assumption sometimes have a large impact, particularly in large datasets

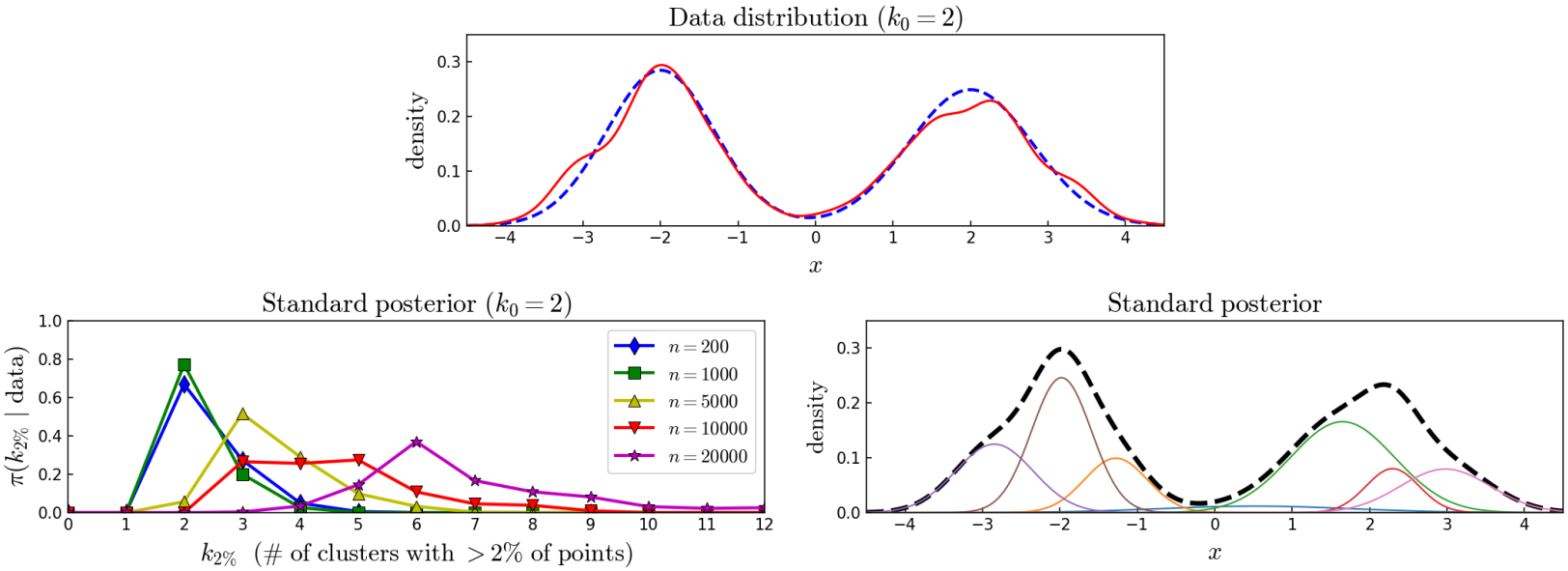
Robustness in big data

- In standard Bayesian inference, it is assumed that the model is correct.
- Small violations of this assumption sometimes have a large impact, particularly in large datasets
- "All models are wrong," & ability to carefully check modeling assumptions decreases for big/complex data

Robustness in big data

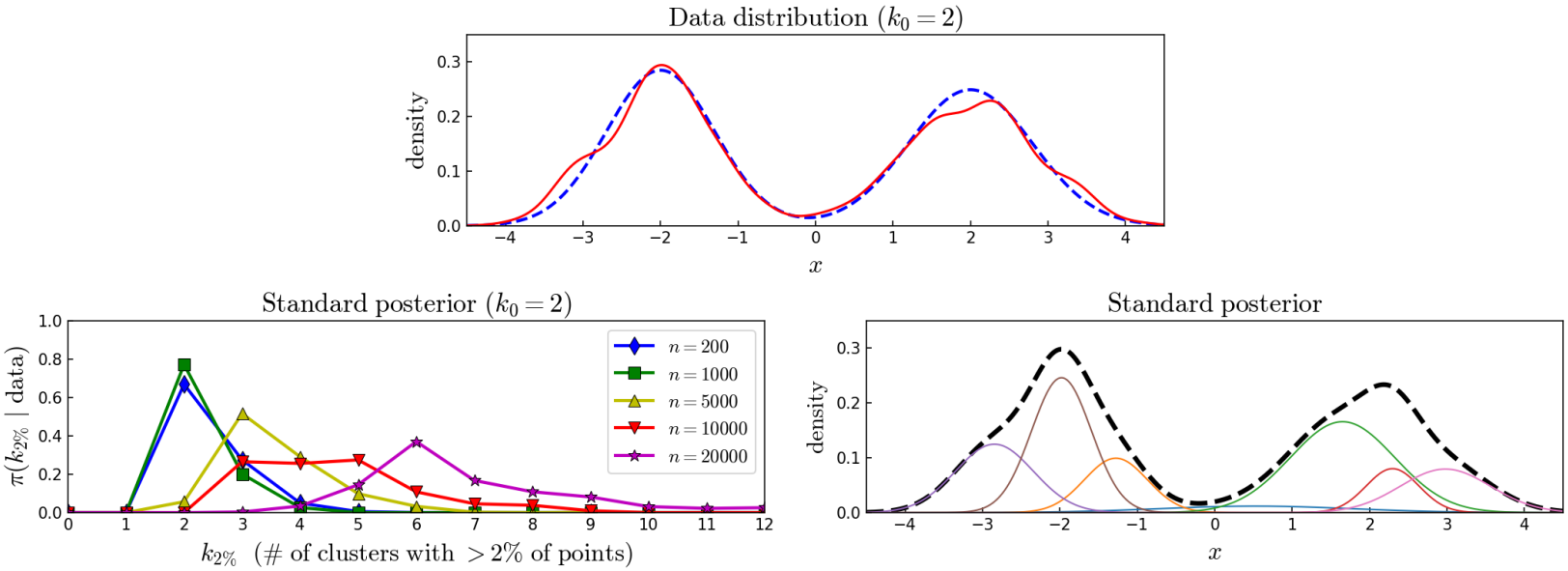
- In standard Bayesian inference, it is assumed that the model is correct.
- Small violations of this assumption sometimes have a large impact, particularly in large datasets
- "All models are wrong," & ability to carefully check modeling assumptions decreases for big/complex data
- Appealing to tweak Bayesian paradigm to be inherently more robust

Example: Perturbed mixture of Gaussians



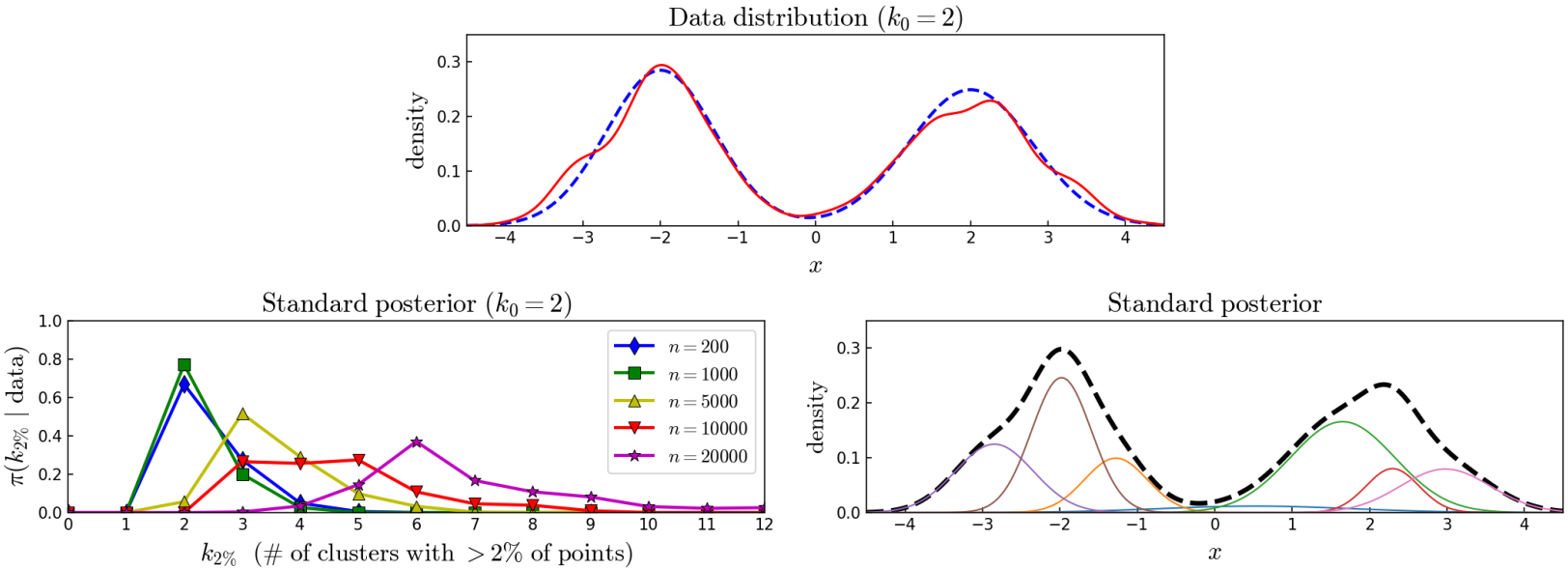
- Mixtures are often used for clustering.

Example: Perturbed mixture of Gaussians



- Mixtures are often used for clustering.
- But if the data distribution is not exactly a mixture from the assumed family, the posterior will tend to introduce more & more clusters as n grows, in order to fit the data.

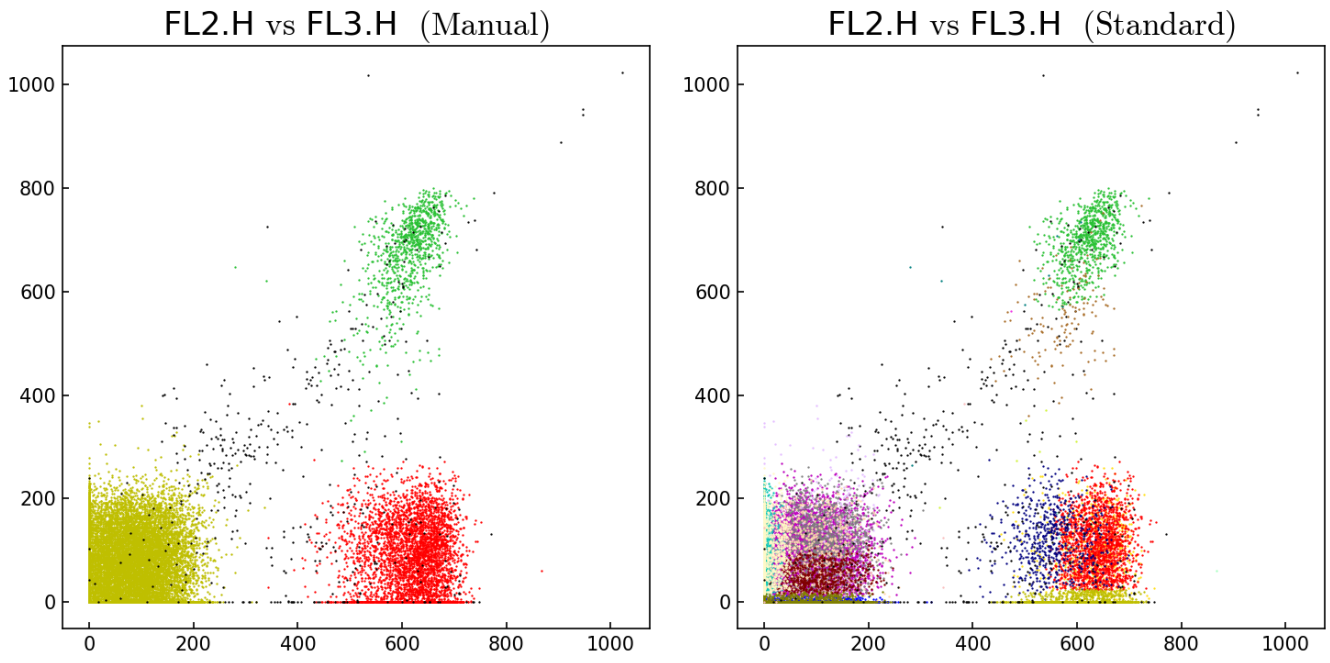
Example: Perturbed mixture of Gaussians



- Mixtures are often used for clustering.
- But if the data distribution is not exactly a mixture from the assumed family, the posterior will tend to introduce more & more clusters as n grows, in order to fit the data.
- As a result, interpretability of clusters may break down.

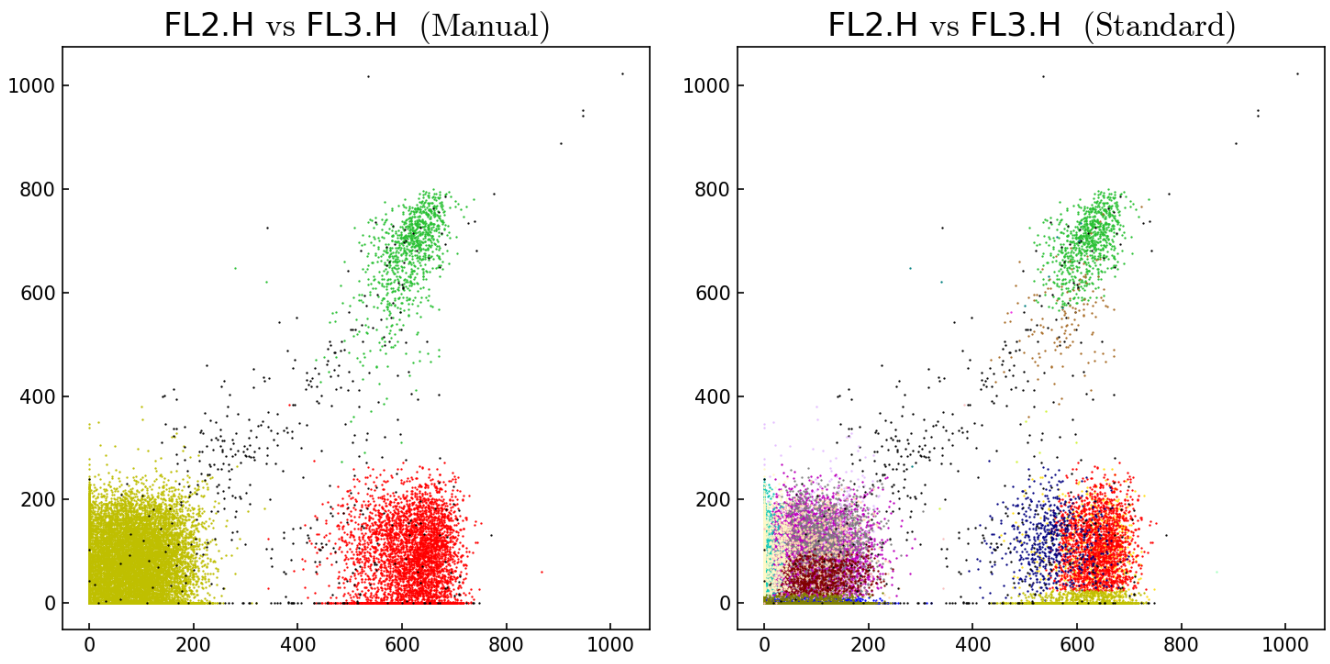
Example: Flow cytometry clustering

- Each sample has 3 to 20-dim measurements on 10K's of cells.



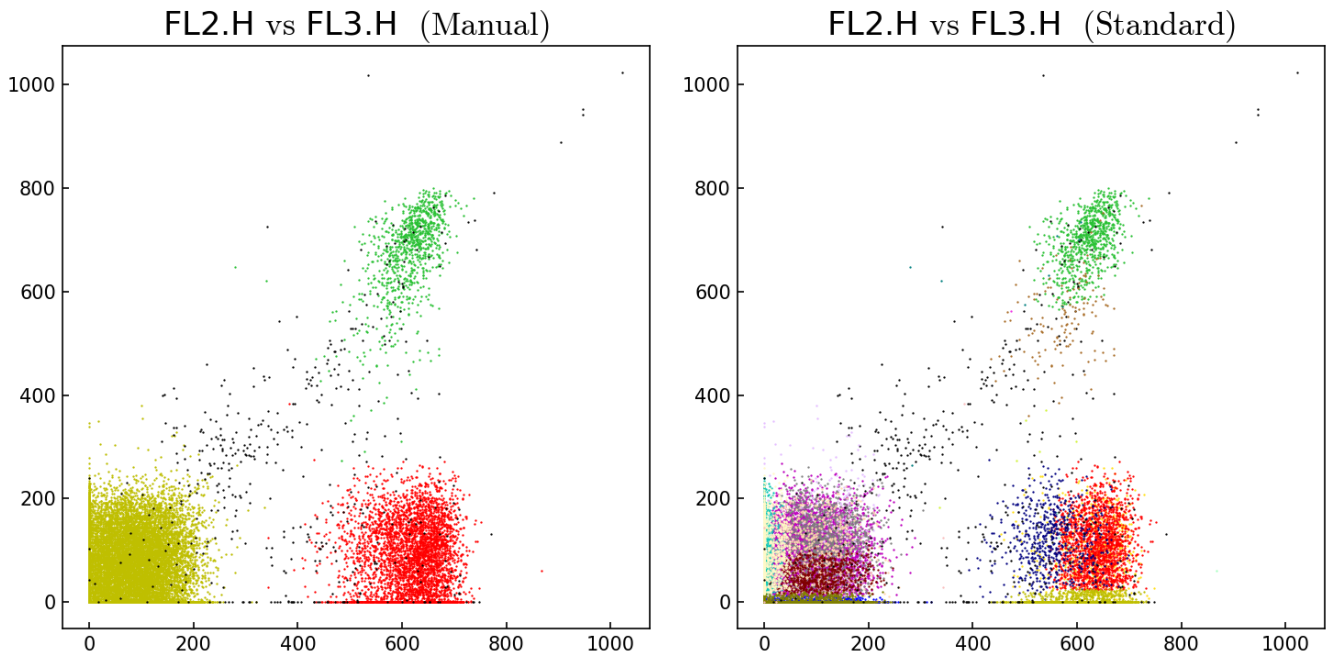
Example: Flow cytometry clustering

- Each sample has 3 to 20-dim measurements on 10K's of cells.
- Manual clustering is time-consuming and subjective.



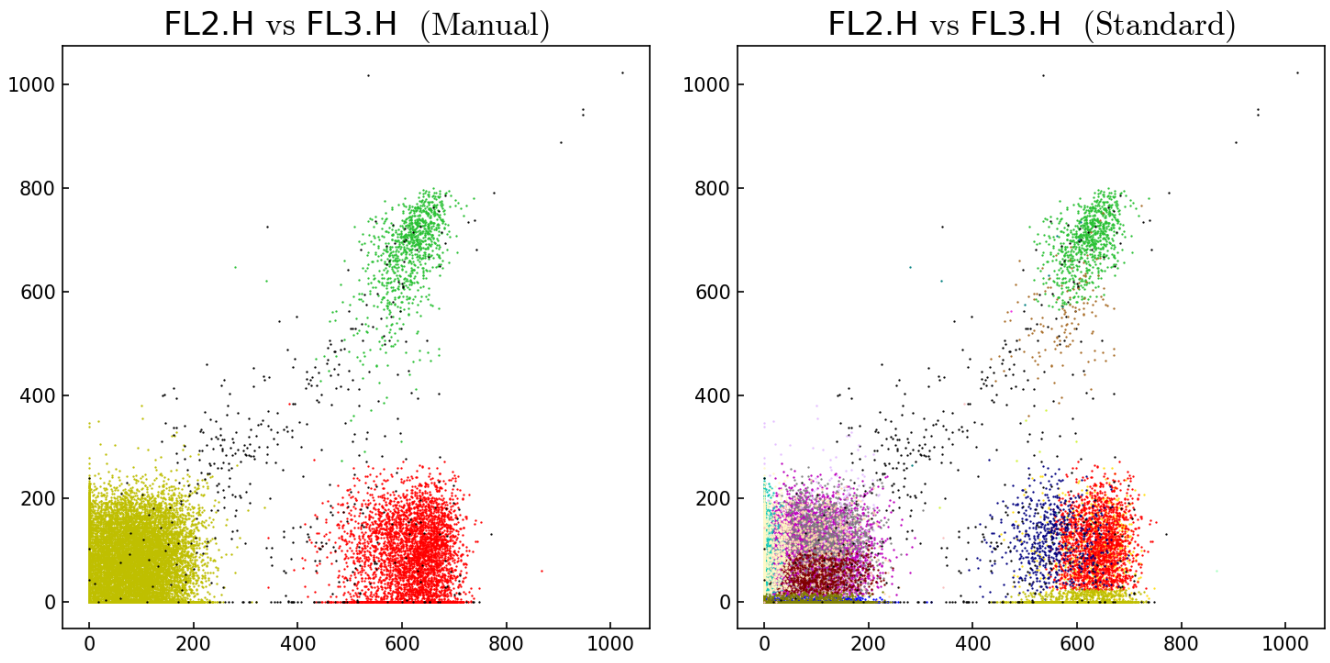
Example: Flow cytometry clustering

- Each sample has 3 to 20-dim measurements on 10K's of cells.
- Manual clustering is time-consuming and subjective.
- Multivariate Gaussian mixture yields too many clusters.



Example: Flow cytometry clustering

- Each sample has 3 to 20-dim measurements on 10K's of cells.
- Manual clustering is time-consuming and subjective.
- Multivariate Gaussian mixture yields too many clusters.
- Example: GvHD data from FLOWCAP-I.



Wait, if the model is wrong, why not just fix it?

- This is often impractical for a number of reasons.

Wait, if the model is wrong, why not just fix it?

- This is often impractical for a number of reasons.
  - insufficient insight into the data generating process

Wait, if the model is wrong, why not just fix it?

- This is often impractical for a number of reasons.
  - insufficient insight into the data generating process
  - time and effort to design model + algorithms, and develop theory

Wait, if the model is wrong, why not just fix it?

- This is often impractical for a number of reasons.
  - insufficient insight into the data generating process
  - time and effort to design model + algorithms, and develop theory
  - slower and more complicated to do inference

Wait, if the model is wrong, why not just fix it?

- This is often impractical for a number of reasons.
  - insufficient insight into the data generating process
  - time and effort to design model + algorithms, and develop theory
  - slower and more complicated to do inference
  - complex models are less likely to be used in practice

Wait, if the model is wrong, why not just fix it?

- This is often impractical for a number of reasons.
  - insufficient insight into the data generating process
  - time and effort to design model + algorithms, and develop theory
  - slower and more complicated to do inference
  - complex models are less likely to be used in practice

Wait, if the model is wrong, why not just fix it?

- This is often impractical for a number of reasons.
  - insufficient insight into the data generating process
  - time and effort to design model + algorithms, and develop theory
  - slower and more complicated to do inference
  - complex models are less likely to be used in practice
- Further, a simple model may be more appropriate, even if wrong.

There are many reasons to prefer simple, interpretable, efficient models. But we need a way to do inference that is robust to misspecification.

Wait, if the model is wrong, why not just fix it?

- This is often impractical for a number of reasons.
  - insufficient insight into the data generating process
  - time and effort to design model + algorithms, and develop theory
  - slower and more complicated to do inference
  - complex models are less likely to be used in practice
- Further, a simple model may be more appropriate, even if wrong.
  - If there is a lack of fit, it may be due to contamination.

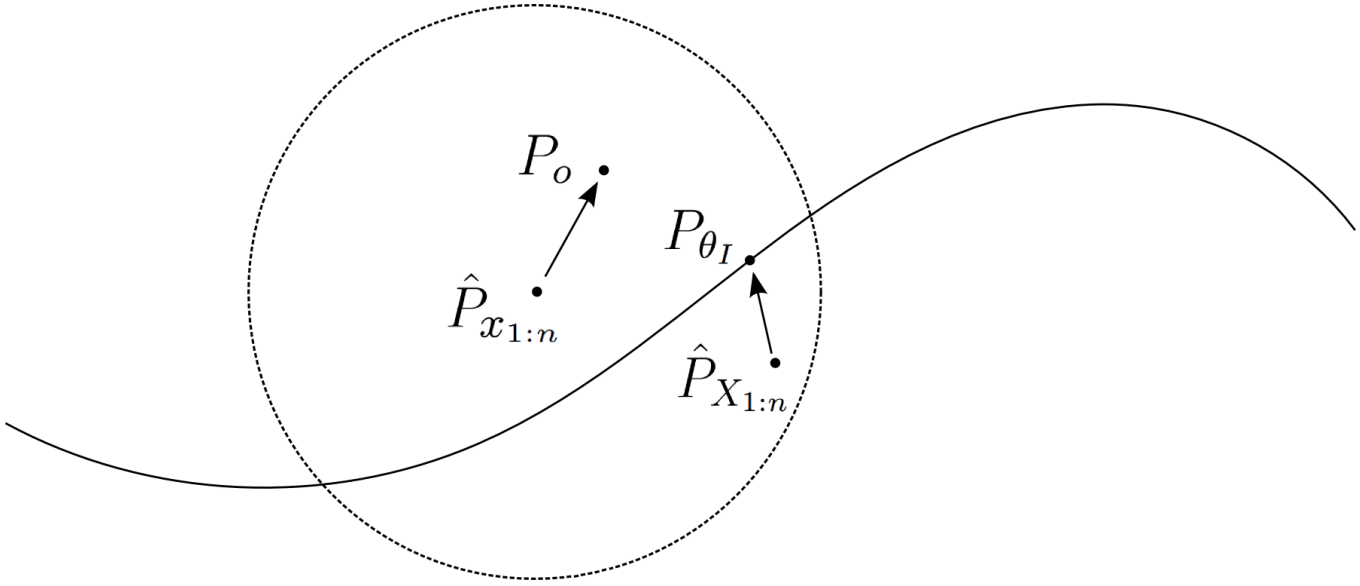
There are many reasons to prefer simple, interpretable, efficient models. But we need a way to do inference that is robust to misspecification.

