Bayesian Analysis of Massive Datasets Via Particle Filters

Greg Ridgeway

RAND Statistics Group

http://i-pensieri.com/gregr

David Madigan

Rutgers University

http://stat.rutgers.edu/~madigan

Bayesian Analysis

- Use Bayes' theorem to learn about model parameters from data

$$f(\theta \mid \text{data}) \propto f(\text{data} \mid \theta) f(\theta)$$

- Examples:
  - Clustered data: hospitals, schools
  - Spatial models: public health
  - Support vector machines
  - Model based clustering

Metropolis-Hastings algorithm

1. Initialize θ_1
2. For i in 2, …, M
   a. Draw a proposal θ' from $q(\theta \mid \theta_{i-1})$
   b. Compute the acceptance probability

$$\alpha(\theta', \theta_{i-1}) = \min\left(1, \frac{f(\theta' \mid \mathbf{x}) q(\theta_{i-1} \mid \theta')}{f(\theta_{i-1} \mid \mathbf{x}) q(\theta' \mid \theta_{i-1})}\right)$$

   c. Set $\theta_i = \theta'$ with probability α

   Otherwise $\theta_i = \theta_{i-1}$

Important ideas

- Metropolis makes Bayesian analysis practical
- Metropolis often requires an enormous number of laps through the dataset
- Given a θ drawn from $f(\theta | \mathbf{x})$, the Metropolis algorithm produces a new draw having the same distribution
- Using particle filtering we reverse the inner and outer for-loops of Metropolis

Importance sampling

- Target distribution is $f(\theta | \mathbf{x})$
- Sampling distribution is $g(\theta)$

$$\int \theta\, f(\theta | x_1, \ldots, x_N)\, d\theta = \int \theta \frac{f(\theta | \mathbf{x})}{g(\theta)} g(\theta)\, d\theta$$

$$= \lim_{M \to \infty} \frac{\sum_{i=1}^{M} w_i \theta_i}{\sum_{i=1}^{M} w_i}$$

- θ_i has density $g(\theta)$ and
- $\mathrm{w}_i = f(\theta_i | \mathbf{x})/\mathrm{g}(\theta_i)$

Important ideas

- We cannot sample from $f(\theta|\mathbf{x})$ directly because the model is complex and $\mathbf{x}$ is massive
- Importance sampling allows us to sample from difficult to sample distributions
- For efficiency, $g(\theta)$ and $f(\theta|\mathbf{x})$ should be similar

Importance sampling for massive datasets

- Set the sampling distribution as

$$g(\theta) = f(\theta \mid x_1, \ldots, x_n)$$

where $n << \mathrm{N}$

- The importance weights greatly simplify

$$w_i = \frac{f(\theta_i \mid x_1, \ldots, x_N)}{f(\theta_i \mid x_1, \ldots, x_n)} \propto \prod_{j=n+1}^{N} f(x_j \mid \theta_i)$$

- Use Metropolis to sample from $g(\theta)$ and reweight the draws to look like a sample from $f(\theta \mid \mathbf{x})$

The algorithm

- Load as much data into memory as possible to form D_1
- Draw M times from $f(\theta | D_1)$ via Metropolis
- Purge D_1 from memory
- Set $w_i = 0$, $i = 1, \ldots, M$

For $j = n+1, \ldots, N$

{

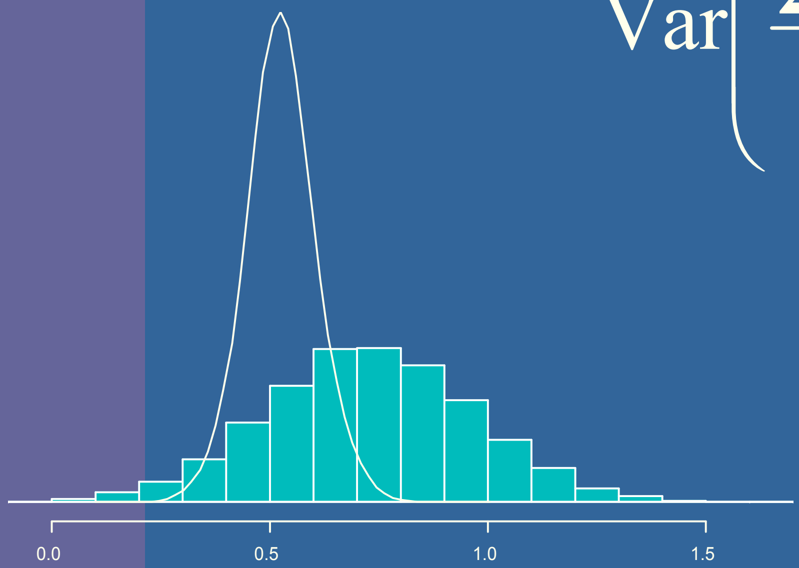
for $i = 1, \ldots, M$

$\log w_i = \log w_i + \log \mathrm{f}(\mathbf{x}_j | \theta_i)$

}

Transform and rescale to obtain the weights