Wait, if the model is wrong, why not just fix it?

- This is often impractical for a number of reasons.
  - insufficient insight into the data generating process
  - time and effort to design model + algorithms, and develop theory
  - slower and more complicated to do inference
  - complex models are less likely to be used in practice
- Further, a simple model may be more appropriate, even if wrong.
  - If there is a lack of fit, it may be due to contamination.
  - Many models are idealizations that are known to be inexact, but have interpretable parameters that provide insight into the questions of interest.

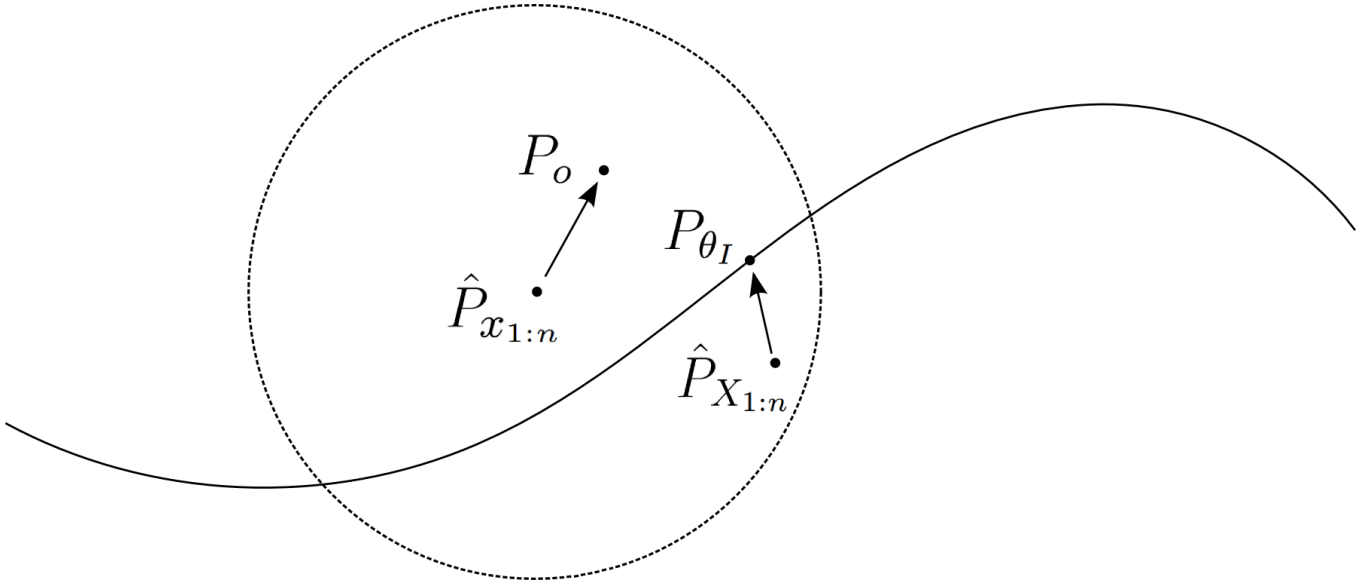
There are many reasons to prefer simple, interpretable, efficient models. But we need a way to do inference that is robust to misspecification.

Coarsened posterior - Miller & Dunson (2018)



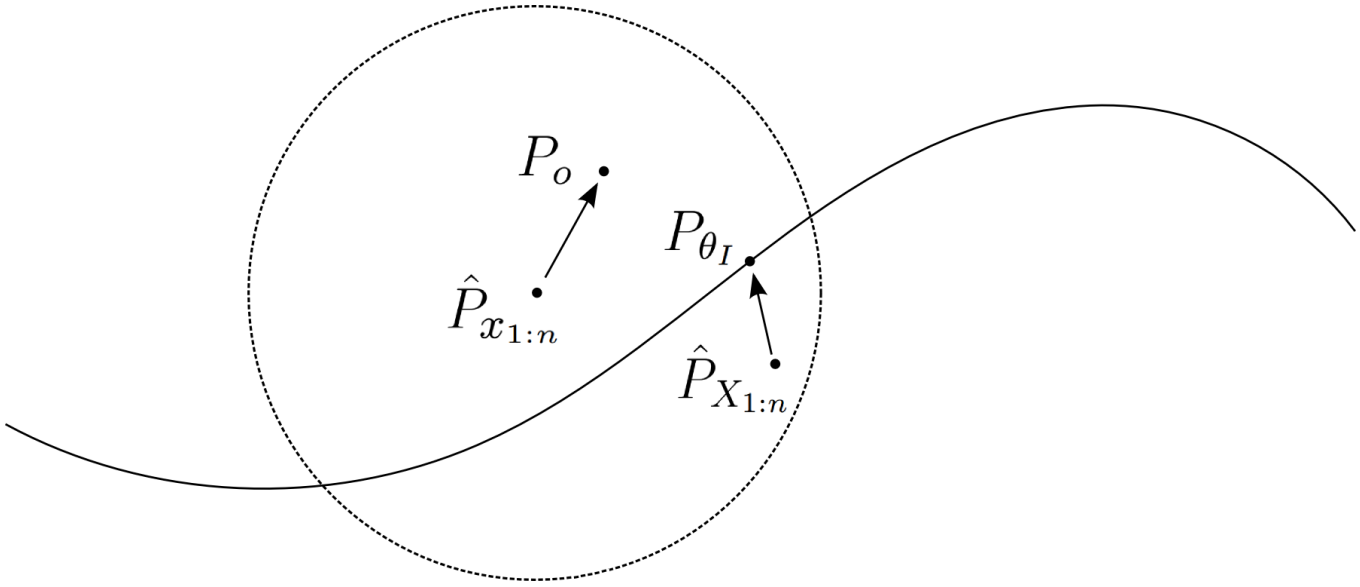
- Assume a model $\{P_\theta : \theta \in \Theta\}$ and a prior $\pi(\theta)$.

Coarsened posterior - Miller & Dunson (2018)



- Assume a model $\{P_\theta : \theta \in \Theta\}$ and a prior $\pi(\theta)$.
- Suppose $\theta_I \in \Theta$ represents the *idealized distribution* of the data.

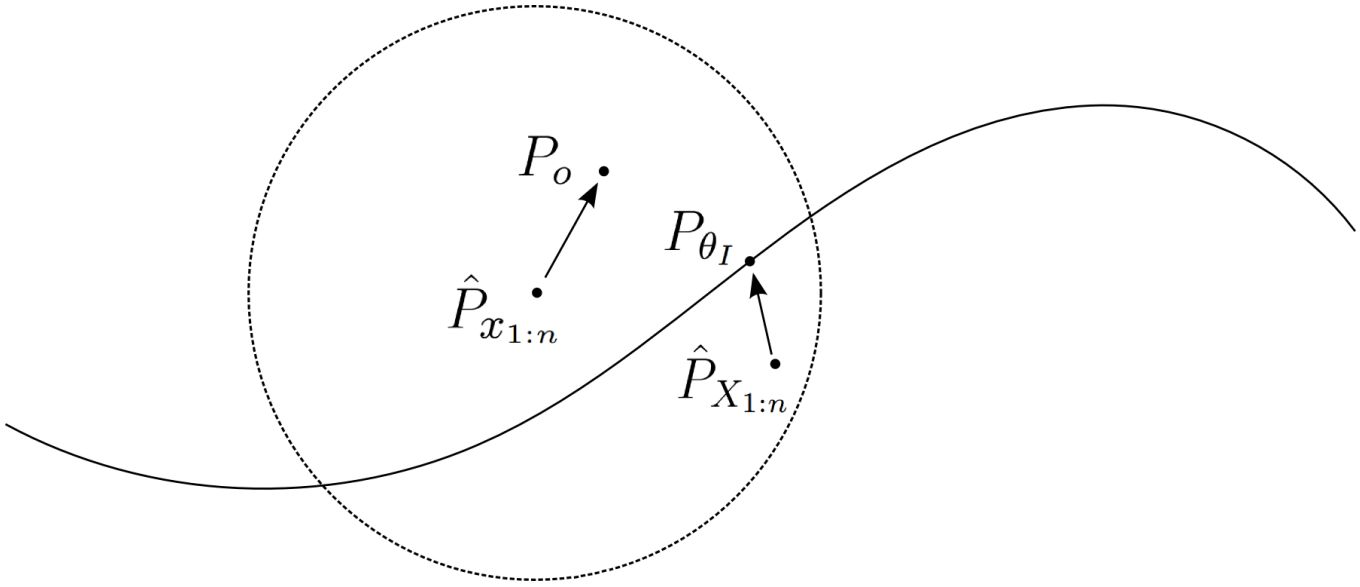
Coarsened posterior - Miller & Dunson (2018)



- Assume a model $\{P_\theta : \theta \in \Theta\}$ and a prior $\pi(\theta)$.
- Suppose $\theta_I \in \Theta$ represents the *idealized distribution* of the data.

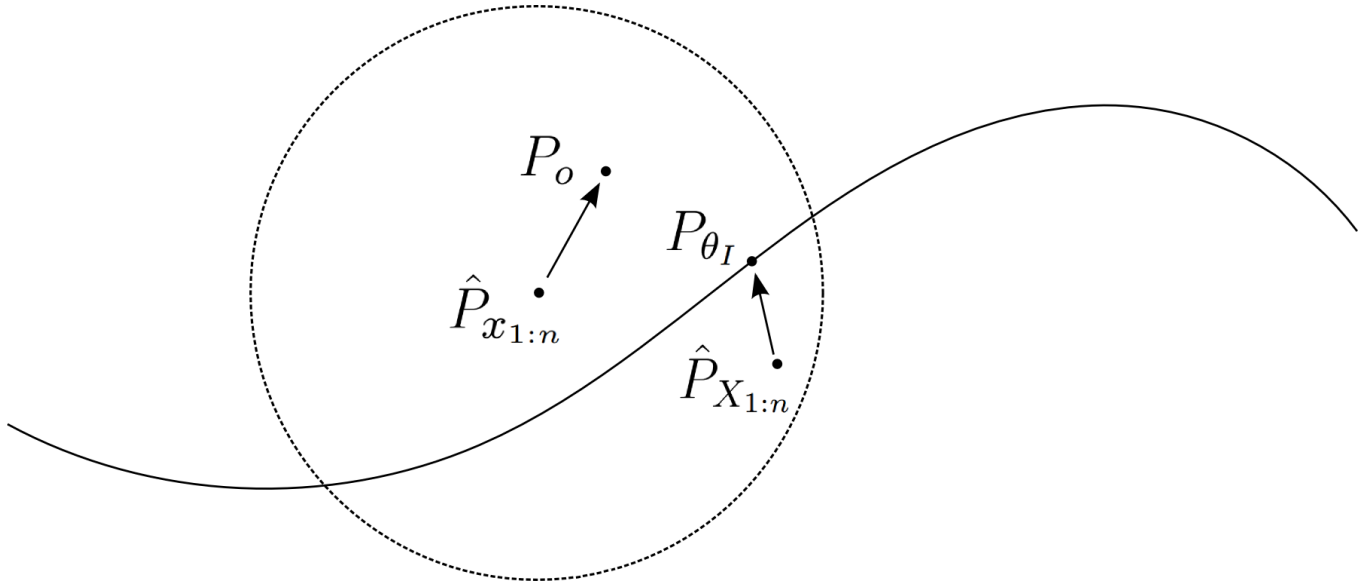
  The interpretation here is that θ_I is the "true" state of nature about which one is interested in making inferences.

Coarsened posterior - Miller & Dunson (2018)



- Assume a model $\{P_\theta : \theta \in \Theta\}$ and a prior $\pi(\theta)$.
- Suppose $\theta_I \in \Theta$ represents the *idealized distribution* of the data.
  The interpretation here is that θ_I is the "true" state of nature about which one is interested in making inferences.
- Suppose $X_1, \ldots, X_n$ i.i.d. $\sim P_{\theta_I}$ are unobserved *idealized data*.

Coarsened posterior - Miller & Dunson (2018)



- Assume a model $\{P_\theta : \theta \in \Theta\}$ and a prior $\pi(\theta)$.
- Suppose $\theta_I \in \Theta$ represents the *idealized distribution* of the data.
  The interpretation here is that θ_I is the "true" state of nature about which one is interested in making inferences.
- Suppose $X_1, \ldots, X_n$ i.i.d. $\sim P_{\theta_I}$ are unobserved *idealized data*.
- However, the *observed data* $x_1, \ldots, x_n$ are actually a slightly corrupted version of $X_1, \ldots, X_n$ in the sense that $d(\hat{P}_{X_{1:n}}, \hat{P}_{x_{1:n}}) < R$ for some statistical distance $d(\cdot, \cdot)$.

Coarsened posterior

- If there were no corruption, then we should use the standard posterior

$$\pi(\theta \mid X_{1:n} = x_{1:n}).$$

Coarsened posterior

- If there were no corruption, then we should use the standard posterior
$$\pi(\theta \mid X_{1:n} = x_{1:n}).$$
- However, due to the corruption this would clearly be incorrect.

Coarsened posterior

- If there were no corruption, then we should use the standard posterior
$$\pi(\theta \mid X_{1:n} = x_{1:n}).$$
- However, due to the corruption this would clearly be incorrect.
- Instead, a natural Bayesian approach would be to condition on what is known, giving us the *coarsened posterior* or *c-posterior*,
$$\pi(\theta \mid d(\hat{P}_{X_{1:n}}, \hat{P}_{x_{1:n}}) < R).$$

Coarsened posterior

- If there were no corruption, then we should use the standard posterior
$$\pi(\theta \mid X_{1:n} = x_{1:n}).$$
- However, due to the corruption this would clearly be incorrect.
- Instead, a natural Bayesian approach would be to condition on what is known, giving us the *coarsened posterior* or *c-posterior*,
$$\pi(\theta \mid d(\hat{P}_{X_{1:n}}, \hat{P}_{x_{1:n}}) < R).$$
- Since R may be difficult to choose *a priori*, put a prior on it: $R \sim H$.

Coarsened posterior

- If there were no corruption, then we should use the standard posterior
$$\pi(\theta \mid X_{1:n} = x_{1:n}).$$
- However, due to the corruption this would clearly be incorrect.
- Instead, a natural Bayesian approach would be to condition on what is known, giving us the *coarsened posterior* or *c-posterior*,
$$\pi(\theta \mid d(\hat{P}_{X_{1:n}}, \hat{P}_{x_{1:n}}) < R).$$
- Since R may be difficult to choose *a priori*, put a prior on it: $R \sim H$.
- More generally, consider
$$\pi\big(\theta \mid d_n(X_{1:n}, x_{1:n}) < R\big)$$
where $d_n(X_{1:n}, x_{1:n}) \geq 0$ is some measure of the discrepancy between $X_{1:n}$ and $x_{1:n}$.

Relative entropy c-posterior ≈ Power posterior

- There are many possible choices of discrepancy but relative entropy works out exceptionally nicely.

Relative entropy c-posterior ≈ Power posterior

- There are many possible choices of discrepancy but relative entropy works out exceptionally nicely.
- Suppose $d_n(X_{1:n}, x_{1:n})$ is a consistent estimator of $D(p_o \| p_\theta)$ when $X_i \overset{iid}{\sim} p_\theta$ and $x_i \overset{iid}{\sim} p_o$.

Relative entropy c-posterior ≈ Power posterior

- There are many possible choices of discrepancy but relative entropy works out exceptionally nicely.
- Suppose $d_n(X_{1:n}, x_{1:n})$ is a consistent estimator of $D(p_o \| p_\theta)$ when $X_i \overset{iid}{\sim} p_\theta$ and $x_i \overset{iid}{\sim} p_o$.
- When $R \sim \exp(\alpha)$, we have the *power posterior* approximation,

$$\pi\big(\theta \,|\, d_n(X_{1:n}, x_{1:n}) < R\big) \;\propto\; \pi(\theta) \prod_{i=1}^{n} p_\theta(x_i)^{\zeta_n}$$

where $\zeta_n = \alpha/(\alpha + n)$.

Relative entropy c-posterior ≈ Power posterior

- There are many possible choices of discrepancy but relative entropy works out exceptionally nicely.
- Suppose $d_n(X_{1:n}, x_{1:n})$ is a consistent estimator of $D(p_o \| p_\theta)$ when $X_i \overset{iid}{\sim} p_\theta$ and $x_i \overset{iid}{\sim} p_o$.
- When $R \sim \exp(\alpha)$, we have the *power posterior* approximation,

$$\pi\big(\theta \,|\, d_n(X_{1:n}, x_{1:n}) < R\big) \propto \pi(\theta) \prod_{i=1}^{n} p_\theta(x_i)^{\zeta_n}$$

  where $\zeta_n = \alpha/(\alpha + n)$.
- The power posterior enables inference using standard techniques:

Relative entropy c-posterior ≈ Power posterior

- There are many possible choices of discrepancy but relative entropy works out exceptionally nicely.
- Suppose $d_n(X_{1:n}, x_{1:n})$ is a consistent estimator of $D(p_o \| p_\theta)$ when $X_i \overset{iid}{\sim} p_\theta$ and $x_i \overset{iid}{\sim} p_o$.
- When $R \sim \exp(\alpha)$, we have the *power posterior* approximation,

$$\pi\big(\theta \,|\, d_n(X_{1:n}, x_{1:n}) < R\big) \;\propto\; \pi(\theta) \prod_{i=1}^{n} p_\theta(x_i)^{\zeta_n}$$

  where $\zeta_n = \alpha/(\alpha + n)$.
- The power posterior enables inference using standard techniques:
  - Analytical solutions in the case of conjugate priors

Relative entropy c-posterior ≈ Power posterior

- There are many possible choices of discrepancy but relative entropy works out exceptionally nicely.
- Suppose $d_n(X_{1:n}, x_{1:n})$ is a consistent estimator of $D(p_o \| p_\theta)$ when $X_i \overset{iid}{\sim} p_\theta$ and $x_i \overset{iid}{\sim} p_o$.
- When $R \sim \exp(\alpha)$, we have the *power posterior* approximation,

$$\pi\big(\theta \,|\, d_n(X_{1:n}, x_{1:n}) < R\big) \;\propto\; \pi(\theta) \prod_{i=1}^{n} p_\theta(x_i)^{\zeta_n}$$

  where $\zeta_n = \alpha/(\alpha + n)$.
- The power posterior enables inference using standard techniques:
  - Analytical solutions in the case of conjugate priors
  - MCMC is also straightforward

Toy example: Bernoulli trials

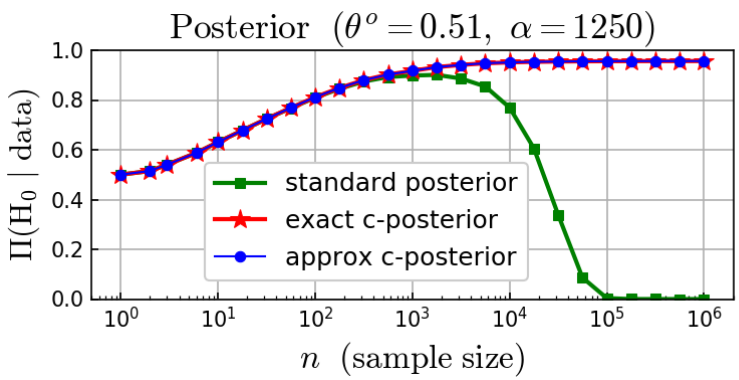
- Suppose $H_0 : \theta = 0.5$ is true; e.g, heads & tails are equally likely in repeated coin flips

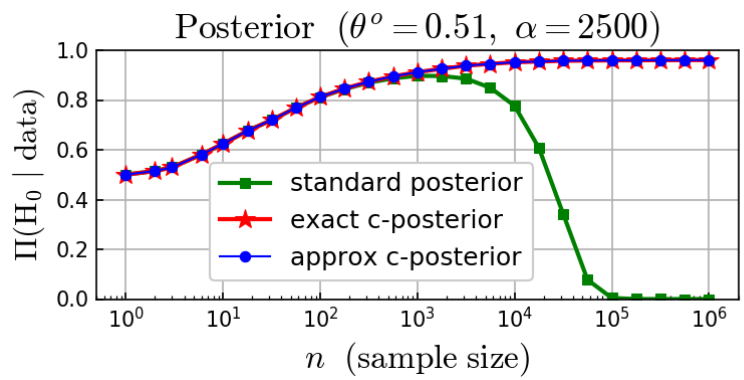
Toy example: Bernoulli trials

- Suppose $H_0 : \theta = 0.5$ is true; e.g, heads & tails are equally likely in repeated coin flips
- But $x_1, \ldots, x_n$ are corrupted and behave like Bernoulli(0.51) samples.

Toy example: Bernoulli trials

- Suppose $H_0 : \theta = 0.5$ is true; e.g, heads & tails are equally likely in repeated coin flips
- But $x_1, \ldots, x_n$ are corrupted and behave like Bernoulli(0.51) samples.
- The c-posterior is robust to this, but the standard posterior is not.





Mixture models

- Model: $X_1, \ldots, X_n | w, \varphi$ i.i.d. $\sim \sum_{i=1}^{K} w_i f_{\varphi_i}(x)$

Mixture models

- Model: $X_1, \ldots, X_n | w, \varphi$ i.i.d. $\sim \sum_{i=1}^{K} w_i f_{\varphi_i}(x)$
- Prior: $w \sim \text{Dirichlet}(\gamma_1, \ldots, \gamma_K)$ and $\varphi_1, \ldots, \varphi_K \overset{iid}{\sim} H$.

Mixture models

- Model: $X_1, \ldots, X_n | w, \varphi$ i.i.d. $\sim \sum_{i=1}^{K} w_i f_{\varphi_i}(x)$
- Prior: $w \sim \text{Dirichlet}(\gamma_1, \ldots, \gamma_K)$ and $\varphi_1, \ldots, \varphi_K \overset{iid}{\sim} H$.
- c-Posterior is approximated as

$$\pi\big(w, \varphi \,|\, d_n(X_{1:n}, x_{1:n}) < R\big) \propto \pi(w, \varphi) \prod_{j=1}^{n} \Big(\sum_{i=1}^{K} w_i f_{\varphi_i}(x_j) \Big)^{\zeta_n}$$

where $\zeta_n = \alpha / (\alpha + n)$.

Mixture models

- Model: $X_1, \ldots, X_n | w, \varphi$ i.i.d. $\sim \sum_{i=1}^{K} w_i f_{\varphi_i}(x)$
- Prior: $w \sim \text{Dirichlet}(\gamma_1, \ldots, \gamma_K)$ and $\varphi_1, \ldots, \varphi_K \overset{iid}{\sim} H$.
- c-Posterior is approximated as

$$\pi\big(w, \varphi \,|\, d_n(X_{1:n}, x_{1:n}) < R\big) \propto \pi(w, \varphi) \prod_{j=1}^{n} \Big(\sum_{i=1}^{K} w_i f_{\varphi_i}(x_j) \Big)^{\zeta_n}$$

  where $\zeta_n = \alpha / (\alpha + n)$.
- A straightforward MCMC algorithm can be used for computation

Mixture models

- Model: $X_1, \ldots, X_n | w, \varphi$ i.i.d. $\sim \sum_{i=1}^{K} w_i f_{\varphi_i}(x)$
- Prior: $w \sim \text{Dirichlet}(\gamma_1, \ldots, \gamma_K)$ and $\varphi_1, \ldots, \varphi_K \overset{iid}{\sim} H$.
- c-Posterior is approximated as

$$\pi\big(w, \varphi \,|\, d_n(X_{1:n}, x_{1:n}) < R\big) \propto \pi(w, \varphi) \prod_{j=1}^{n} \Big(\sum_{i=1}^{K} w_i f_{\varphi_i}(x_j) \Big)^{\zeta_n}$$

  where $\zeta_n = \alpha / (\alpha + n)$.
- A straightforward MCMC algorithm can be used for computation
- Scales well to large datasets

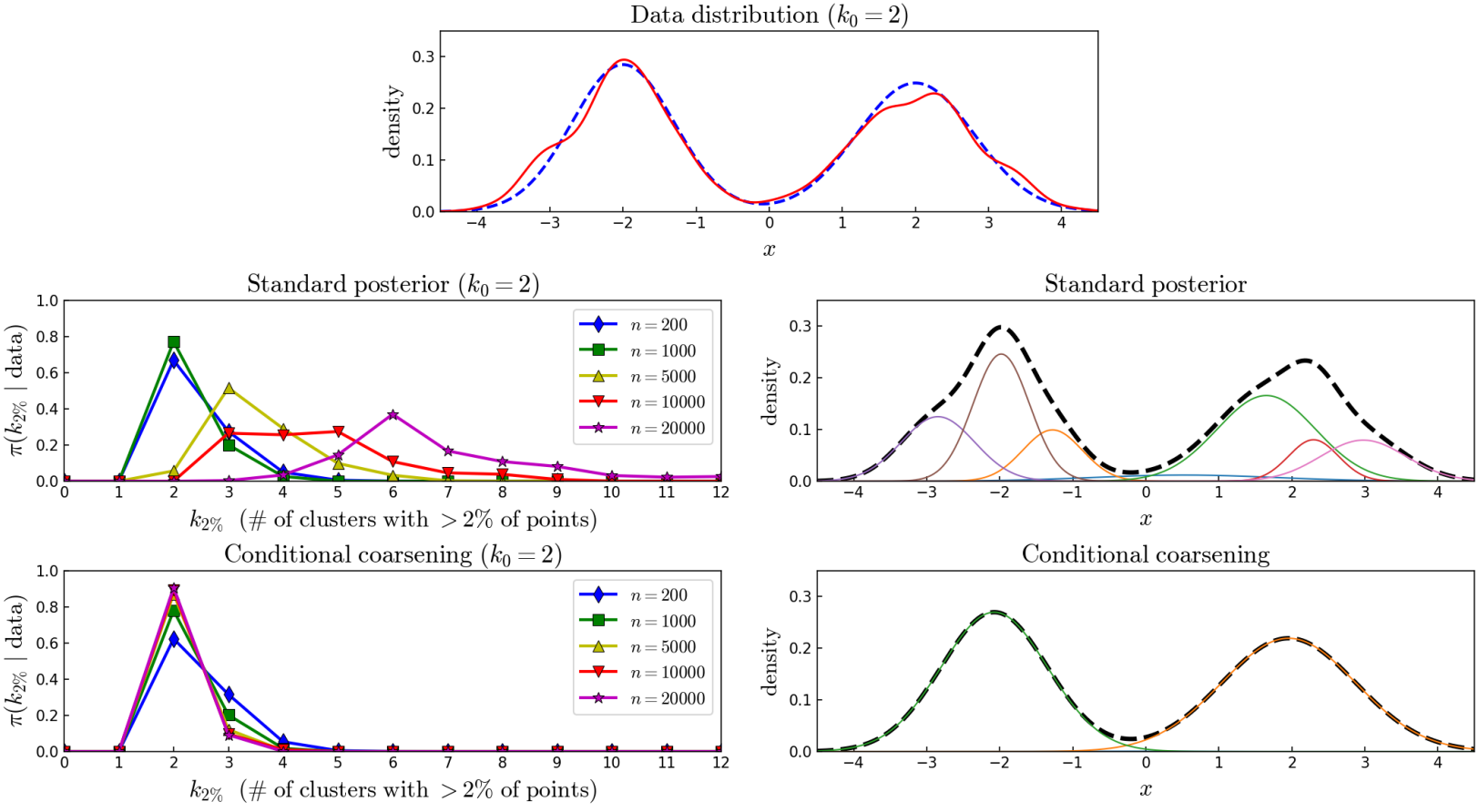
Mixture models

- Model: $X_1, \ldots, X_n | w, \varphi$ i.i.d. $\sim \sum_{i=1}^{K} w_i f_{\varphi_i}(x)$
- Prior: $w \sim \text{Dirichlet}(\gamma_1, \ldots, \gamma_K)$ and $\varphi_1, \ldots, \varphi_K \overset{iid}{\sim} H$.
- c-Posterior is approximated as

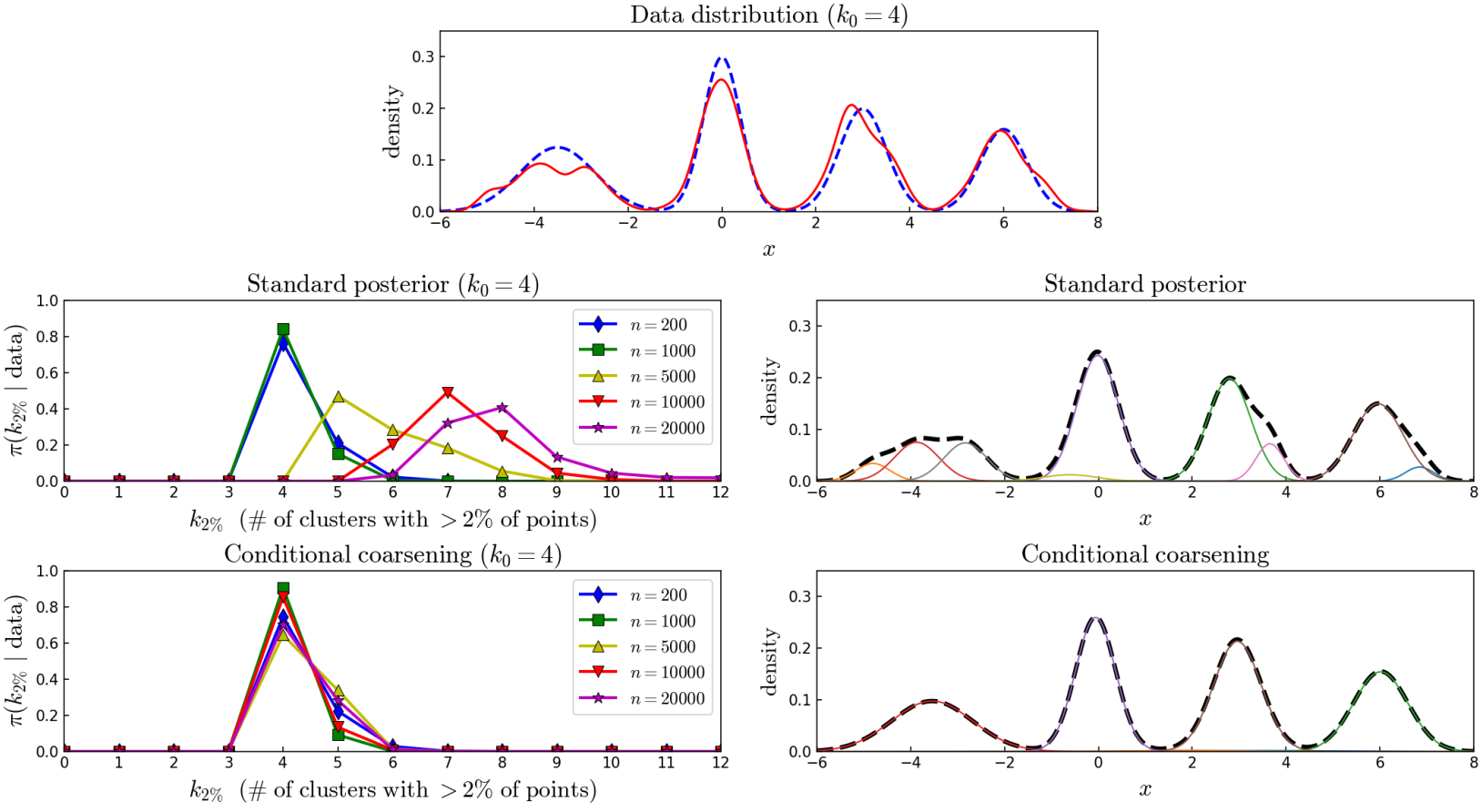
$$\pi\big(w, \varphi \,|\, d_n(X_{1:n}, x_{1:n}) < R\big) \propto \pi(w, \varphi) \prod_{j=1}^{n} \Big(\sum_{i=1}^{K} w_i f_{\varphi_i}(x_j) \Big)^{\zeta_n}$$

  where $\zeta_n = \alpha / (\alpha + n)$.
- A straightforward MCMC algorithm can be used for computation
- Scales well to large datasets
- EP-MCMC, a-MCMC etc can be used to enhance scalability

Example: Perturbed mixture of Gaussians

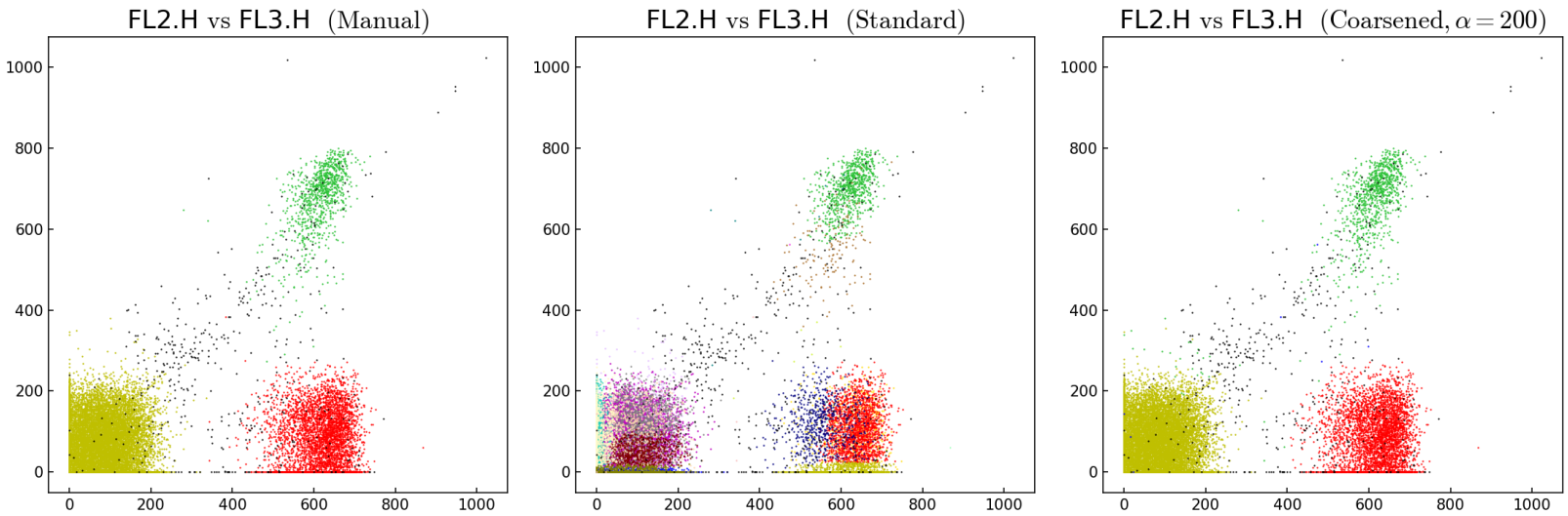


Example: Perturbed mixture of Gaussians



Results: Flow cytometry clustering

Clustering on test datasets closely matches manual ground truth.



Results: Flow cytometry clustering

Table 1: Average F-measures on the flow cytometry test set (GvHD datasets 7–12).

	7	8	9	10	11	12
Standard	0.532	0.478	0.619	0.453	0.542	0.585
Coarsened	0.667	0.875	0.931	0.998	0.989	0.993

- Clustering on test datasets closely matches manual ground truth.
- Use F-measure to quantify similarity of partitions $\mathscr{A}$ and $\mathscr{B}$:

$$F(\mathscr{A},\mathscr{B}) = \sum_{A\in\mathscr{A}} \frac{|A|}{N} \max_{B\in\mathscr{B}} \frac{2|A\cap B|}{|A|+|B|}.$$

c-Bayes discussion

- c-Bayes provides a framework for improving robustness to model misspecification

c-Bayes discussion

- c-Bayes provides a framework for improving robustness to model misspecification
- Particularly useful when interest is in model-based inferences & sample size n is large

c-Bayes discussion

- c-Bayes provides a framework for improving robustness to model misspecification
- Particularly useful when interest is in model-based inferences & sample size n is large
- If we just want a black box for prediction may as well let the model grow (unnecessarily) in complexity with n

c-Bayes discussion

- c-Bayes provides a framework for improving robustness to model misspecification
- Particularly useful when interest is in model-based inferences & sample size n is large
- If we just want a black box for prediction may as well let the model grow (unnecessarily) in complexity with n
- c-Bayes can be implemented with a particular power posterior

c-Bayes discussion

- c-Bayes provides a framework for improving robustness to model misspecification
- Particularly useful when interest is in model-based inferences & sample size n is large
- If we just want a black box for prediction may as well let the model grow (unnecessarily) in complexity with n
- c-Bayes can be implemented with a particular power posterior
- All the scalable MCMC tricks developed for regular posteriors can be used directly

c-Bayes discussion

- c-Bayes provides a framework for improving robustness to model misspecification
- Particularly useful when interest is in model-based inferences & sample size n is large
- If we just want a black box for prediction may as well let the model grow (unnecessarily) in complexity with n
- c-Bayes can be implemented with a particular power posterior
- All the scalable MCMC tricks developed for regular posteriors can be used directly
- Also provides a motivation for doing Bayesian inferences based on subsamples

Hybrid high-dimensional density estimation

Ye, Canale & Dunson (2016, AISTATS)

- $y_i = (y_{i1}, \ldots, y_{ip})^T \sim f$ with p large & f an unknown density

Hybrid high-dimensional density estimation

Ye, Canale & Dunson (2016, AISTATS)

- $y_i = (y_{i1}, \ldots, y_{ip})^T \sim f$ with p large & f an unknown density
- Potentially use Dirichlet process mixtures of factor models

Hybrid high-dimensional density estimation

Ye, Canale & Dunson (2016, AISTATS)

- $y_i = (y_{i1}, \ldots, y_{ip})^T \sim f$ with p large & f an unknown density
- Potentially use Dirichlet process mixtures of factor models
- Approach doesn't scale well at all with p

Hybrid high-dimensional density estimation

Ye, Canale & Dunson (2016, AISTATS)

- $y_i = (y_{i1}, \ldots, y_{ip})^T \sim f$ with p large & f an unknown density
- Potentially use Dirichlet process mixtures of factor models
- Approach doesn't scale well at all with p
- Instead use hybrid of Gibbs sampling & fast multiscale SVD

Hybrid high-dimensional density estimation

Ye, Canale & Dunson (2016, AISTATS)

- $y_i = (y_{i1}, \ldots, y_{ip})^T \sim f$ with p large & f an unknown density
- Potentially use Dirichlet process mixtures of factor models
- Approach doesn't scale well at all with p
- Instead use hybrid of Gibbs sampling & fast multiscale SVD
- Scalable, excellent mixing & empirical/predictive performance

Outline

Scaling Bayes to high-dimensional data

- Thus far we have focused on solving computational & robustness problems arising in large n

Scaling Bayes to high-dimensional data

- Thus far we have focused on solving computational & robustness problems arising in large n
- In many ways these problems are easier to deal with then issues with high-dimensional/complex data

Scaling Bayes to high-dimensional data

- Thus far we have focused on solving computational & robustness problems arising in large n
- In many ways these problems are easier to deal with then issues with high-dimensional/complex data
- For example, in biomedical studies we routinely measure HUGE numbers of features/study subjects

Scaling Bayes to high-dimensional data

- Thus far we have focused on solving computational & robustness problems arising in large n
- In many ways these problems are easier to deal with then issues with high-dimensional/complex data
- For example, in biomedical studies we routinely measure HUGE numbers of features/study subjects
- Genomics, precision medicine, neuroimaging, etc

Scaling Bayes to high-dimensional data

- Thus far we have focused on solving computational & robustness problems arising in large n
- In many ways these problems are easier to deal with then issues with high-dimensional/complex data
- For example, in biomedical studies we routinely measure HUGE numbers of features/study subjects
- Genomics, precision medicine, neuroimaging, etc
- We have very few labeled data relative to data dimensionality p

Scaling Bayes to high-dimensional data

- Thus far we have focused on solving computational & robustness problems arising in large n
- In many ways these problems are easier to deal with then issues with high-dimensional/complex data
- For example, in biomedical studies we routinely measure HUGE numbers of features/study subjects
- Genomics, precision medicine, neuroimaging, etc
- We have very few labeled data relative to data dimensionality p
- We also don't want a black box for prediction but want to do scientific inferences

Scaling Bayes to high-dimensional data

- Thus far we have focused on solving computational & robustness problems arising in large n
- In many ways these problems are easier to deal with then issues with high-dimensional/complex data
- For example, in biomedical studies we routinely measure HUGE numbers of features/study subjects
- Genomics, precision medicine, neuroimaging, etc
- We have very few labeled data relative to data dimensionality p
- We also don't want a black box for prediction but want to do scientific inferences
- Bayes for big p is a huge topic - I'll just provide some vignettes to give a flavor

Variable/feature selection in large p regression

- Huge focus in sciences on variable selection

Variable/feature selection in large p regression

- Huge focus in sciences on variable selection
- For example, select the genetic variants x_j associated with a response (phenotype) y

Variable/feature selection in large p regression

- Huge focus in sciences on variable selection
- For example, select the genetic variants x_j associated with a response (phenotype) y
- Sample size n is modest & # genetic variants p is huge

Variable/feature selection in large p regression

- Huge focus in sciences on variable selection
- For example, select the genetic variants x_j associated with a response (phenotype) y
- Sample size n is modest & # genetic variants p is huge
- Large p, small n problem

Variable/feature selection in large p regression

- Huge focus in sciences on variable selection
- For example, select the genetic variants x_j associated with a response (phenotype) y
- Sample size n is modest & # genetic variants p is huge
- Large p, small n problem
- Huge literature for dealing with this problem

Variable/feature selection in large p regression

- Huge focus in sciences on variable selection
- For example, select the genetic variants x_j associated with a response (phenotype) y
- Sample size n is modest & # genetic variants p is huge
- Large p, small n problem
- Huge literature for dealing with this problem
- Two main approaches:

Variable/feature selection in large p regression

- Huge focus in sciences on variable selection
- For example, select the genetic variants x_j associated with a response (phenotype) y
- Sample size n is modest & # genetic variants p is huge
- Large p, small n problem
- Huge literature for dealing with this problem
- Two main approaches:
  1. Independent Screening

Variable/feature selection in large p regression

- Huge focus in sciences on variable selection
- For example, select the genetic variants x_j associated with a response (phenotype) y
- Sample size n is modest & # genetic variants p is huge
- Large p, small n problem
- Huge literature for dealing with this problem
- Two main approaches:
  1. Independent Screening
  2. Penalized estimation/shrinkage

Independent Screening

- Test for an association between two variables at a time (e.g, a phenotype & a SNP)

Independent Screening

- Test for an association between two variables at a time (e.g, a phenotype & a SNP)
- Repeat this for all possible pairs, getting a large number of p-values

Independent Screening

- Test for an association between two variables at a time (e.g, a phenotype & a SNP)
- Repeat this for all possible pairs, getting a large number of p-values
- Choose p-value threshold controlling False Discovery Rate (FDR) - eg Benjamini-Hochberg (BH)

Independent Screening

- Test for an association between two variables at a time (e.g, a phenotype & a SNP)
- Repeat this for all possible pairs, getting a large number of p-values
- Choose p-value threshold controlling False Discovery Rate (FDR) - eg Benjamini-Hochberg (BH)
- Get a list of discoveries & hopefully run follow-up studies to verify

Independent Screening - Continued

- Very appealing in its simplicity

Independent Screening - Continued

- Very appealing in its simplicity
- Very widely used

Independent Screening - Continued

- Very appealing in its simplicity
- Very widely used
- Many false positives & negatives; for sparse data false negatives huge problem

Independent Screening - Continued

- Very appealing in its simplicity
- Very widely used
- Many false positives & negatives; for sparse data false negatives huge problem
- Just considering a pair of variables at a time leads to limited insights

Problems with classical approaches

- Consider the canonical linear regression problem:

$$y_i = x_i'\beta + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2),$$

where $x_i = (x_{i1}, \ldots, x_{ip})'$ & $\beta = (\beta_1, \ldots, \beta_p)'$

Problems with classical approaches

- Consider the canonical linear regression problem:

$$y_i = x_i'\beta + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2),$$

  where $x_i = (x_{i1}, \ldots, x_{ip})'$ & $\beta = (\beta_1, \ldots, \beta_p)'$

- The classical approach is to estimate β using MLE which reduces to the least squares estimator $\hat{\beta} = (X'X)^{-1}X'y$

Problems with classical approaches

- Consider the canonical linear regression problem:

$$y_i = x_i'\beta + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2),$$

  where $x_i = (x_{i1}, \ldots, x_{ip})'$ & $\beta = (\beta_1, \ldots, \beta_p)'$
- The classical approach is to estimate β using MLE which reduces to the least squares estimator $\hat{\beta} = (X'X)^{-1}X'y$
- Unfortunately as p increases OR x_{ij}s become more correlated OR more sparse, the variance of $\hat{\beta}$ blows up

Problems with classical approaches

- Consider the canonical linear regression problem:

$$y_i = x_i'\beta + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2),$$

  where $x_i = (x_{i1}, \ldots, x_{ip})'$ & $\beta = (\beta_1, \ldots, \beta_p)'$
- The classical approach is to estimate β using MLE which reduces to the least squares estimator $\hat{\beta} = (X'X)^{-1}X'y$
- Unfortunately as p increases OR x_{ij}s become more correlated OR more sparse, the variance of $\hat{\beta}$ blows up
- For $p > n$ a unique MLE doesn't exist

Including prior information

- We need to include some sort of outside or *prior* information

Including prior information

- We need to include some sort of outside or *prior* information
- In a Bayesian approach, we choose a prior probability distribution $\pi(\beta)$ characterizing our uncertainty in β prior to observing the current data

Including prior information

- We need to include some sort of outside or *prior* information
- In a Bayesian approach, we choose a prior probability distribution $\pi(\beta)$ characterizing our uncertainty in β prior to observing the current data
- Then, we would use Bayes rule to update the prior with information in the likelihood:

$$\pi(\beta|Y,X) = \frac{\pi(\beta)L(Y|X,\beta)}{\int \pi(\beta)L(Y|X,\beta)d\beta} = \frac{\pi(\beta)L(Y|X,\beta)}{L(Y|X)},$$

where $L(Y|X,\beta)$ is the likelihood & $L(Y|X)$ is the marginal likelihood

Bayes in normal linear regression

- Suppose $\pi(\beta) = N_p(0, \Sigma_0)$ & we have a normal linear regression model

Bayes in normal linear regression

- Suppose $\pi(\beta) = N_p(0, \Sigma_0)$ & we have a normal linear regression model
- Then, the posterior distribution of β has a simple form as

$$\pi(\beta|Y, X) = N_p(\tilde{\beta}, V_\beta)$$

Bayes in normal linear regression

- Suppose $\pi(\beta) = N_p(0, \Sigma_0)$ & we have a normal linear regression model
- Then, the posterior distribution of β has a simple form as

$$\pi(\beta|Y, X) = N_p(\tilde{\beta}, V_\beta)$$

- Posterior covariance $V_\beta = (\Sigma_0^{-1} + \sigma^{-2} X'X)^{-1}$ combines the two sources of information

Bayes in normal linear regression

- Suppose $\pi(\beta) = N_p(0, \Sigma_0)$ & we have a normal linear regression model
- Then, the posterior distribution of β has a simple form as

$$\pi(\beta|Y, X) = N_p(\tilde{\beta}, V_\beta)$$

- Posterior covariance $V_\beta = (\Sigma_0^{-1} + \sigma^{-2} X'X)^{-1}$ combines the two sources of information
- The posterior mean is $\tilde{\beta} = (\sigma^2 \Sigma_0^{-1} + X'X)^{-1} X'Y$, which is a weighted average of 0 and $\hat{\beta} = (X'X)^{-1} X'Y$.

Penalized estimation

- We can get the same estimator for β by solving:

$$\begin{aligned}\tilde{\beta} &= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - x_i'\beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \\ &= \underset{\beta}{\operatorname{argmin}} ||Y - X\beta||_2^2 + \lambda ||\beta||_2^2.\end{aligned}$$

Penalized estimation

- We can get the same estimator for β by solving:

$$\begin{aligned}\tilde{\beta} &= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - x_i'\beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \\ &= \underset{\beta}{\operatorname{argmin}} ||Y - X\beta||_2^2 + \lambda ||\beta||_2^2.\end{aligned}$$

- Known as ridge or L2 *penalized* regression

Penalized estimation

- We can get the same estimator for β by solving:

$$\begin{aligned}\tilde{\beta} &= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - x_i'\beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \\ &= \underset{\beta}{\operatorname{argmin}} ||Y - X\beta||_2^2 + \lambda ||\beta||_2^2.\end{aligned}$$

- Known as ridge or L2 *penalized* regression
- Dual interpretation as a Bayesian estimator under a Gaussian prior centered at zero & a least squares estimator with a penalty on large coefficients

Penalized estimation

- We can get the same estimator for β by solving:

$$\begin{aligned}\tilde{\beta} &= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - x_i'\beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \\ &= \underset{\beta}{\operatorname{argmin}} ||Y - X\beta||_2^2 + \lambda ||\beta||_2^2.\end{aligned}$$

- Known as ridge or L2 *penalized* regression
- Dual interpretation as a Bayesian estimator under a Gaussian prior centered at zero & a least squares estimator with a penalty on large coefficients
- Such estimators introduce some bias while reducing the variance a lot to improve mean square error

L1 - Lasso sparse estimation

- The above penalized loss function can be generalized as

$$\tilde{\beta} = \underset{\beta}{\operatorname{argmin}} \, ||Y - X\beta||_2^2 + p_\lambda(\beta),$$

where $p_\lambda(\beta)$ is a *penalty* term - L2 in the case discussed above

L1 - Lasso sparse estimation

- The above penalized loss function can be generalized as

$$\tilde{\beta} = \underset{\beta}{\operatorname{argmin}} ||Y - X\beta||_2^2 + p_\lambda(\beta),$$

  where $p_\lambda(\beta)$ is a *penalty* term - L2 in the case discussed above
- Another very common penalty is L1 - penalizing the sum of absolute values $|\beta_j|$

L1 - Lasso sparse estimation

- The above penalized loss function can be generalized as

$$\tilde{\beta} = \underset{\beta}{\operatorname{argmin}} ||Y - X\beta||_2^2 + p_\lambda(\beta),$$

  where $p_\lambda(\beta)$ is a *penalty* term - L2 in the case discussed above
- Another very common penalty is L1 - penalizing the sum of absolute values $|\beta_j|$
- *Lasso* & the resulting estimator has a Bayesian interpretation under a double exponential (Laplace) prior

L1 - Lasso sparse estimation

- The above penalized loss function can be generalized as

$$\tilde{\beta} = \underset{\beta}{\operatorname{argmin}} ||Y - X\beta||_2^2 + p_\lambda(\beta),$$

  where $p_\lambda(\beta)$ is a *penalty* term - L2 in the case discussed above
- Another very common penalty is L1 - penalizing the sum of absolute values $|\beta_j|$
- *Lasso* & the resulting estimator has a Bayesian interpretation under a double exponential (Laplace) prior
- $\tilde{\beta}$ is sparse & contains exact zeros values

A proliferation of penalties & priors

- There is a HUGE literature proposing many different penalties

A proliferation of penalties & priors

- There is a HUGE literature proposing many different penalties
- Adaptive Lasso, fused Lasso, elastic net, etc etc

A proliferation of penalties & priors

- There is a HUGE literature proposing many different penalties
- Adaptive Lasso, fused Lasso, elastic net, etc etc
- In general, methods only produce a sparse point estimate & are dangerous scientifically

A proliferation of penalties & priors

- There is a HUGE literature proposing many different penalties
- Adaptive Lasso, fused Lasso, elastic net, etc etc
- In general, methods only produce a sparse point estimate & are dangerous scientifically
- Many errors in interpreting the zero vs non-zero elements

A proliferation of penalties & priors

- There is a HUGE literature proposing many different penalties
- Adaptive Lasso, fused Lasso, elastic net, etc etc
- In general, methods only produce a sparse point estimate & are dangerous scientifically
- Many errors in interpreting the zero vs non-zero elements
- Parallel Bayesian literature on shrinkage priors - horseshoe, generalized double Pareto, Dirichlet-Laplace, etc

Bayesian shrinkage priors

- Appropriate prior $\pi(\beta)$ for the high-dimensional vector of coefficients?

Bayesian shrinkage priors

- Appropriate prior $\pi(\beta)$ for the high-dimensional vector of coefficients?
- Most commonly local-global scale mixture of Gaussians,

$$\beta_j \overset{iid}{\sim} N(0, \psi_j \lambda), \quad \psi_j \sim f, \quad \lambda \sim g,$$

ψ_j=local scale, λ = global scale

Bayesian shrinkage priors

- Appropriate prior $\pi(\beta)$ for the high-dimensional vector of coefficients?
- Most commonly local-global scale mixture of Gaussians,

$$\beta_j \overset{iid}{\sim} N(0, \psi_j \lambda), \quad \psi_j \sim f, \quad \lambda \sim g,$$

  ψ_j=local scale, λ = global scale
- Choose $\lambda \approx 0$ & ψ_j to have many small values with some large

Bayesian shrinkage priors

- Appropriate prior $\pi(\beta)$ for the high-dimensional vector of coefficients?
- Most commonly local-global scale mixture of Gaussians,

$$\beta_j \overset{iid}{\sim} N(0, \psi_j \lambda), \quad \psi_j \sim f, \quad \lambda \sim g,$$

  ψ_j=local scale, λ = global scale
- Choose $\lambda \approx 0$ & ψ_j to have many small values with some large
- Different choices of f, g lead to different priors in the literature - Bayesian Lasso is a poor choice, as horseshoe, gDP, DL etc have much better theoretical & practical performance

Bayesian shrinkage priors

- Appropriate prior $\pi(\beta)$ for the high-dimensional vector of coefficients?
- Most commonly local-global scale mixture of Gaussians,

$$\beta_j \overset{iid}{\sim} N(0, \psi_j \lambda), \quad \psi_j \sim f, \quad \lambda \sim g,$$

  ψ_j=local scale, λ = global scale
- Choose $\lambda \approx 0$ & ψ_j to have many small values with some large
- Different choices of f, g lead to different priors in the literature - Bayesian Lasso is a poor choice, as horseshoe, gDP, DL etc have much better theoretical & practical performance
- Literature on scalable computation using MCMC - e.g, Johndrow et al arXiv:1705.00841

Bayesian shrinkage priors

- Appropriate prior $\pi(\beta)$ for the high-dimensional vector of coefficients?
- Most commonly local-global scale mixture of Gaussians,

$$\beta_j \overset{iid}{\sim} N(0, \psi_j \lambda), \quad \psi_j \sim f, \quad \lambda \sim g,$$

  ψ_j=local scale, λ = global scale
- Choose $\lambda \approx 0$ & ψ_j to have many small values with some large
- Different choices of f, g lead to different priors in the literature - Bayesian Lasso is a poor choice, as horseshoe, gDP, DL etc have much better theoretical & practical performance
- Literature on scalable computation using MCMC - e.g, Johndrow et al arXiv:1705.00841
- Datta & Dunson (20)16, *Biometrika*) - develop such approaches for huge dimensional sparse count data arising in genomics

Features of a Bayesian approach

- Bayesian approach provides a full posterior $\pi(\beta|Y,X)$ characterizing uncertainty instead of just a sparse point estimate $\hat{\beta}$

Features of a Bayesian approach

- Bayesian approach provides a full posterior $\pi(\beta|Y,X)$ characterizing uncertainty instead of just a sparse point estimate $\hat{\beta}$
- By using MCMC, we can easily get credible bands (Bayesian confidence intervals) for not only the β_j's but also for any functional of interest

Features of a Bayesian approach

- Bayesian approach provides a full posterior $\pi(\beta|Y,X)$ characterizing uncertainty instead of just a sparse point estimate $\hat{\beta}$
- By using MCMC, we can easily get credible bands (Bayesian confidence intervals) for not only the β_j's but also for any functional of interest
- Relatively straightforward to incorporate extensions to allow hierarchical dependence structures, multivariate responses, missing data, etc

Features of a Bayesian approach

- Bayesian approach provides a full posterior $\pi(\beta|Y,X)$ characterizing uncertainty instead of just a sparse point estimate $\hat{\beta}$
- By using MCMC, we can easily get credible bands (Bayesian confidence intervals) for not only the β_j's but also for any functional of interest
- Relatively straightforward to incorporate extensions to allow hierarchical dependence structures, multivariate responses, missing data, etc
- However, there is a need for approaches that are

Features of a Bayesian approach

- Bayesian approach provides a full posterior $\pi(\beta|Y,X)$ characterizing uncertainty instead of just a sparse point estimate $\hat{\beta}$
- By using MCMC, we can easily get credible bands (Bayesian confidence intervals) for not only the β_j's but also for any functional of interest
- Relatively straightforward to incorporate extensions to allow hierarchical dependence structures, multivariate responses, missing data, etc
- However, there is a need for approaches that are
  1. More robust to parametric assumptions,

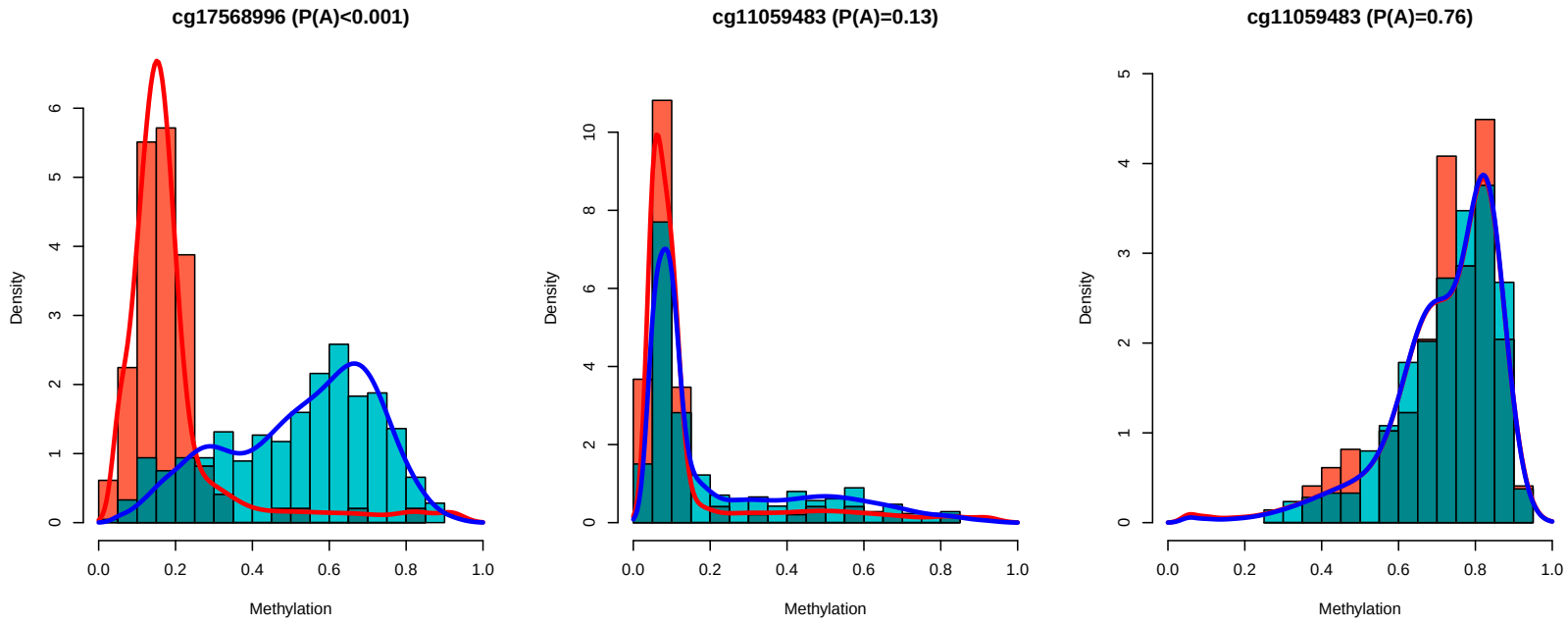
Features of a Bayesian approach

- Bayesian approach provides a full posterior $\pi(\beta|Y,X)$ characterizing uncertainty instead of just a sparse point estimate $\hat{\beta}$
- By using MCMC, we can easily get credible bands (Bayesian confidence intervals) for not only the β_j's but also for any functional of interest
- Relatively straightforward to incorporate extensions to allow hierarchical dependence structures, multivariate responses, missing data, etc
- However, there is a need for approaches that are
  1. More robust to parametric assumptions,
  2. easily computationally scalable to huge datasets

Features of a Bayesian approach

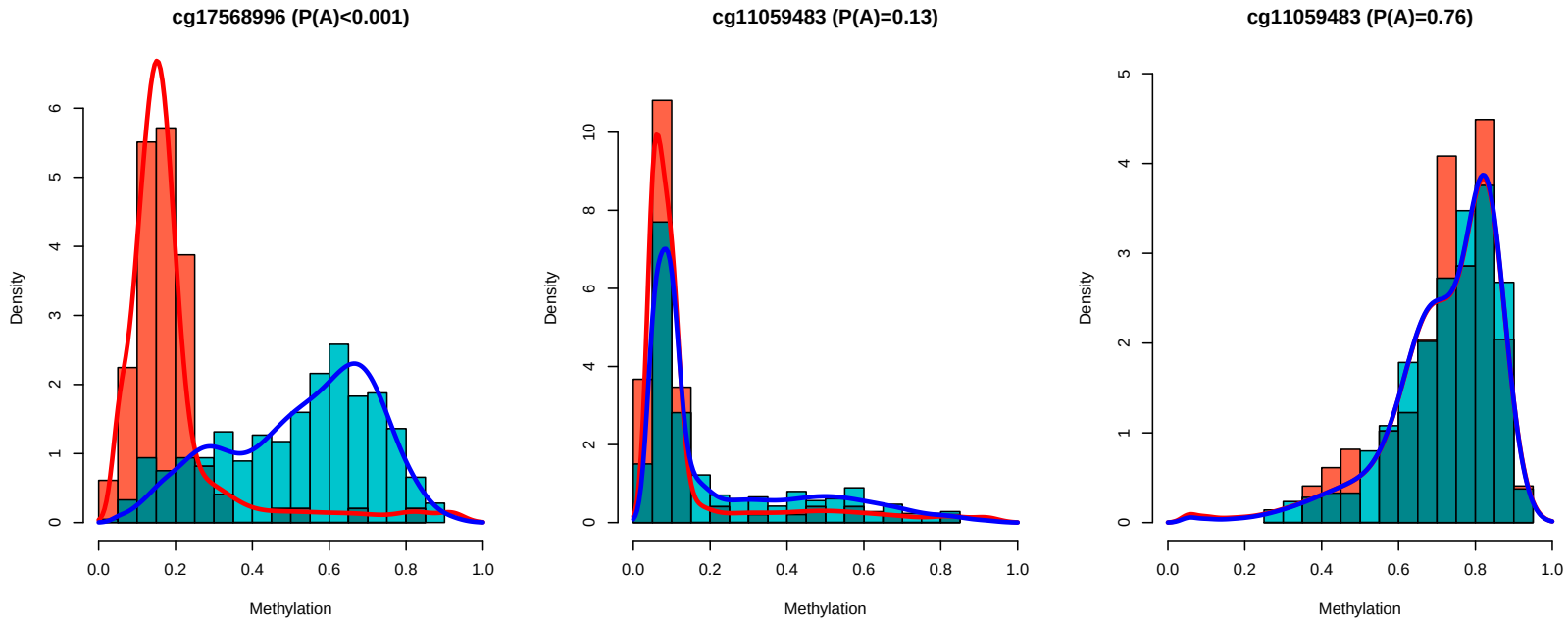
- Bayesian approach provides a full posterior $\pi(\beta|Y,X)$ characterizing uncertainty instead of just a sparse point estimate $\hat{\beta}$
- By using MCMC, we can easily get credible bands (Bayesian confidence intervals) for not only the β_j's but also for any functional of interest
- Relatively straightforward to incorporate extensions to allow hierarchical dependence structures, multivariate responses, missing data, etc
- However, there is a need for approaches that are
  1. More robust to parametric assumptions,
  2. easily computationally scalable to huge datasets
  3. provide a way to deal with intractable $p \gg n$ problems

Application 1: DNA methylation arrays



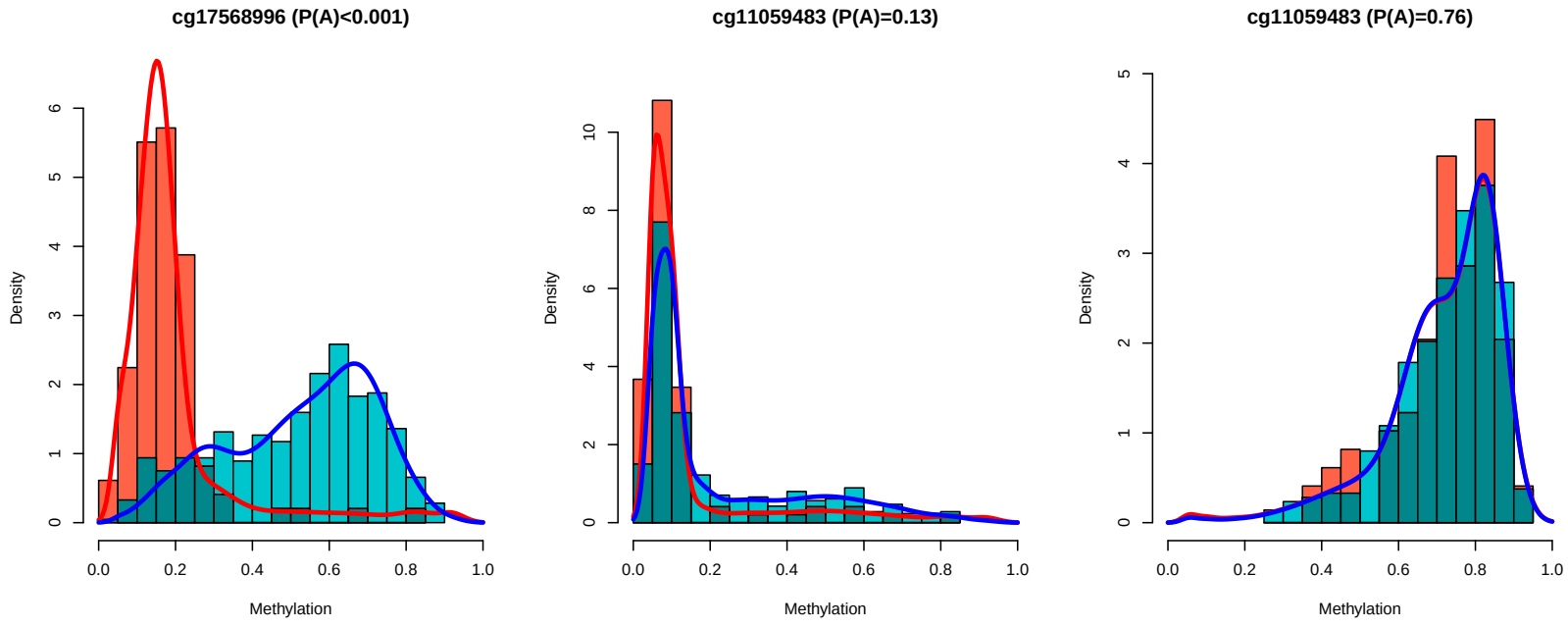
- Focus: screening for differentially methylated CpG sites

Application 1: DNA methylation arrays



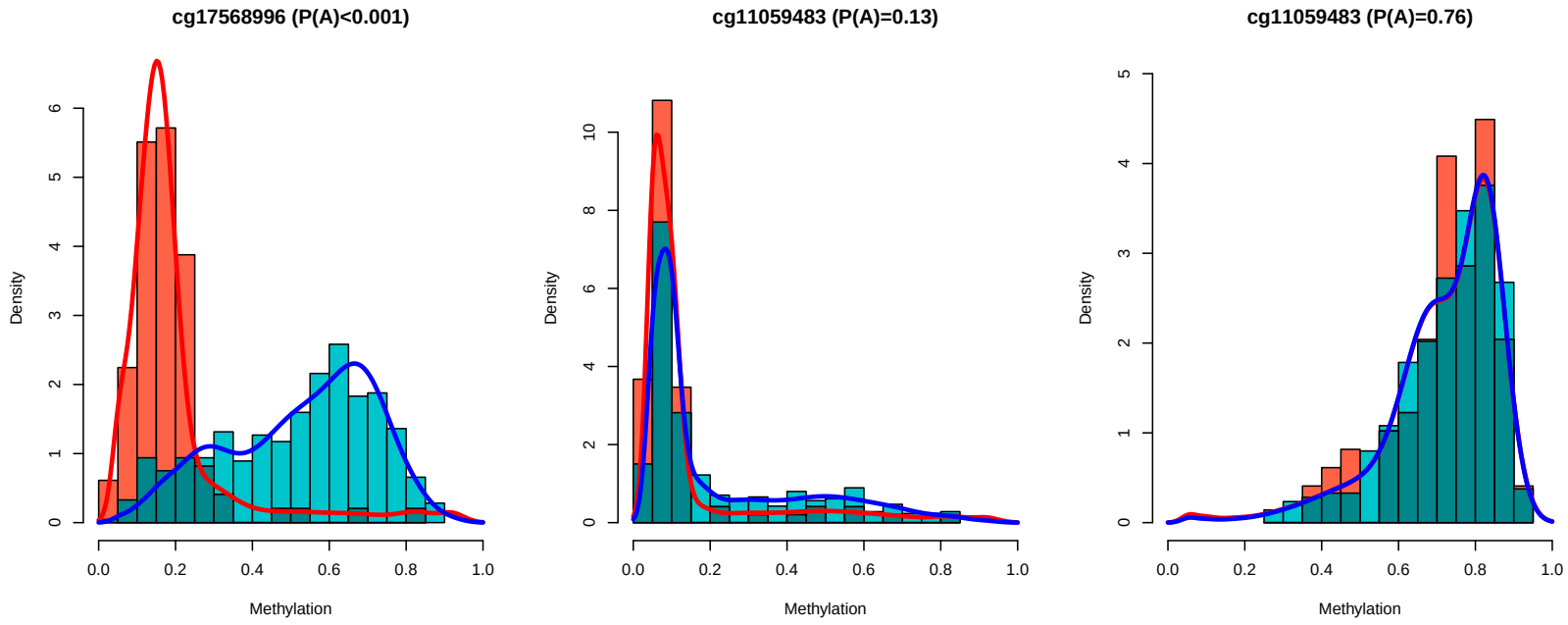
- Focus: screening for differentially methylated CpG sites
- High-throughput arrays are routinely used - eg., Illumina Human Methylation450 Beadchip

Application 1: DNA methylation arrays



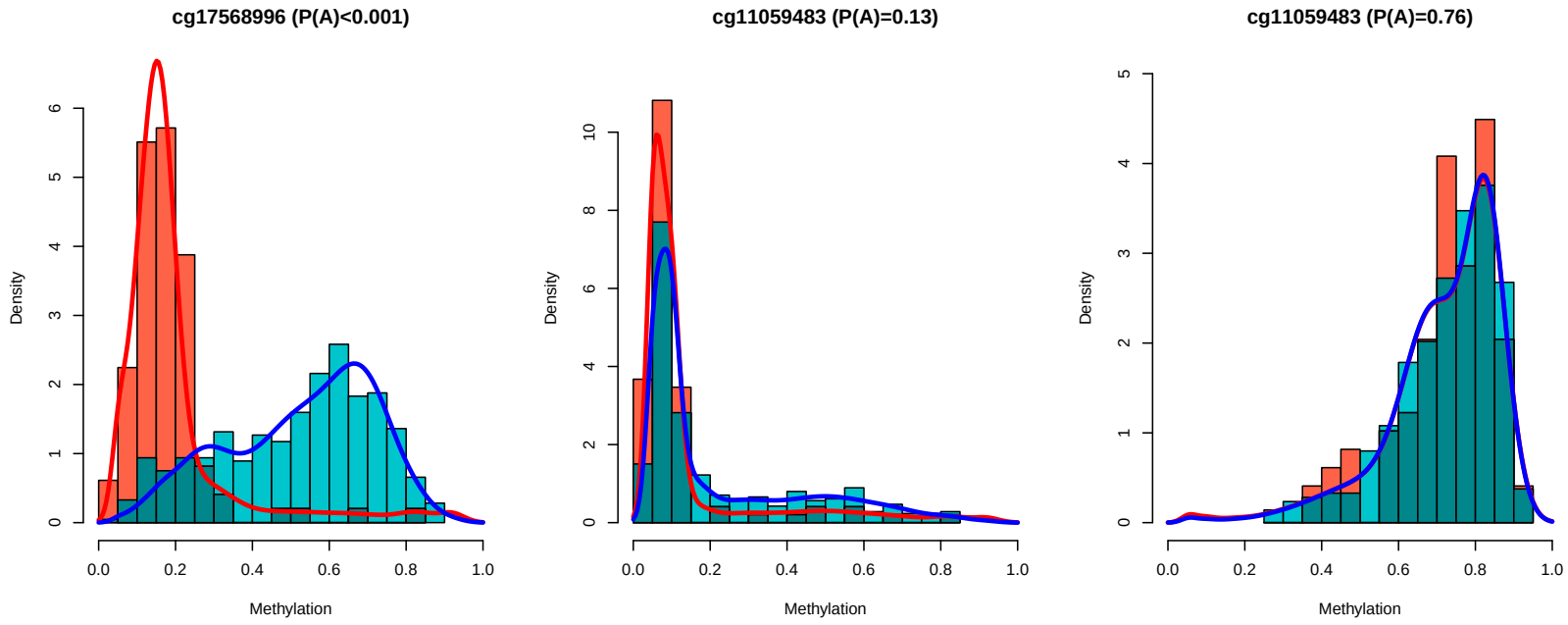
- Focus: screening for differentially methylated CpG sites
- High-throughput arrays are routinely used - eg., Illumina Human Methylation450 Beadchip
- Measurements in [0,1] interval, ranging from no methylation to fully methylated

Application 1: DNA methylation arrays



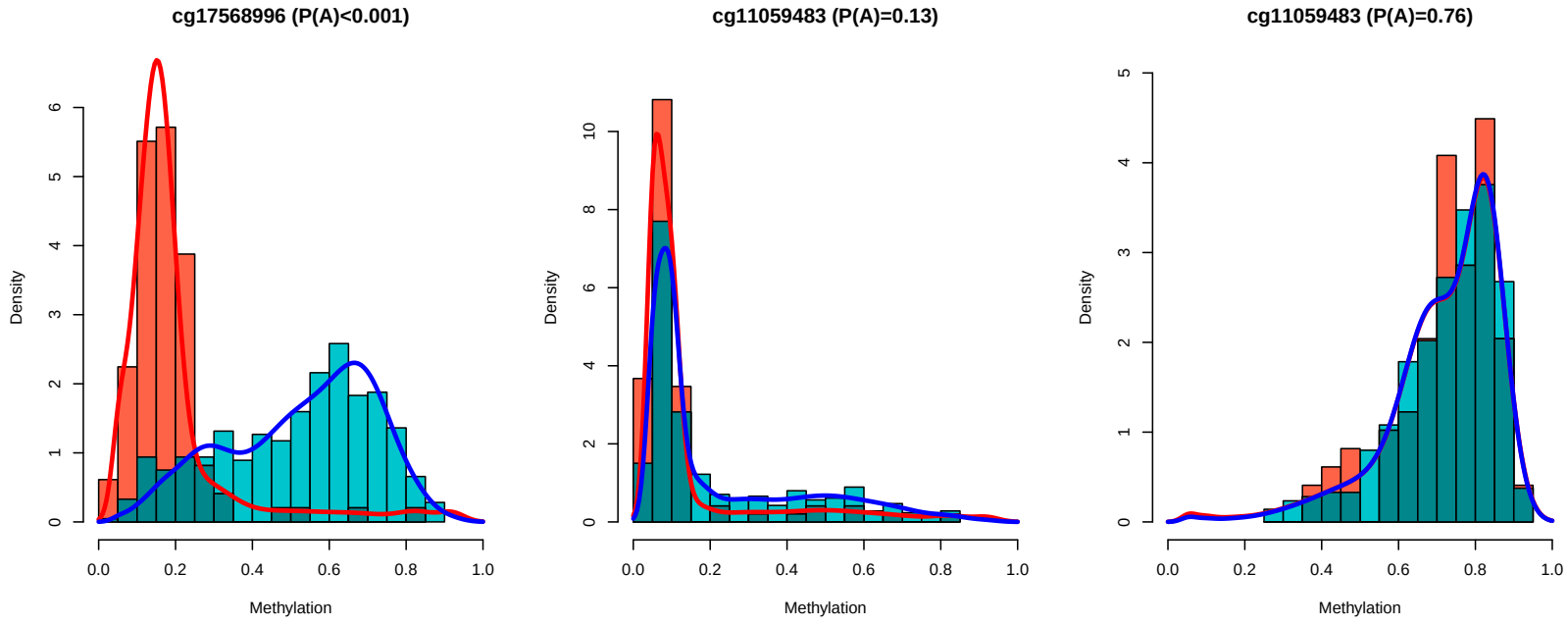
- Focus: screening for differentially methylated CpG sites
- High-throughput arrays are routinely used - eg., Illumina Human Methylation450 Beadchip
- Measurements in [0,1] interval, ranging from no methylation to fully methylated
- Representative data from the Cancer Genome Atlas

Application 1: DNA methylation arrays



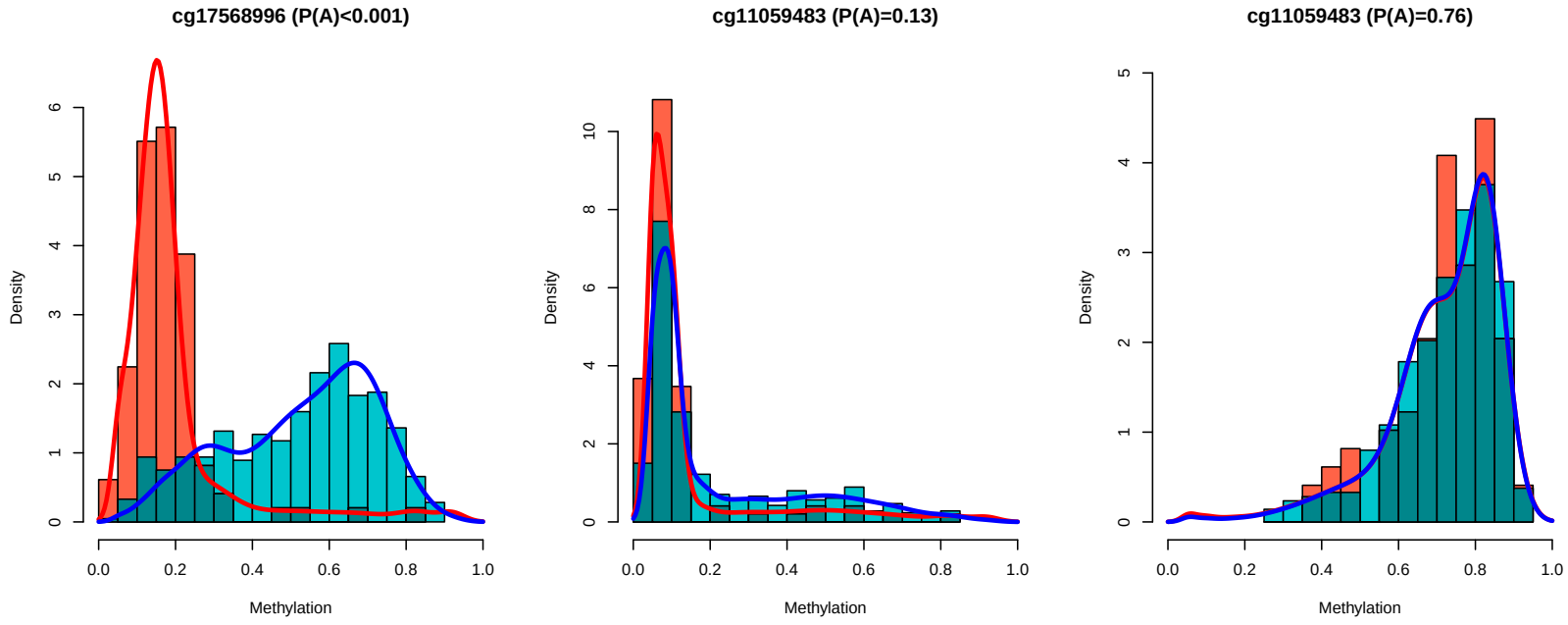
- Focus: screening for differentially methylated CpG sites
- High-throughput arrays are routinely used - eg., Illumina Human Methylation450 Beadchip
- Measurements in [0,1] interval, ranging from no methylation to fully methylated
- Representative data from the Cancer Genome Atlas
- Clearly distributions exhibit multimodality & skewness

Comments



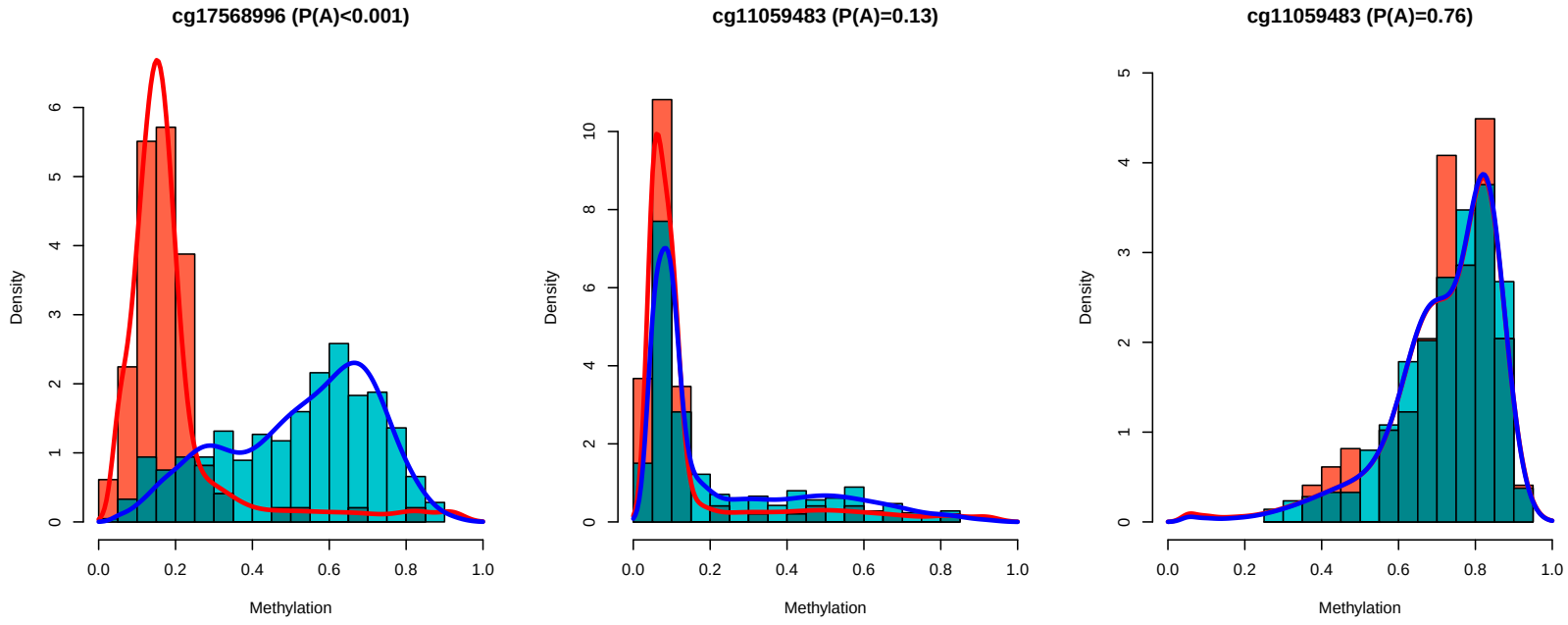
- We observe data like this at a HUGE number of CpG sites

Comments



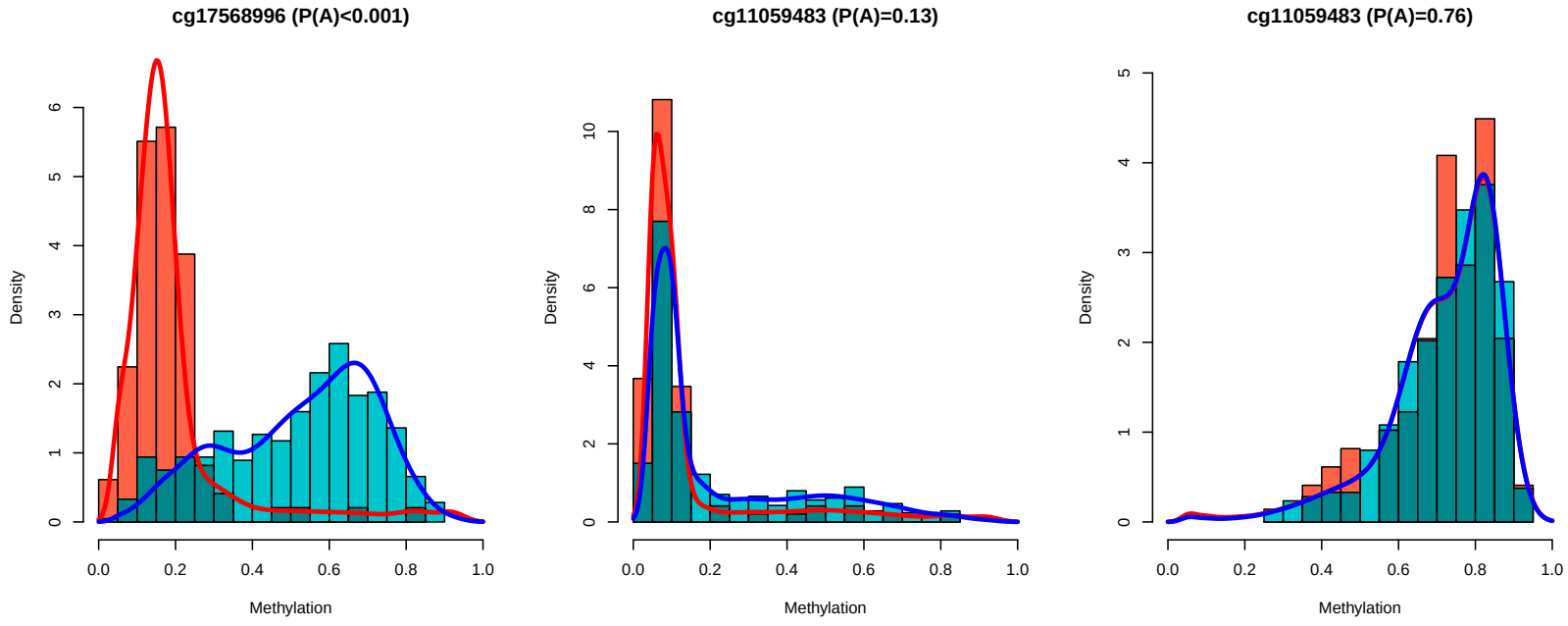
- We observe data like this at a HUGE number of CpG sites
- Many distributions share common attributes - modes etc

Comments



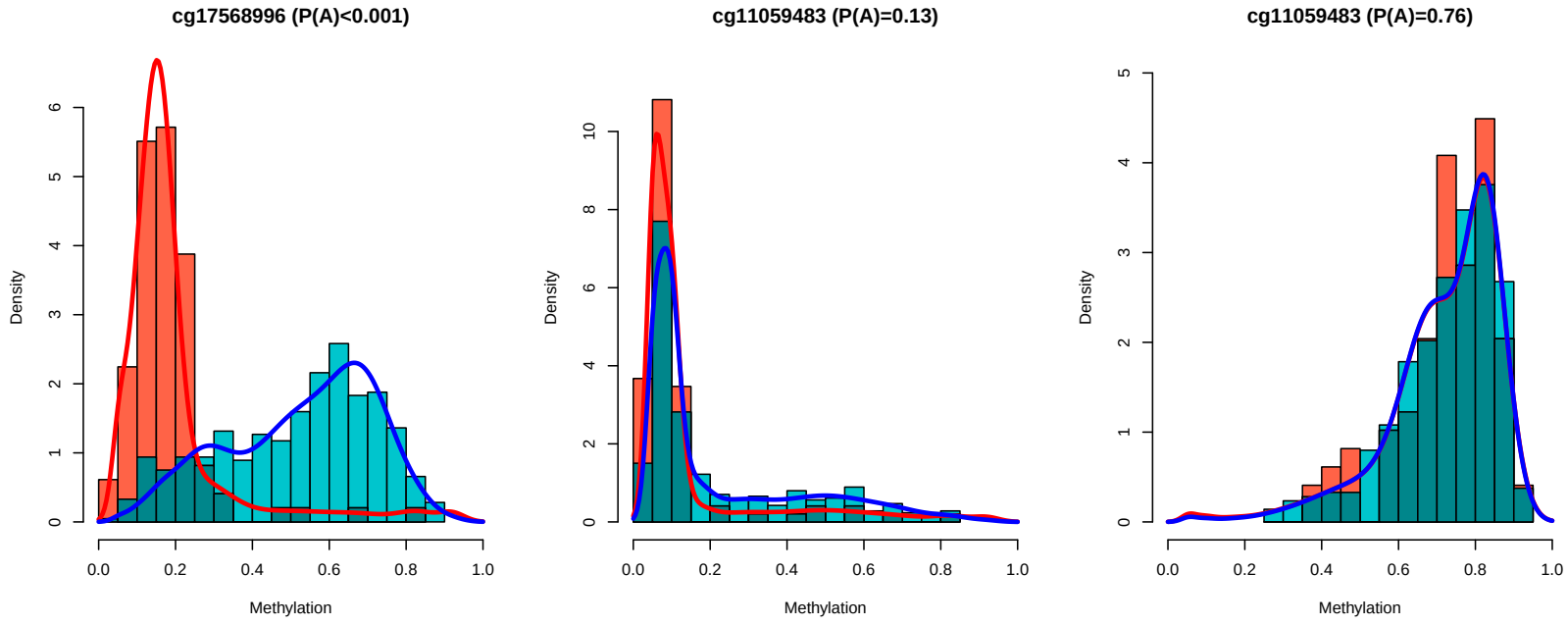
- We observe data like this at a HUGE number of CpG sites
- Many distributions share common attributes - modes etc
- Can accurately characterize the methylation densities using a kernel mixture model

Comments



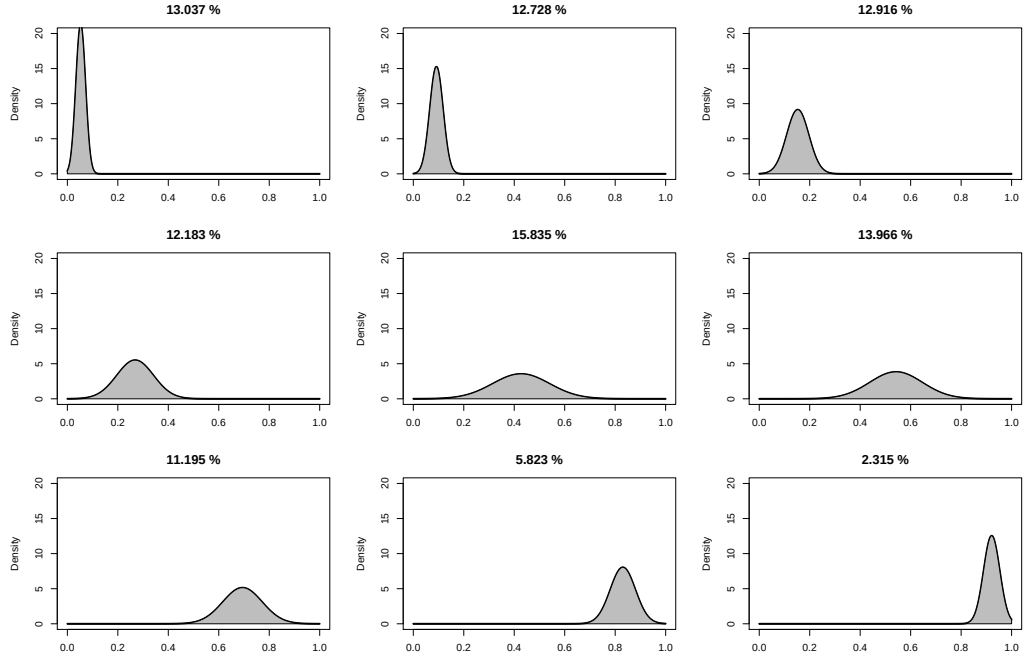
- We observe data like this at a HUGE number of CpG sites
- Many distributions share common attributes - modes etc
- Can accurately characterize the methylation densities using a kernel mixture model
- Key idea: use the same kernels across the sites & groups but allow the weights to vary

Comments



- We observe data like this at a HUGE number of CpG sites
- Many distributions share common attributes - modes etc
- Can accurately characterize the methylation densities using a kernel mixture model
- Key idea: use the same kernels across the sites & groups but allow the weights to vary
- SHARed Kernel (SHARK) method (*Lock & Dunson, 2015*)

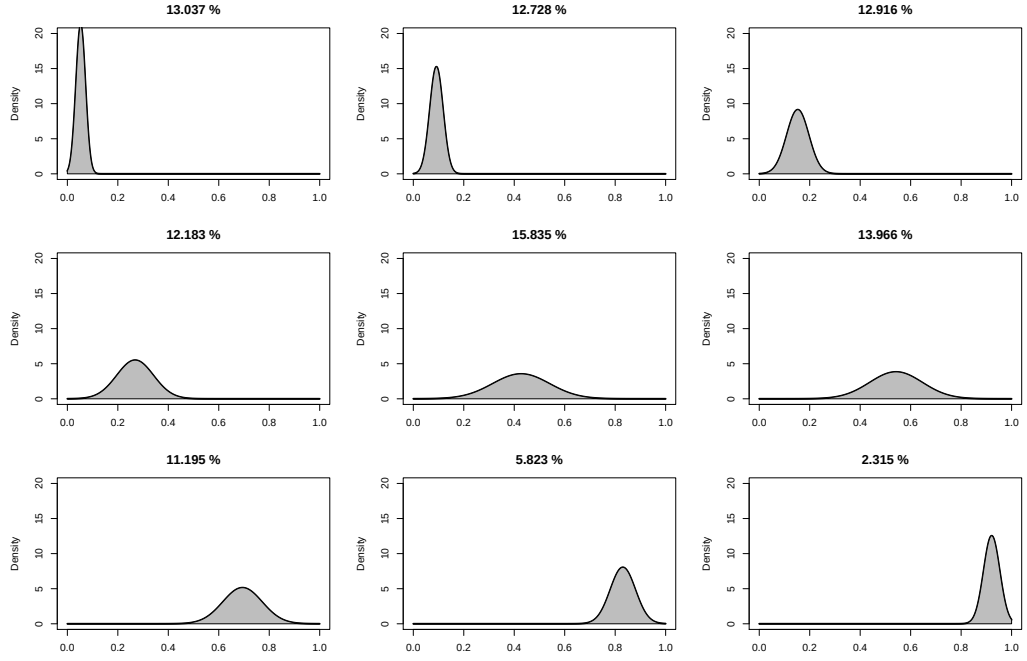
Shark - some details



- The methylation density at site j in group g is f_{jg}:

$$f_{jg}(y) = \sum_{h=1}^{k} \pi_{jgh} \mathcal{K}(y; \theta_h)$$

Shark - some details

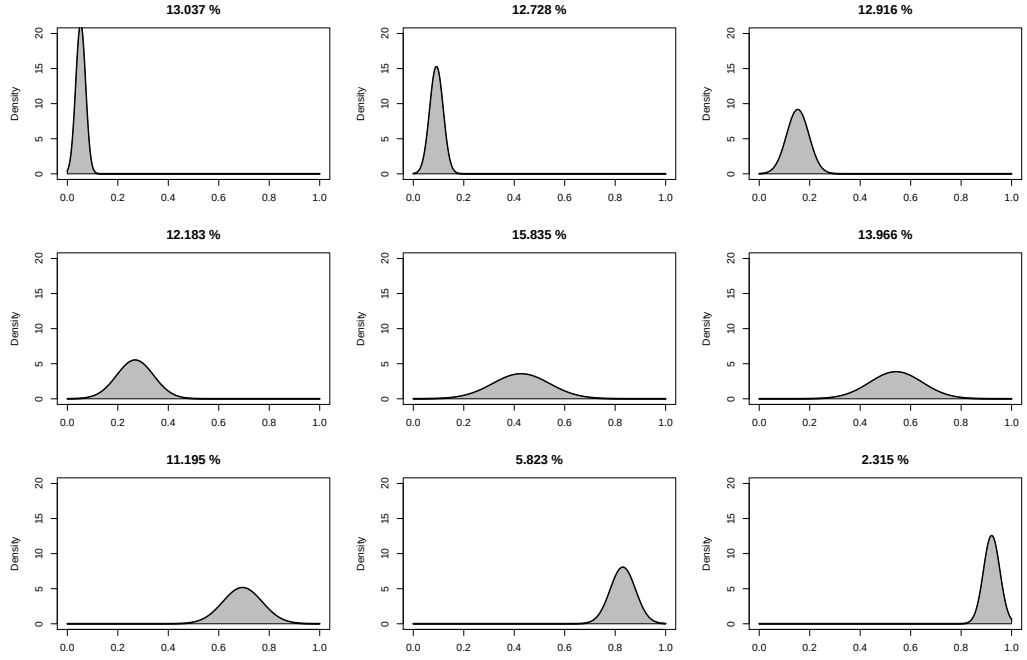


- The methylation density at site j in group g is f_{jg}:

$$f_{jg}(y) = \sum_{h=1}^{k} \pi_{jgh} \mathcal{K}(y; \theta_h)$$

- $\pi_{jg} = (\pi_{jg1}, \ldots, \pi_{jgk})'$ are weights specific to j, g

Shark - some details

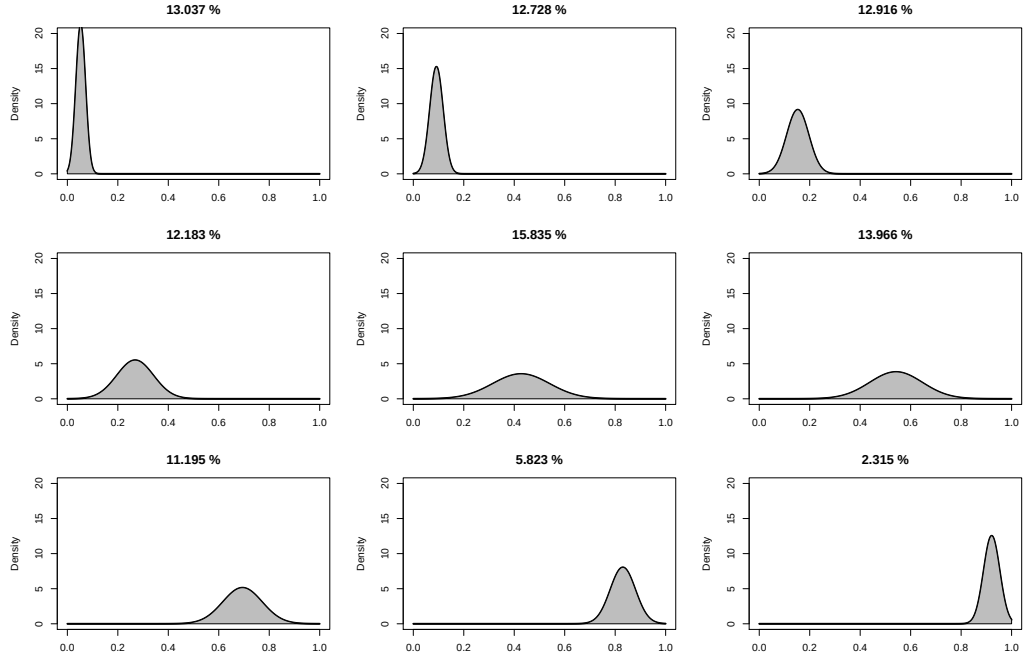


- The methylation density at site j in group g is f_{jg}:

$$f_{jg}(y) = \sum_{h=1}^{k} \pi_{jgh} \mathcal{K}(y;\theta_h)$$

- $\pi_{jg} = (\pi_{jg1}, \ldots, \pi_{jgk})'$ are weights specific to j, g
- $\mathcal{K}(y;\theta_h)$ is a *shared* kernel (truncated normal in this case)

Shark - some details

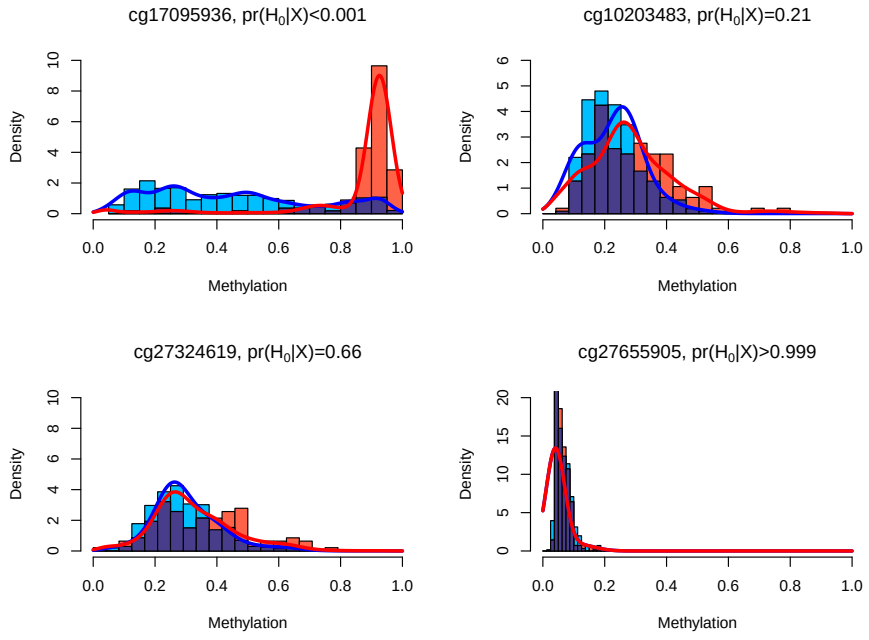


- The methylation density at site j in group g is f_{jg}:

$$f_{jg}(y) = \sum_{h=1}^{k} \pi_{jgh} \mathcal{K}(y;\theta_h)$$

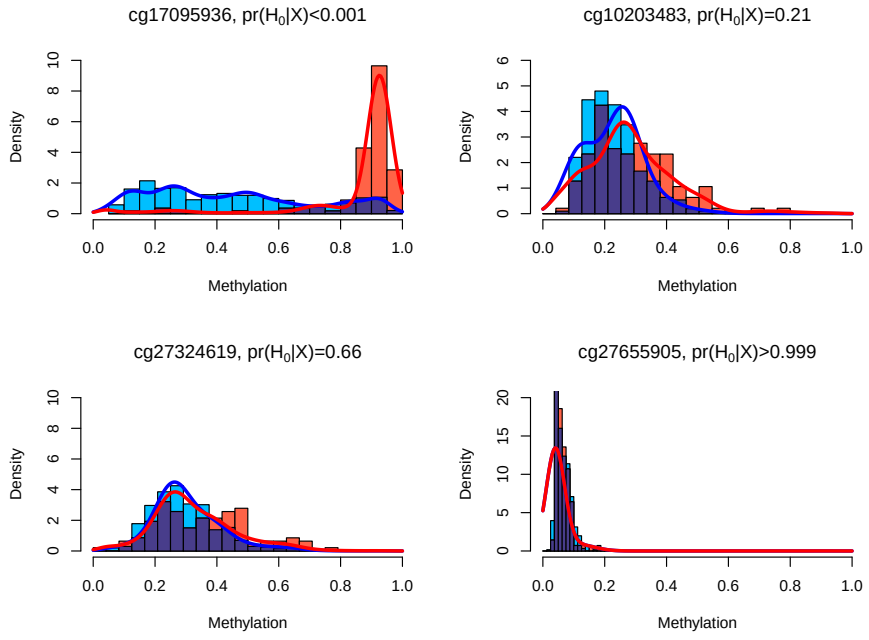
- $\pi_{jg} = (\pi_{jg1}, \ldots, \pi_{jgk})'$ are weights specific to j, g
- $\mathcal{K}(y;\theta_h)$ is a *shared* kernel (truncated normal in this case)
- We estimate the above kernels in a first stage relying on a subsample of 500 sites - only need 9 kernels

Shark - implementation (continued)



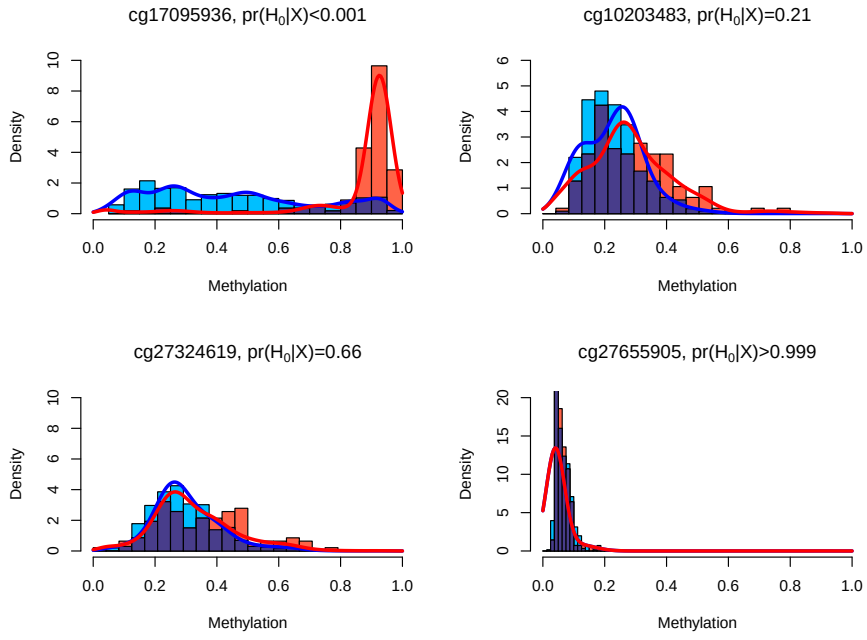
- We put a simple hierarchical prior on π_{jg} - Dirichlet in each group

Shark - implementation (continued)



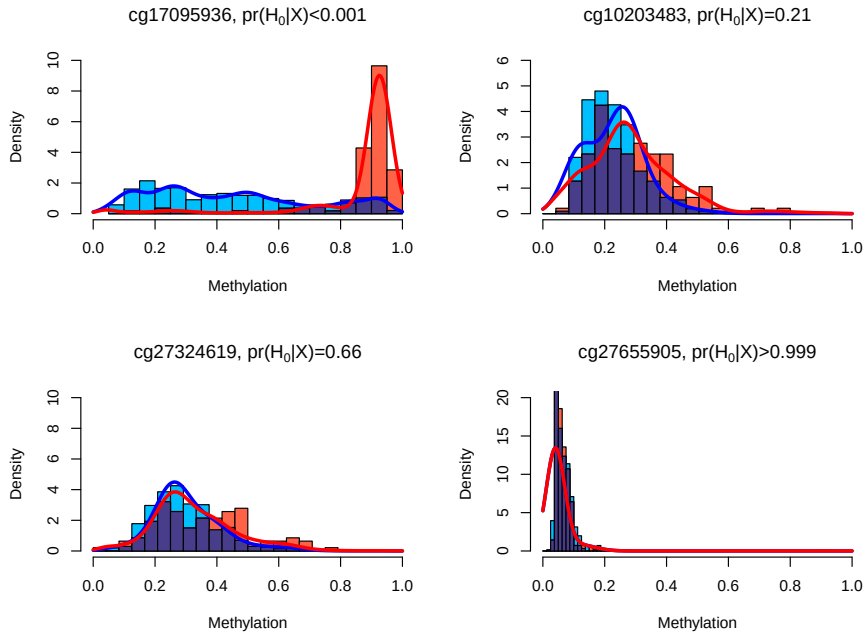
- We put a simple hierarchical prior on π_{jg} - Dirichlet in each group
- Prior probability random CpG site is differentially methylated given a beta hyperprior

Shark - implementation (continued)



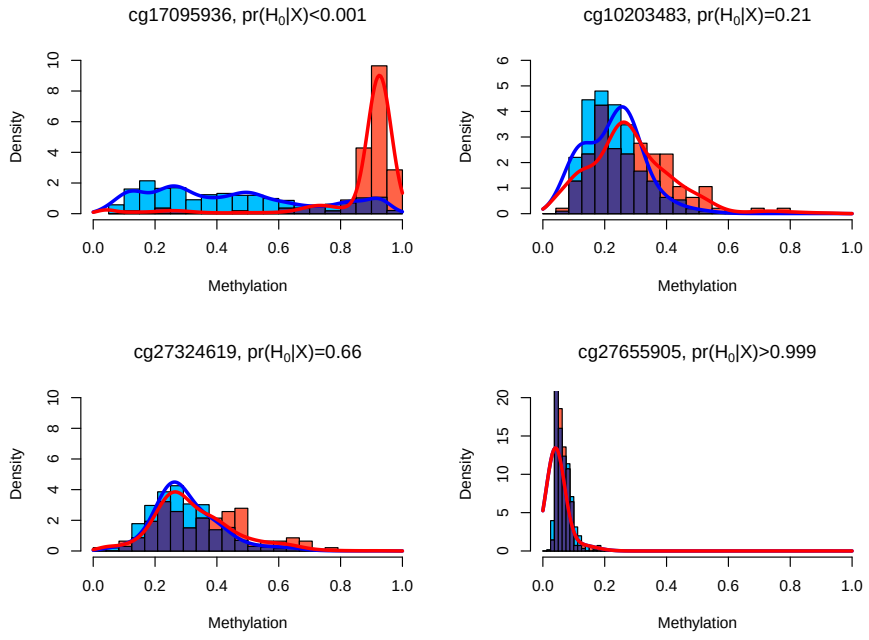
- We put a simple hierarchical prior on π_{jg} - Dirichlet in each group
- Prior probability random CpG site is differentially methylated given a beta hyperprior
- Automatically adjusts for multiple testing error, controlling FDR

Shark - implementation (continued)



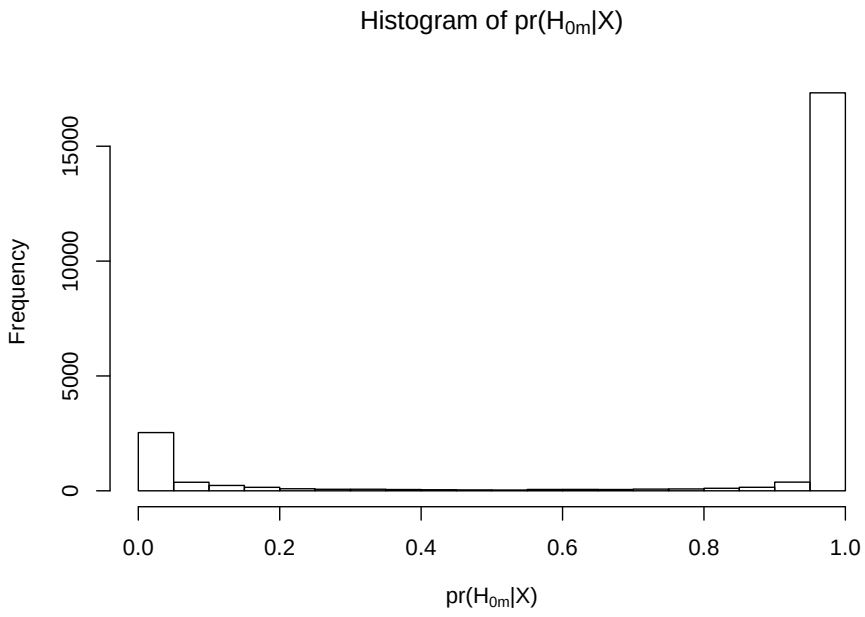
- We put a simple hierarchical prior on π_{jg} - Dirichlet in each group
- Prior probability random CpG site is differentially methylated given a beta hyperprior
- Automatically adjusts for multiple testing error, controlling FDR
- Computation - very fast Gibbs & parallelizable Gibbs sampler

Shark - implementation (continued)



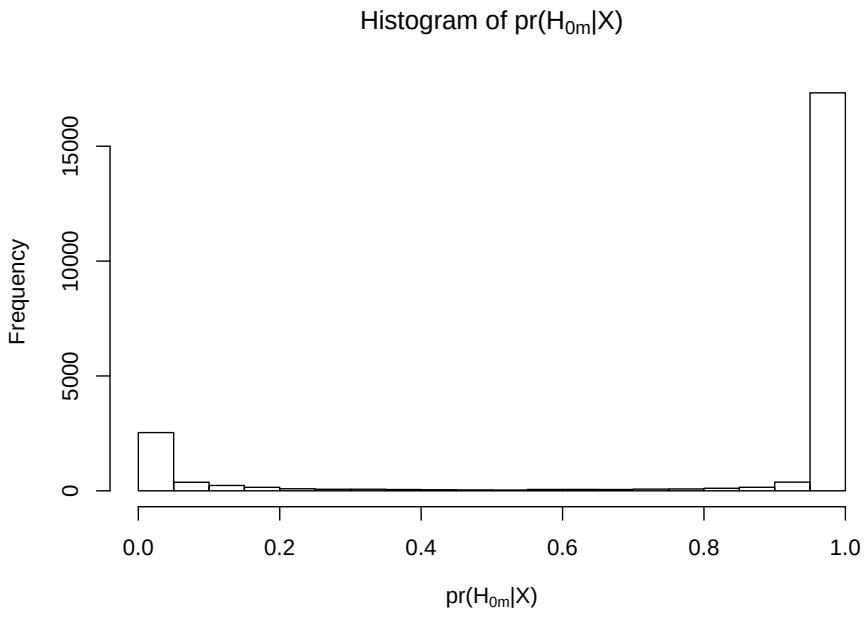
- We put a simple hierarchical prior on π_{jg} - Dirichlet in each group
- Prior probability random CpG site is differentially methylated given a beta hyperprior
- Automatically adjusts for multiple testing error, controlling FDR
- Computation - very fast Gibbs & parallelizable Gibbs sampler
- Theory support, including under misspecification

Results for Cancer Genome Atlas Data



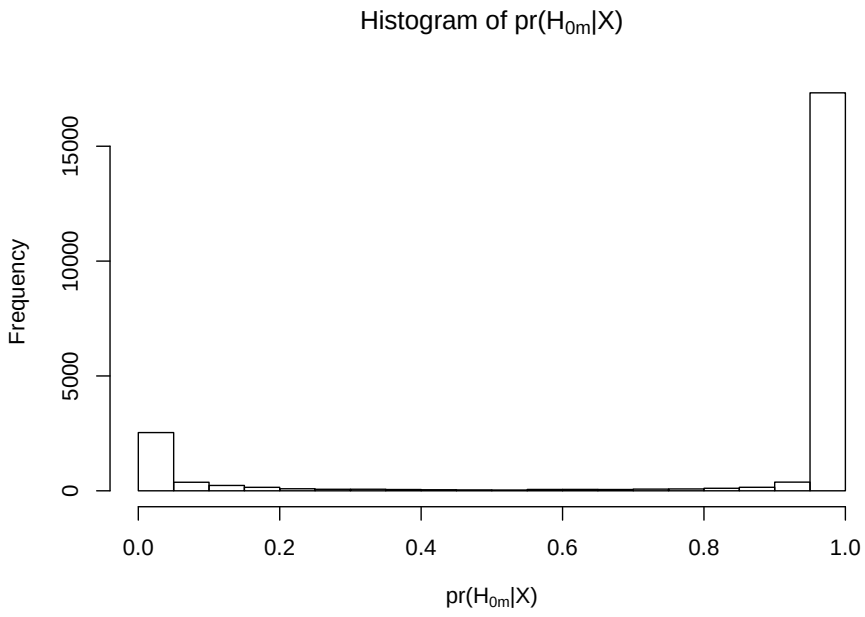
- Illustrate using $n = 597$ breast cancer samples & 21,986 CpG sites from TGGA

Results for Cancer Genome Atlas Data



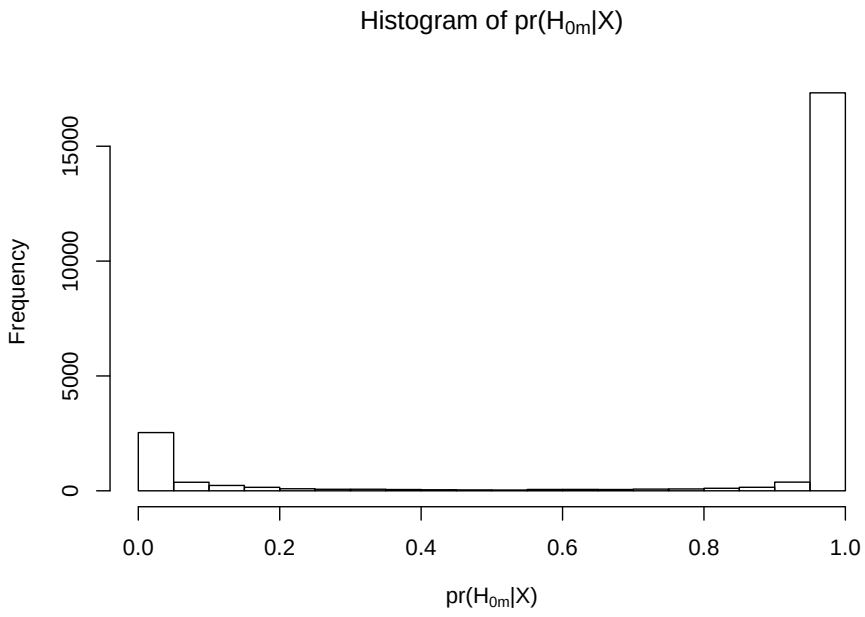
- Illustrate using $n = 597$ breast cancer samples & 21,986 CpG sites from TGGA
- Focus on testing difference between basal-like (112) and not (485) at each site

Results for Cancer Genome Atlas Data



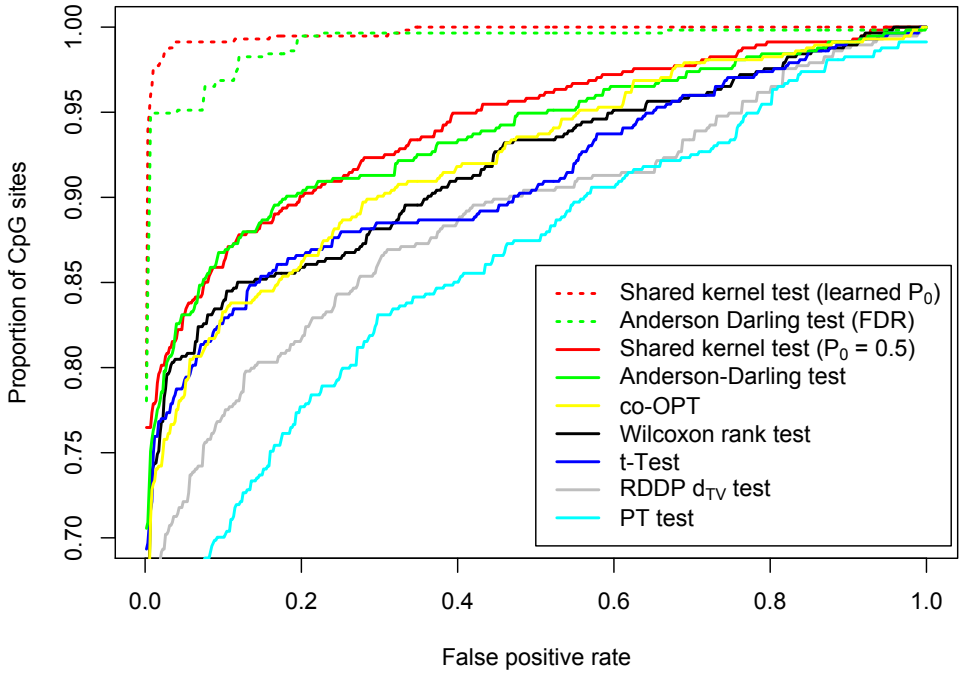
- Illustrate using $n = 597$ breast cancer samples & 21,986 CpG sites from TGGA
- Focus on testing difference between basal-like (112) and not (485) at each site
- Global proportion of no difference was 0.821

Results for Cancer Genome Atlas Data



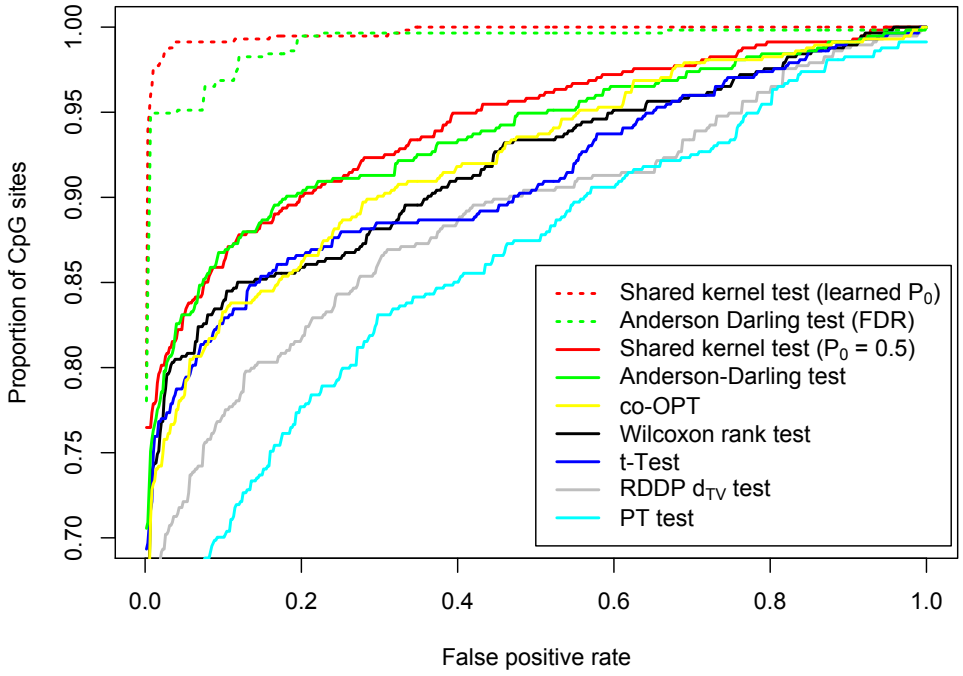
- Illustrate using $n = 597$ breast cancer samples & 21,986 CpG sites from TGGA
- Focus on testing difference between basal-like (112) and not (485) at each site
- Global proportion of no difference was 0.821
- Distribution of posterior probabilities of H_{0m} shown above

Discussion & Comparisons



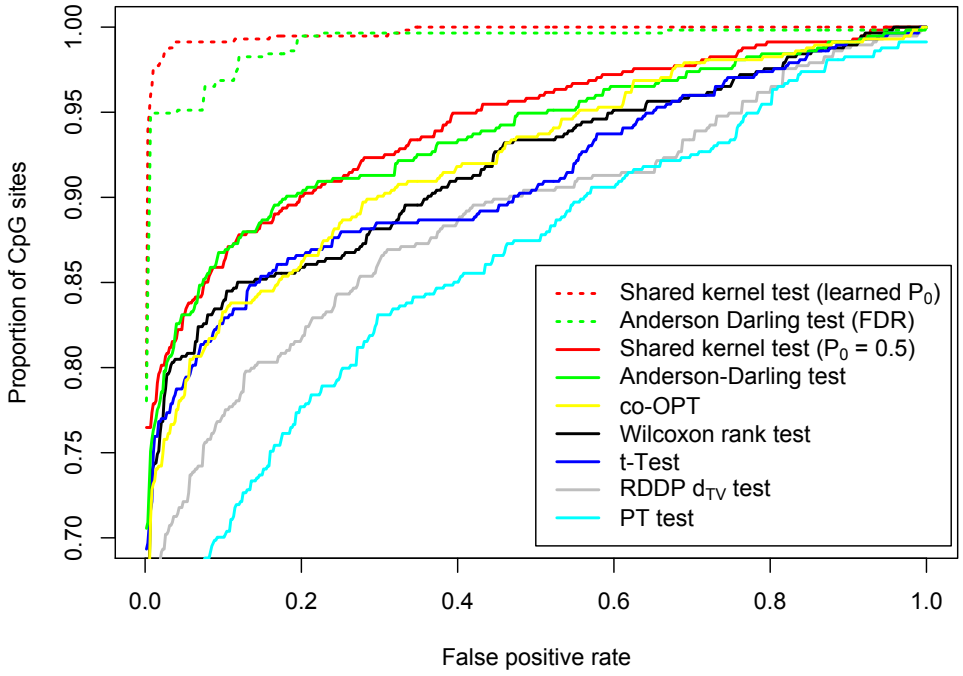
- Of 2117 CpG sites with pr$(H_{0m}) < 0.01$, 1256 have a significant negative association with gene expression ($p < 0.01$ spearman's rank correlation)

Discussion & Comparisons



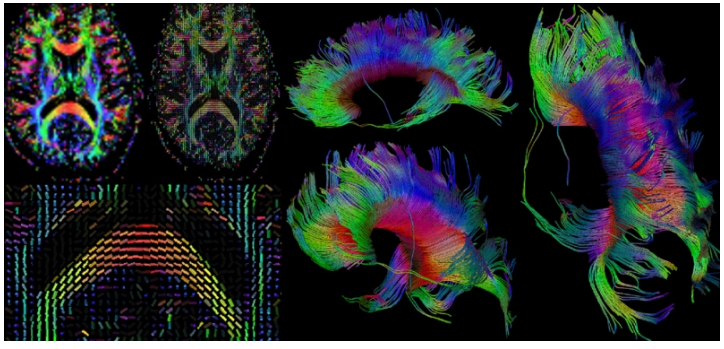
- Of 2117 CpG sites with pr$(H_{0m}) < 0.01$, 1256 have a significant negative association with gene expression ($p < 0.01$ spearman's rank correlation)
- Methylation gives potential mechanistic explanation for differences in gene transcription levels

Discussion & Comparisons



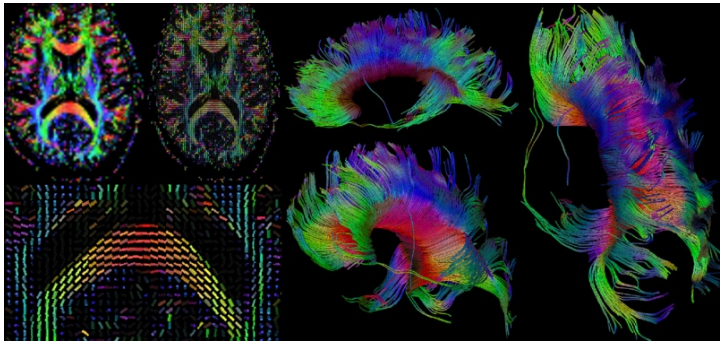
- Of 2117 CpG sites with pr$(H_{0m}) < 0.01$, 1256 have a significant negative association with gene expression ($p < 0.01$ spearman's rank correlation)
- Methylation gives potential mechanistic explanation for differences in gene transcription levels
- We compared power of our approach with alternatives

Shared kernel testing for complex phenotypes



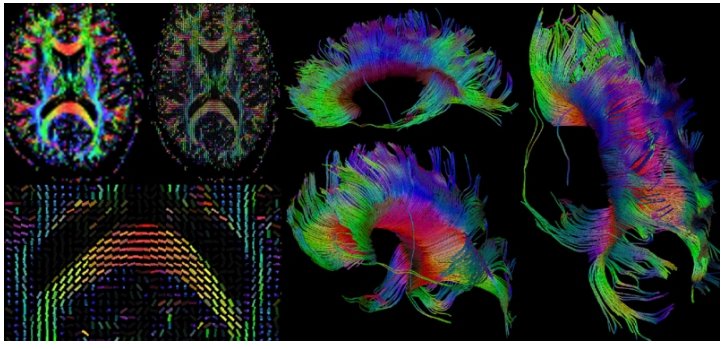
- Shared kernel approach can be applied to very complex phenotypes

Shared kernel testing for complex phenotypes



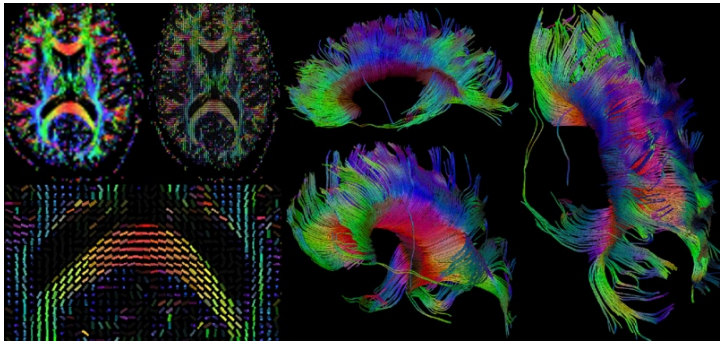
- Shared kernel approach can be applied to very complex phenotypes
- As long as a mixture model can be defined for the phenotype distribution under one condition

Shared kernel testing for complex phenotypes



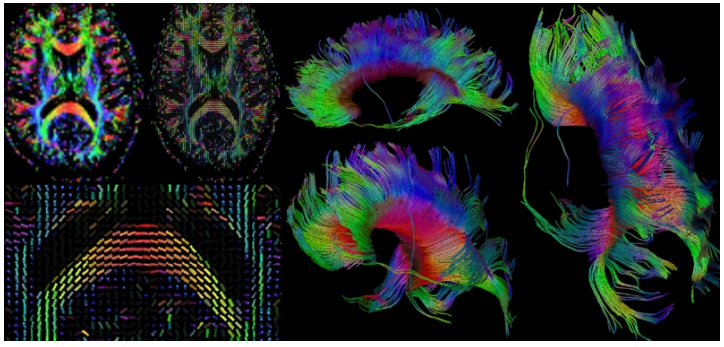
- Shared kernel approach can be applied to very complex phenotypes
- As long as a mixture model can be defined for the phenotype distribution under one condition
- I'll illustrate briefly using brain connectome phenotypes

Shared kernel testing for complex phenotypes



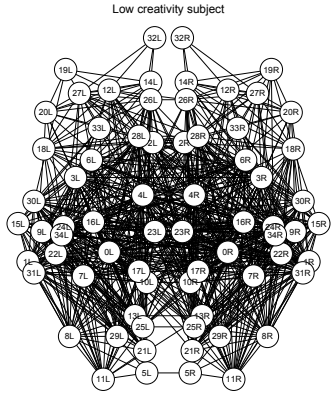
- Shared kernel approach can be applied to very complex phenotypes
- As long as a mixture model can be defined for the phenotype distribution under one condition
- I'll illustrate briefly using brain connectome phenotypes
- For each individual i, we extract a structural connectome X_i from MRI data

Shared kernel testing for complex phenotypes



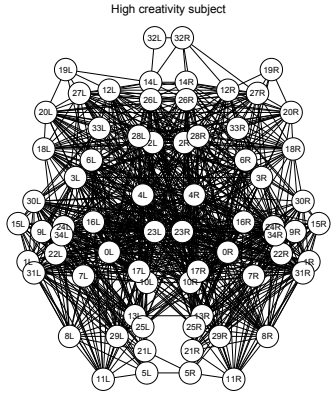
- Shared kernel approach can be applied to very complex phenotypes
- As long as a mixture model can be defined for the phenotype distribution under one condition
- I'll illustrate briefly using brain connectome phenotypes
- For each individual i, we extract a structural connectome X_i from MRI data
- Then, $X_{i[u,v]} = 1$ if there is any connection between regions u & v for individual i, and $X_{i[u,v]} = 0$ otherwise

A nonparametric model of variation in brain networks



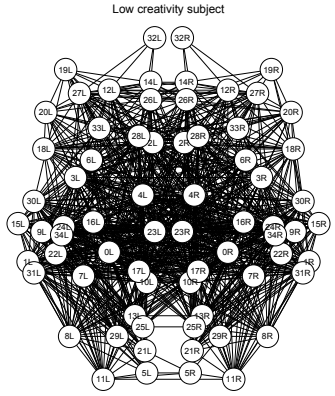
- Kernel for characterizing variation in brain network data across individuals: $X_i \sim P$, $P = ?$.

A nonparametric model of variation in brain networks



- Kernel for characterizing variation in brain network data across individuals: $X_i \sim P$, $P = ?$.
- For each brain region (r) & component (h), assign an individual-specific score $\eta_{ih[r]}$

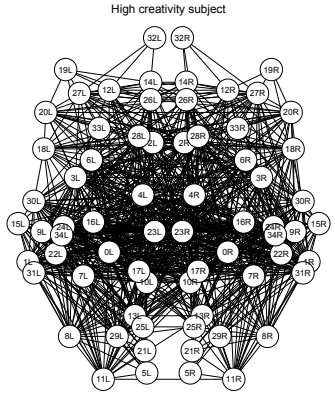
A nonparametric model of variation in brain networks



- Kernel for characterizing variation in brain network data across individuals: $X_i \sim P$, $P = ?$.
- For each brain region (r) & component (h), assign an individual-specific score $\eta_{ih[r]}$
- Characterize variation among individuals with:

$$\text{logit}\{\text{pr}(X_{i[u,v]} = 1)\} = \mu_{[u,v]} + \sum_{h=1}^{K} \lambda_{ih}\eta_{ih[u]}\eta_{ih[v]}, \quad \theta_i = \{\lambda_{ih}, \eta_{ir}\} \sim Q.$$

A nonparametric model of variation in brain networks

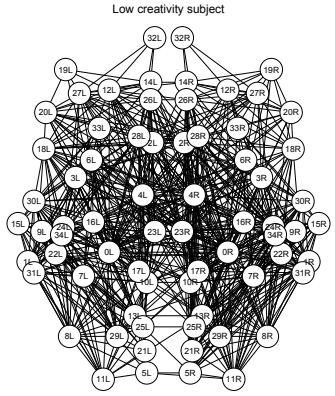


- Kernel for characterizing variation in brain network data across individuals: $X_i \sim P$, $P = ?$.
- For each brain region (r) & component (h), assign an individual-specific score $\eta_{ih[r]}$
- Characterize variation among individuals with:

$$\text{logit}\{\text{pr}(X_{i[u,v]} = 1)\} = \mu_{[u,v]} + \sum_{h=1}^{K} \lambda_{ih}\eta_{ih[u]}\eta_{ih[v]}, \quad \theta_i = \{\lambda_{ih}, \eta_{ir}\} \sim Q.$$

- Using Bayesian nonparametrics, allow Q (& P) to be unknown

A nonparametric model of variation in brain networks

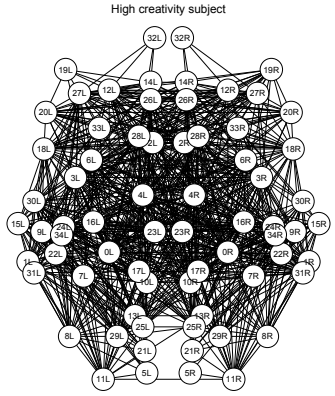


- Kernel for characterizing variation in brain network data across individuals: $X_i \sim P$, $P = ?$.
- For each brain region (r) & component (h), assign an individual-specific score $\eta_{ih[r]}$
- Characterize variation among individuals with:

$$\text{logit}\{\text{pr}(X_{i[u,v]} = 1)\} = \mu_{[u,v]} + \sum_{h=1}^{K} \lambda_{ih}\eta_{ih[u]}\eta_{ih[v]}, \quad \theta_i = \{\lambda_{ih}, \eta_{ir}\} \sim Q.$$

- Using Bayesian nonparametrics, allow Q (& P) to be unknown

A nonparametric model of variation in brain networks

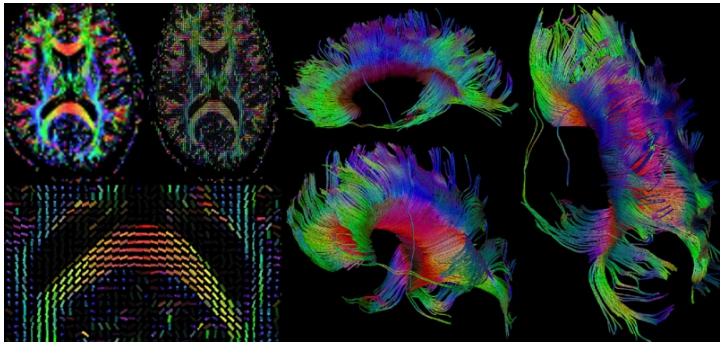


- Kernel for characterizing variation in brain network data across individuals: $X_i \sim P$, $P = ?$.
- For each brain region (r) & component (h), assign an individual-specific score $\eta_{ih[r]}$
- Characterize variation among individuals with:

$$\text{logit}\{\text{pr}(X_{i[u,v]} = 1)\} = \mu_{[u,v]} + \sum_{h=1}^{K} \lambda_{ih}\eta_{ih[u]}\eta_{ih[v]}, \quad \theta_i = \{\lambda_{ih}, \eta_{ir}\} \sim Q.$$

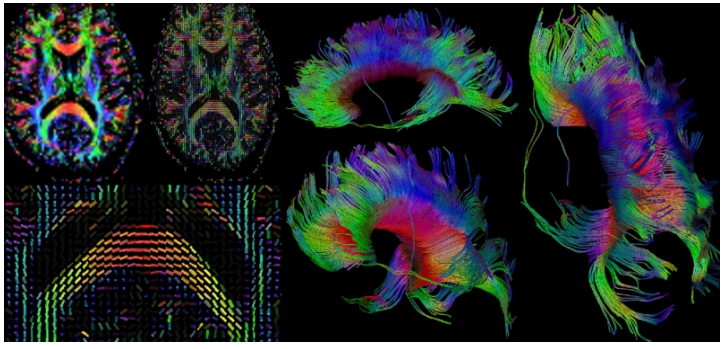
- Using Bayesian nonparametrics, allow Q (& P) to be unknown

Bayesian inferences



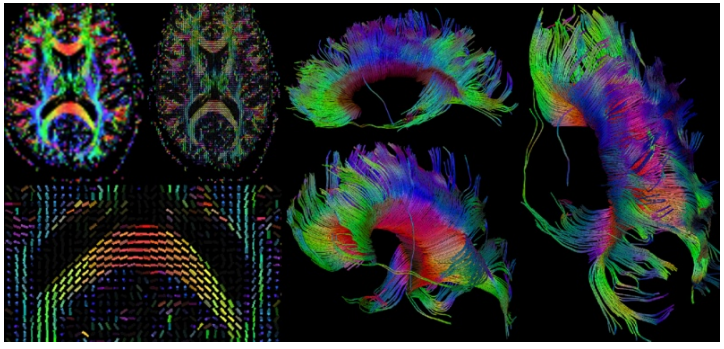
- Based on this framework, we can cluster individuals in terms of their brain structure

Bayesian inferences



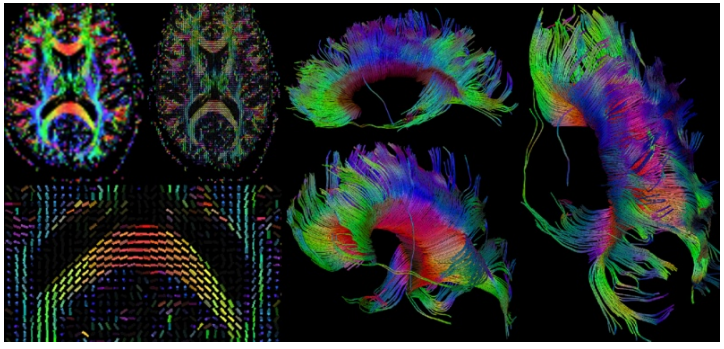
- Based on this framework, we can cluster individuals in terms of their brain structure
- We can also *test* for relationships between brain structure & traits/genotype

Bayesian inferences



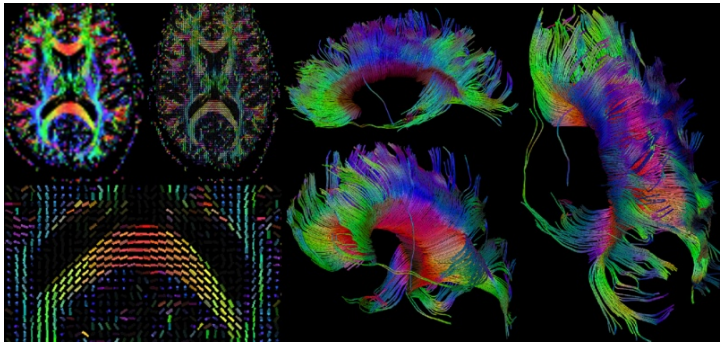
- Based on this framework, we can cluster individuals in terms of their brain structure
- We can also *test* for relationships between brain structure & traits/genotype
- Just allow the weights in our mixture model to vary with traits/genotypes with fixed kernels

Bayesian inferences



- Based on this framework, we can cluster individuals in terms of their brain structure
- We can also *test* for relationships between brain structure & traits/genotype
- Just allow the weights in our mixture model to vary with traits/genotypes with fixed kernels
- Allows scientific inference of global & local group differences in network structures with traits

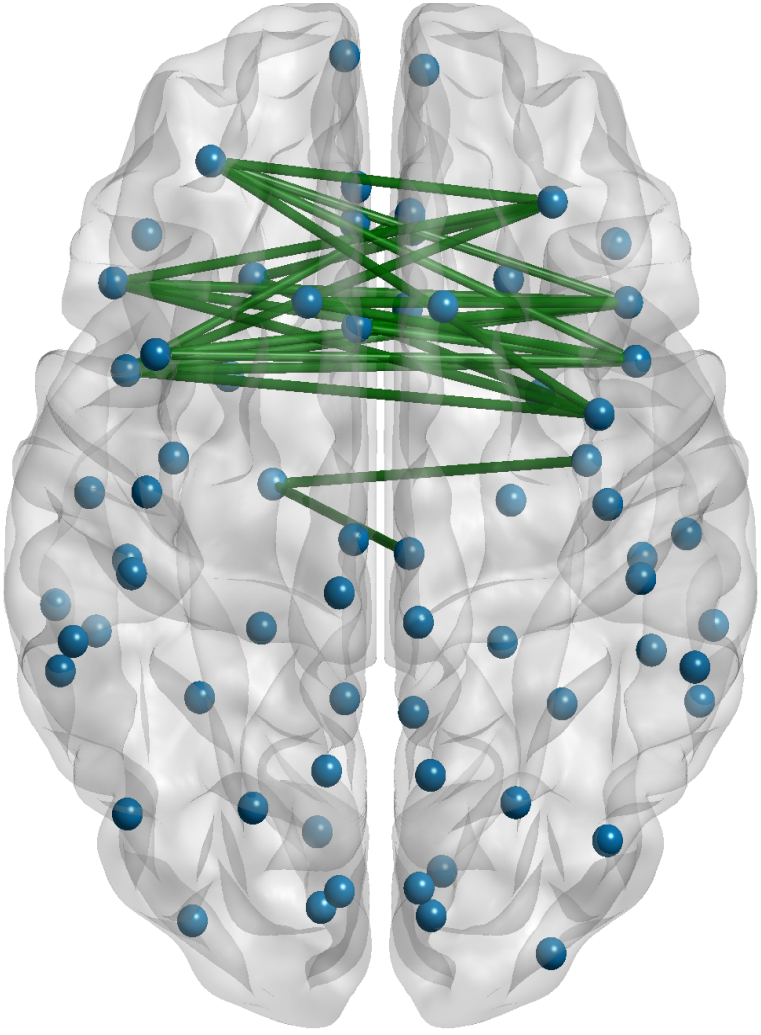
Bayesian inferences



- Based on this framework, we can cluster individuals in terms of their brain structure
- We can also *test* for relationships between brain structure & traits/genotype
- Just allow the weights in our mixture model to vary with traits/genotypes with fixed kernels
- Allows scientific inference of global & local group differences in network structures with traits
- Adjusts for multiple testing reducing false positives

Application to creativity

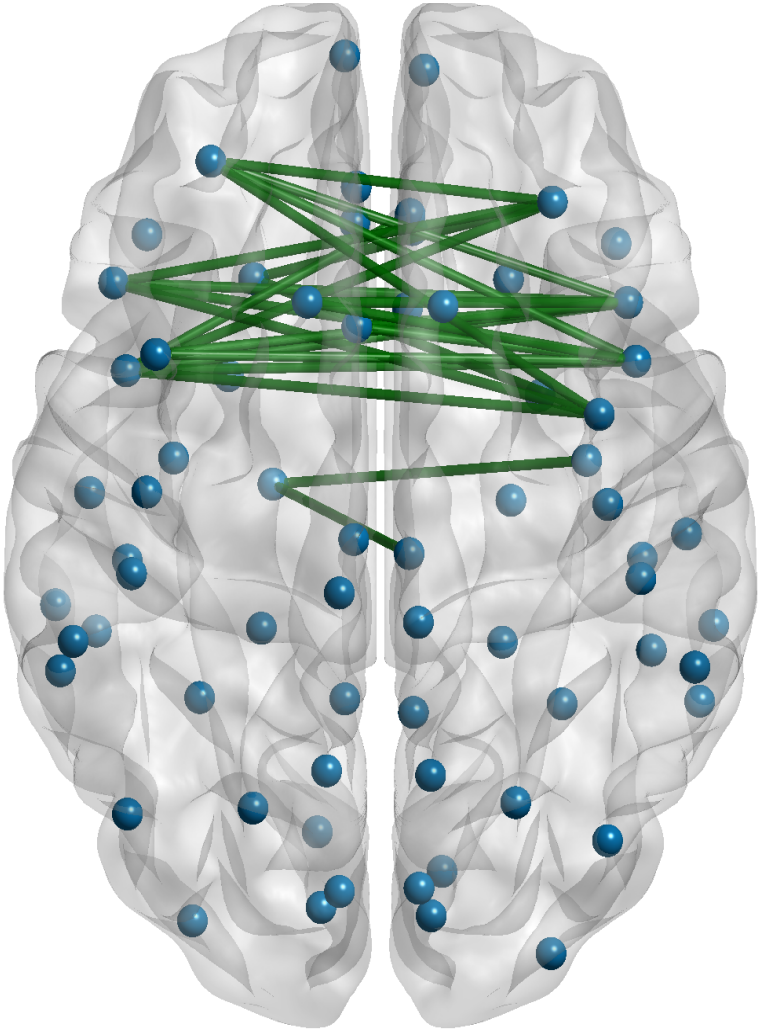
Results from local testing



- Apply model to brain networks of 36 subjects (19 with high creativity, 17 with low creativity—measured via CCI).

Application to creativity

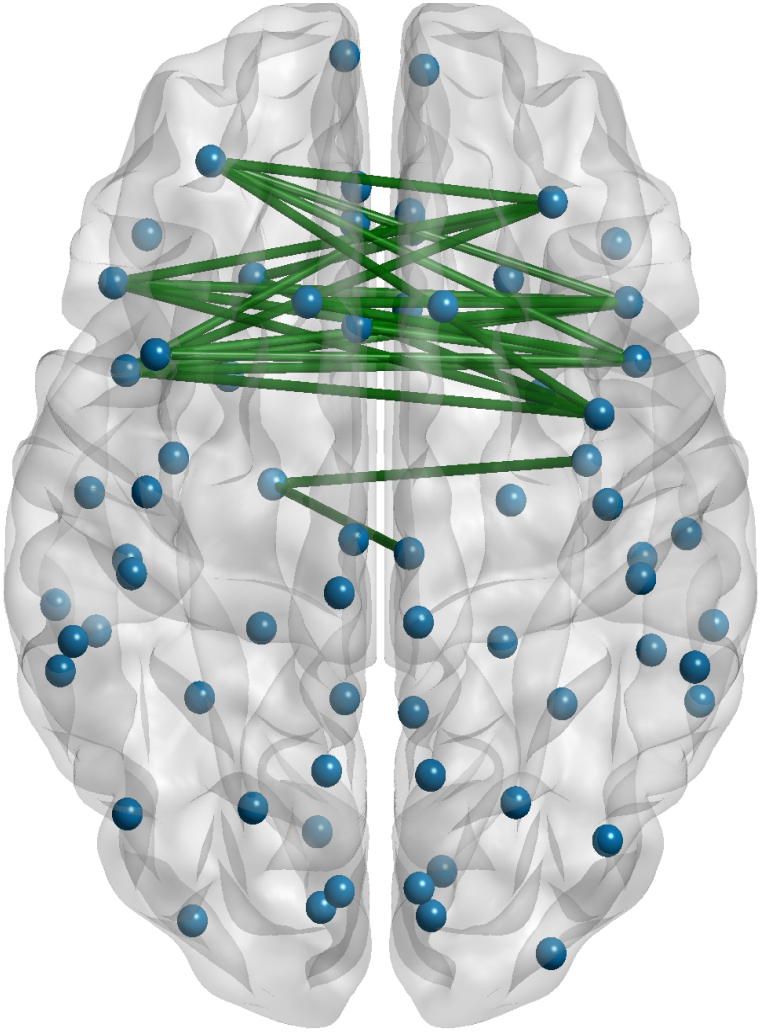
Results from local testing



- Apply model to brain networks of 36 subjects (19 with high creativity, 17 with low creativity—measured via CCI).
- $\hat{\text{pr}}(H_1 \mid \text{data}) = 0.995$.

Application to creativity

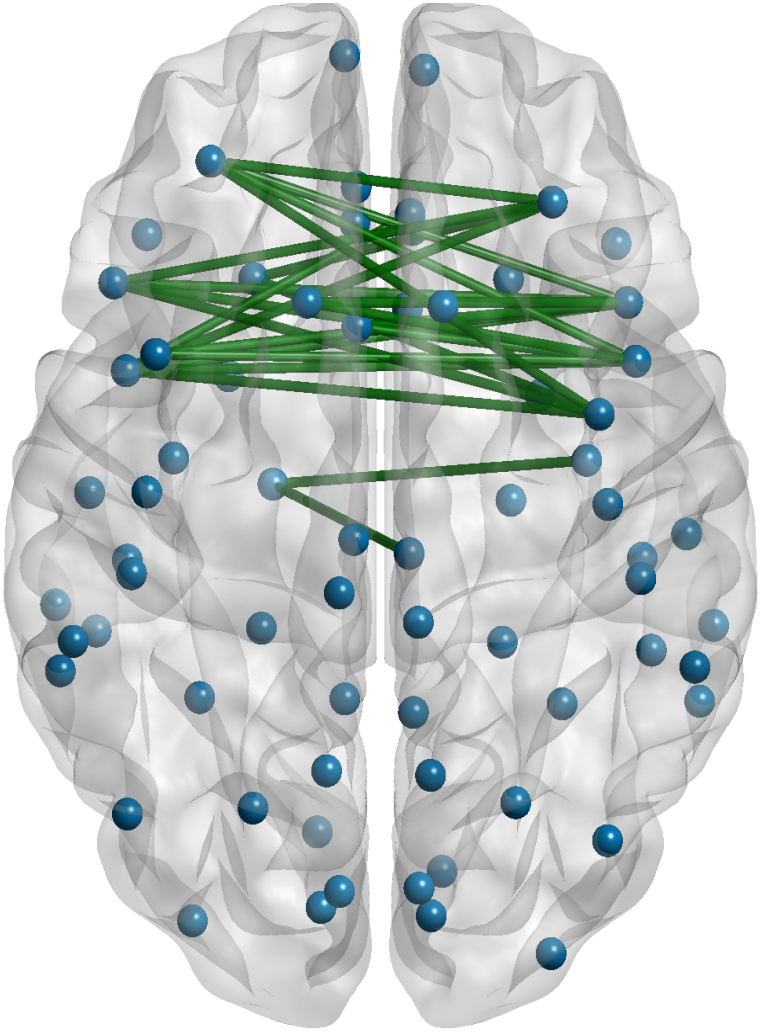
Results from local testing



- Apply model to brain networks of 36 subjects (19 with high creativity, 17 with low creativity—measured via CCI).
- $\hat{\text{pr}}(H_1 \mid \text{data}) = 0.995$.
- Highly creative individuals have significantly > inter-hemispheric connections.

Application to creativity

Results from local testing



- Apply model to brain networks of 36 subjects (19 with high creativity, 17 with low creativity—measured via CCI).
- $\hat{\text{pr}}(H_1 \mid \text{data}) = 0.995$.
- Highly creative individuals have significantly > inter-hemispheric connections.
- Differences in frontal lobe consistent with recent fMRI studies analyzing regional activity in isolation.

Modularization

- Idea: don't allow for all the dependencies implied by the joint Bayesian model

Modularization

- Idea: don't allow for all the dependencies implied by the joint Bayesian model
- *Cutting* dependence useful for computational scalability & robustness to model misspecification

Modularization

- Idea: don't allow for all the dependencies implied by the joint Bayesian model
- *Cutting* dependence useful for computational scalability & robustness to model misspecification
- As an example, suppose we have a phenotype y_i and SNPs $x_i = (x_{i1}, \ldots, x_{ip})'$

Modularization

- Idea: don't allow for all the dependencies implied by the joint Bayesian model
- *Cutting* dependence useful for computational scalability & robustness to model misspecification
- As an example, suppose we have a phenotype y_i and SNPs $x_i = (x_{i1}, \ldots, x_{ip})'$
- We want to screen for SNPs x_{ij} that are significantly related with y_i in a nonparametric manner

Modularization

- Idea: don't allow for all the dependencies implied by the joint Bayesian model
- *Cutting* dependence useful for computational scalability & robustness to model misspecification
- As an example, suppose we have a phenotype y_i and SNPs $x_i = (x_{i1}, \ldots, x_{ip})'$
- We want to screen for SNPs x_{ij} that are significantly related with y_i in a nonparametric manner
- We also want to account for dependence in the many different tests

Modular Bayes screening (Chen & Dunson, 2018)

- Start with kernel mixture model for marginal distribution of y_i:

$$f(y) = \sum_{h=1}^{k} \pi_h \mathcal{K}(y; \theta_h).$$

Modular Bayes screening (Chen & Dunson, 2018)

- Start with kernel mixture model for marginal distribution of y_i:

$$f(y) = \sum_{h=1}^{k} \pi_h \mathcal{K}(y; \theta_h).$$

- This implies $y_i \sim \mathcal{K}(\theta_{c_i})$, pr$(c_i = h) = \pi_h$, with $c_i \in \{1, \ldots, k\}$ a cluster index

Modular Bayes screening (Chen & Dunson, 2018)

- Start with kernel mixture model for marginal distribution of y_i:

$$f(y) = \sum_{h=1}^{k} \pi_h \mathcal{K}(y; \theta_h).$$

- This implies $y_i \sim \mathcal{K}(\theta_{c_i})$, $\text{pr}(c_i = h) = \pi_h$, with $c_i \in \{1, \ldots, k\}$ a cluster index
- Run an MCMC algorithm to get samples of the unknown parameters & cluster indices

Modular Bayes screening (Chen & Dunson, 2018)

- Start with kernel mixture model for marginal distribution of y_i:

$$f(y) = \sum_{h=1}^{k} \pi_h \mathcal{K}(y; \theta_h).$$

- This implies $y_i \sim \mathcal{K}(\theta_{c_i})$, pr$(c_i = h) = \pi_h$, with $c_i \in \{1, \ldots, k\}$ a cluster index
- Run an MCMC algorithm to get samples of the unknown parameters & cluster indices
- For each SNP, define a simple Bayesian test for association between x_{ij} & c_i

Modular Bayes screening (Chen & Dunson, 2018)

- Start with kernel mixture model for marginal distribution of y_i:

$$f(y) = \sum_{h=1}^{k} \pi_h \mathcal{K}(y; \theta_h).$$

- This implies $y_i \sim \mathcal{K}(\theta_{c_i})$, pr$(c_i = h) = \pi_h$, with $c_i \in \{1, \ldots, k\}$ a cluster index
- Run an MCMC algorithm to get samples of the unknown parameters & cluster indices
- For each SNP, define a simple Bayesian test for association between x_{ij} & c_i
- Include common parameters across these tests - eg, probability of an association in a random SNP.

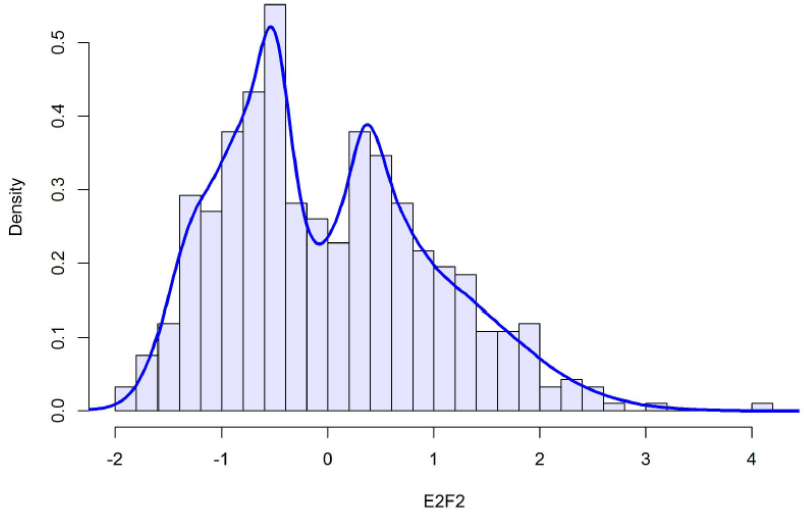
Modular Bayes screening (Chen & Dunson, 2018)

- Start with kernel mixture model for marginal distribution of y_i:

$$f(y) = \sum_{h=1}^{k} \pi_h \mathcal{K}(y; \theta_h).$$

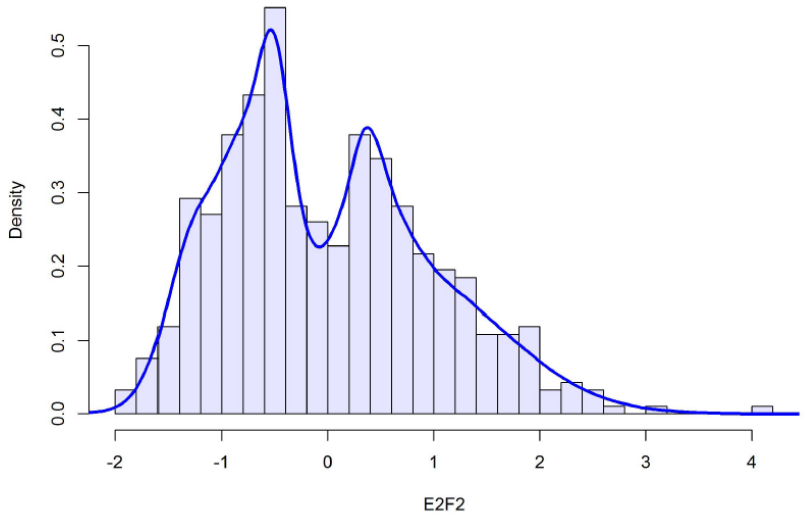
- This implies $y_i \sim \mathcal{K}(\theta_{c_i})$, pr$(c_i = h) = \pi_h$, with $c_i \in \{1, \ldots, k\}$ a cluster index
- Run an MCMC algorithm to get samples of the unknown parameters & cluster indices
- For each SNP, define a simple Bayesian test for association between x_{ij} & c_i
- Include common parameters across these tests - eg, probability of an association in a random SNP.
- Marginalize over MCMC samples of $\{c_i\}$ to take into account uncertainty

MOBS - comments



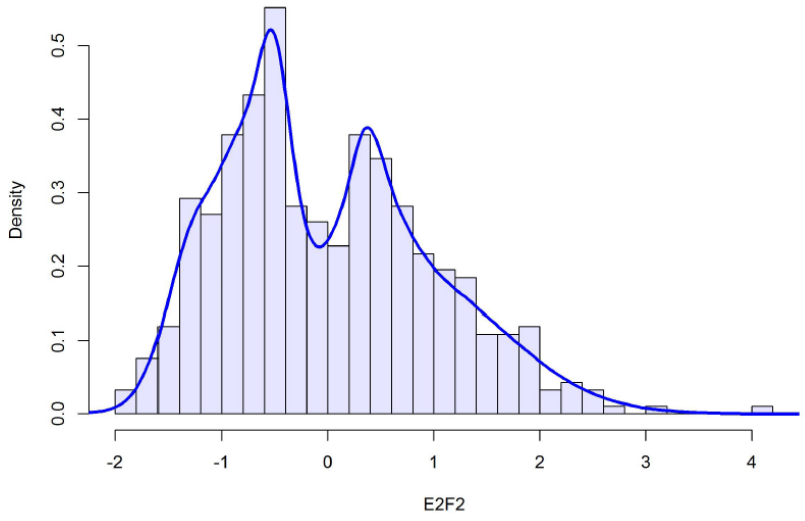
- Algorithm is very fast & scalable to huge p + trivially parallelizable

MOBS - comments



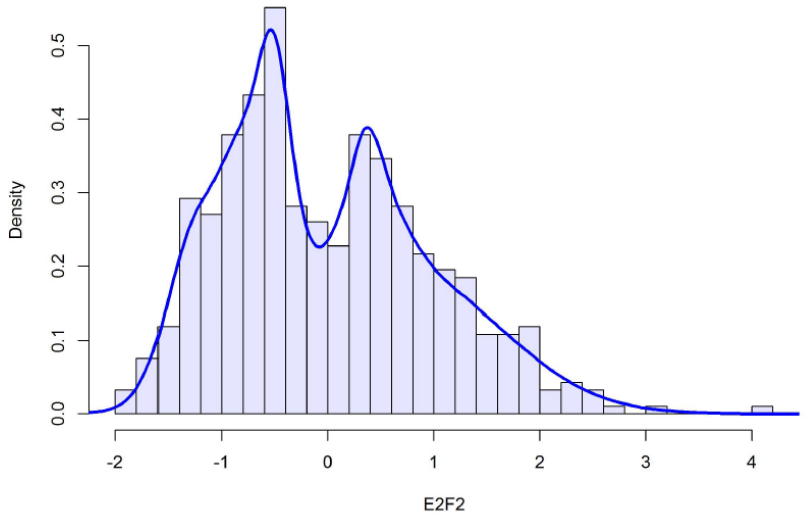
- Algorithm is very fast & scalable to huge p + trivially parallelizable
- Has strong frequentist theoretical guarantees - comparable to state-of-the-art

MOBS - comments



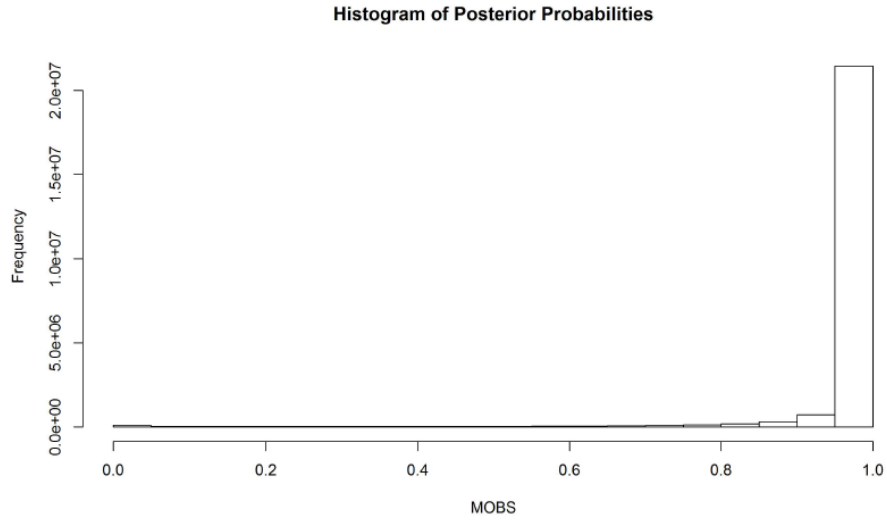
- Algorithm is very fast & scalable to huge p + trivially parallelizable
- Has strong frequentist theoretical guarantees - comparable to state-of-the-art
- Competitive with the state of the art in performance

MOBS - comments



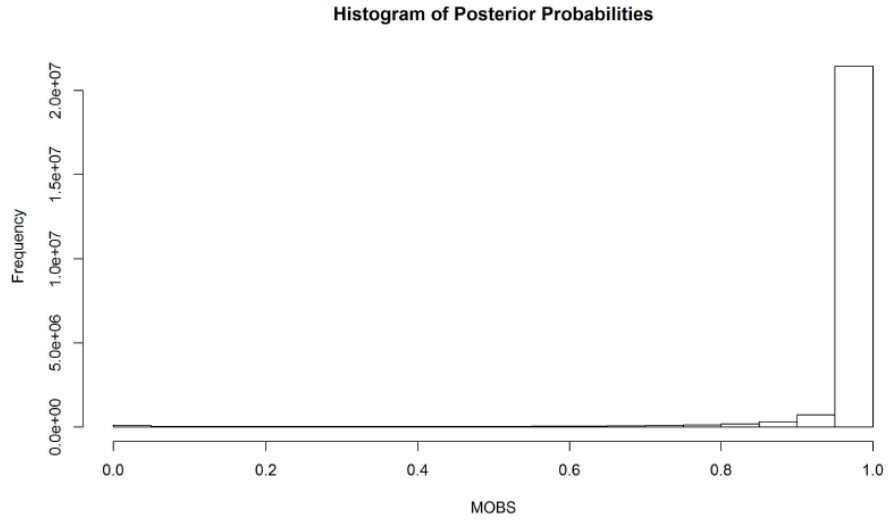
- Algorithm is very fast & scalable to huge p + trivially parallelizable
- Has strong frequentist theoretical guarantees - comparable to state-of-the-art
- Competitive with the state of the art in performance
- Particularly good at detecting complex distributional changes

Application to cis-eQTL data



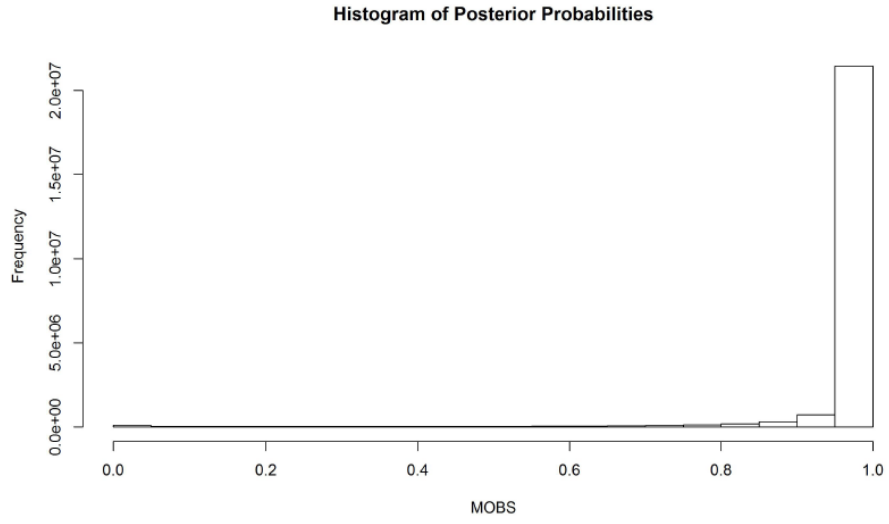
- Applied approach to GEUVADIS cis-eQTL data set

Application to cis-eQTL data



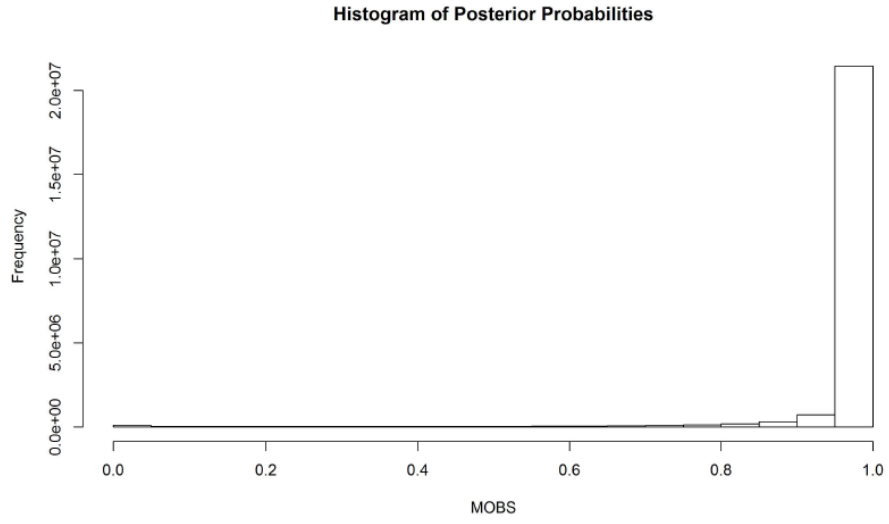
- Applied approach to GEUVADIS cis-eQTL data set
- Messenger RNA & microRNA on lymphoblastoid cell line samples from 462 individuals in 1000 genomes

Application to cis-eQTL data



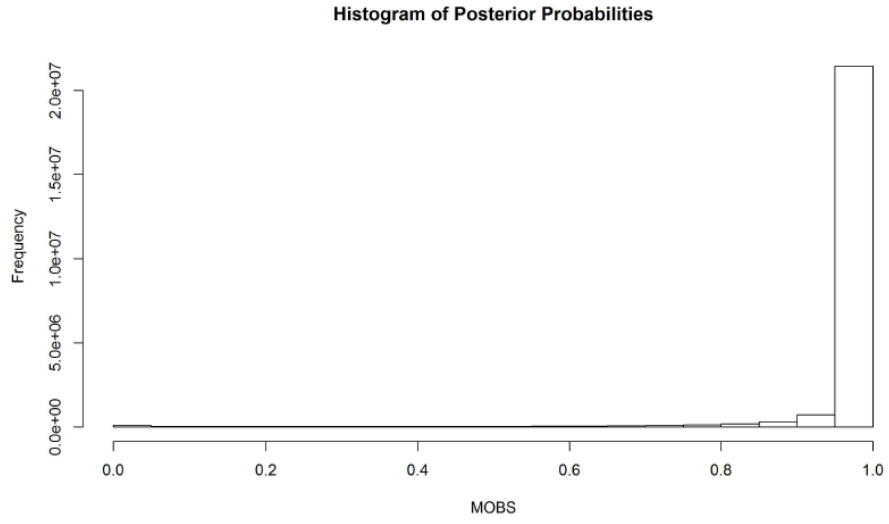
- Applied approach to GEUVADIS cis-eQTL data set
- Messenger RNA & microRNA on lymphoblastoid cell line samples from 462 individuals in 1000 genomes
- 38 million Single Nucleotide Polymorphisms (SNPs)

Application to cis-eQTL data



- Applied approach to GEUVADIS cis-eQTL data set
- Messenger RNA & microRNA on lymphoblastoid cell line samples from 462 individuals in 1000 genomes
- 38 million Single Nucleotide Polymorphisms (SNPs)
- gene E2F2 (y_i) - key role in control of cell cycle & is multimodal

Application to cis-eQTL data



- Applied approach to GEUVADIS cis-eQTL data set
- Messenger RNA & microRNA on lymphoblastoid cell line samples from 462 individuals in 1000 genomes
- 38 million Single Nucleotide Polymorphisms (SNPs)
- gene E2F2 (y_i) - key role in control of cell cycle & is multimodal
- 0.4% of pr(H_{0j}) < 0.05 - picking up differences in distribution other methods miss

Discussion

- Brief intro to Bayesian methods for large p problems

Discussion

- Brief intro to Bayesian methods for large p problems
- Highlighted some recent work using shared kernels &/or modularization

Discussion

- Brief intro to Bayesian methods for large p problems
- Highlighted some recent work using shared kernels &/or modularization
- There is a very rich literature & increasing focus on scalability

Discussion

- Brief intro to Bayesian methods for large p problems
- Highlighted some recent work using shared kernels &/or modularization
- There is a very rich literature & increasing focus on scalability
- One important direction is to obtain methods for assessing when we are attempting inferences on too fine of a scale for our data

Discussion

- Brief intro to Bayesian methods for large p problems
- Highlighted some recent work using shared kernels &/or modularization
- There is a very rich literature & increasing focus on scalability
- One important direction is to obtain methods for assessing when we are attempting inferences on too fine of a scale for our data
- Ideally can then automatically coarsen the scale to answer solvable questions - e.g., Peruzzi & Dunson (2018)

Some references - large n Bayes

- Miller J, Dunson D (2018) Robust Bayesian inference via coarsening. *Journal of the American Statistical Association*, Online.
- Srivastava S, Li C, Dunson DB (2018) Scalable Bayes via barycenter in Wasserstein space. *Journal of Machine Learning Research* 19(1):312-346.
- Li C, Srivastava S, Dunson DB (2017) Simple, scalable and accurate posterior interval estimation. *Biometrika* 104(3):665-680.
- Duan LL, Johndrow JE, Dunson DB (2018) Calibrated data augmentation for scalable Markov chain Monte Carlo. *Journal of Machine Learning Research*, to appear.
- Minsker S, Srivastava S, Lin L, Dunson DB (2017) Robust and scalable Bayes via a median of subset posterior measures. *Journal of Machine Learning Research* 18(1):4488-527.
- Johndrow JE, Smith A, Pillai N, Dunson DB (2018) MCMC for imbalanced categorical data. *Journal of the American Statistical Association*, Online.

Some references - large p Bayes

- Armagan A, Dunson DB, Lee J (2013) Generalized double Pareto shrinkage. *Statistica Sinica* 23:119.
- Bhattacharya A, Pati D, Pillai NS, Dunson D (2015) Dirichlet-Laplace priors for optimal shrinkage. *JASA* 110:1479-90.
- Chen Y, Dunson DB (2017) Modular Bayes screening for high-dimensional predictors. *Biometrika*, under revision.
- Datta J, Dunson DB (2016) Bayesian inferences on quasi-sparse count data. *Biometrika* 103:971-83.
- Durante D, Dunson DB (2018) Bayesian inference and testing of group differences in brain networks. *Bayesian Analysis* 13:29-58.
- Lee K, Lin L, Dunson D (2018) Maximum pairwise Bayes factors for covariance structure testing. *arXiv:1809.03105*
- Lock EF, Dunson DB (2015) Shared kernel Bayesian screening. *Biometrika* 102:829-842.
- Peruzzi M, Dunson DB (2018) Bayesian modular and multiscale regression. *arXiv:1809.05935*.

Discussion

- Take home message: Bayes is scalable & MCMC is scalable

Discussion

- Take home message: Bayes is scalable & MCMC is scalable
- But in big data & dimensionality problems we can't necessarily be naive & use off the shelf algorithms

Discussion

- Take home message: Bayes is scalable & MCMC is scalable
- But in big data & dimensionality problems we can't necessarily be naive & use off the shelf algorithms
- We need to think carefully about how to exploit parallel processing & accurate approximations to reduce bottlenecks

Discussion

- Take home message: Bayes is scalable & MCMC is scalable
- But in big data & dimensionality problems we can't necessarily be naive & use off the shelf algorithms
- We need to think carefully about how to exploit parallel processing & accurate approximations to reduce bottlenecks
- Also useful to take a step away from the fully Bayes framework by using modularization, composite likelihoods, c-Bayes, etc

Discussion

- Take home message: Bayes is scalable & MCMC is scalable
- But in big data & dimensionality problems we can't necessarily be naive & use off the shelf algorithms
- We need to think carefully about how to exploit parallel processing & accurate approximations to reduce bottlenecks
- Also useful to take a step away from the fully Bayes framework by using modularization, composite likelihoods, c-Bayes, etc
- Such generalized Bayes methods can have improved computational performance & robustness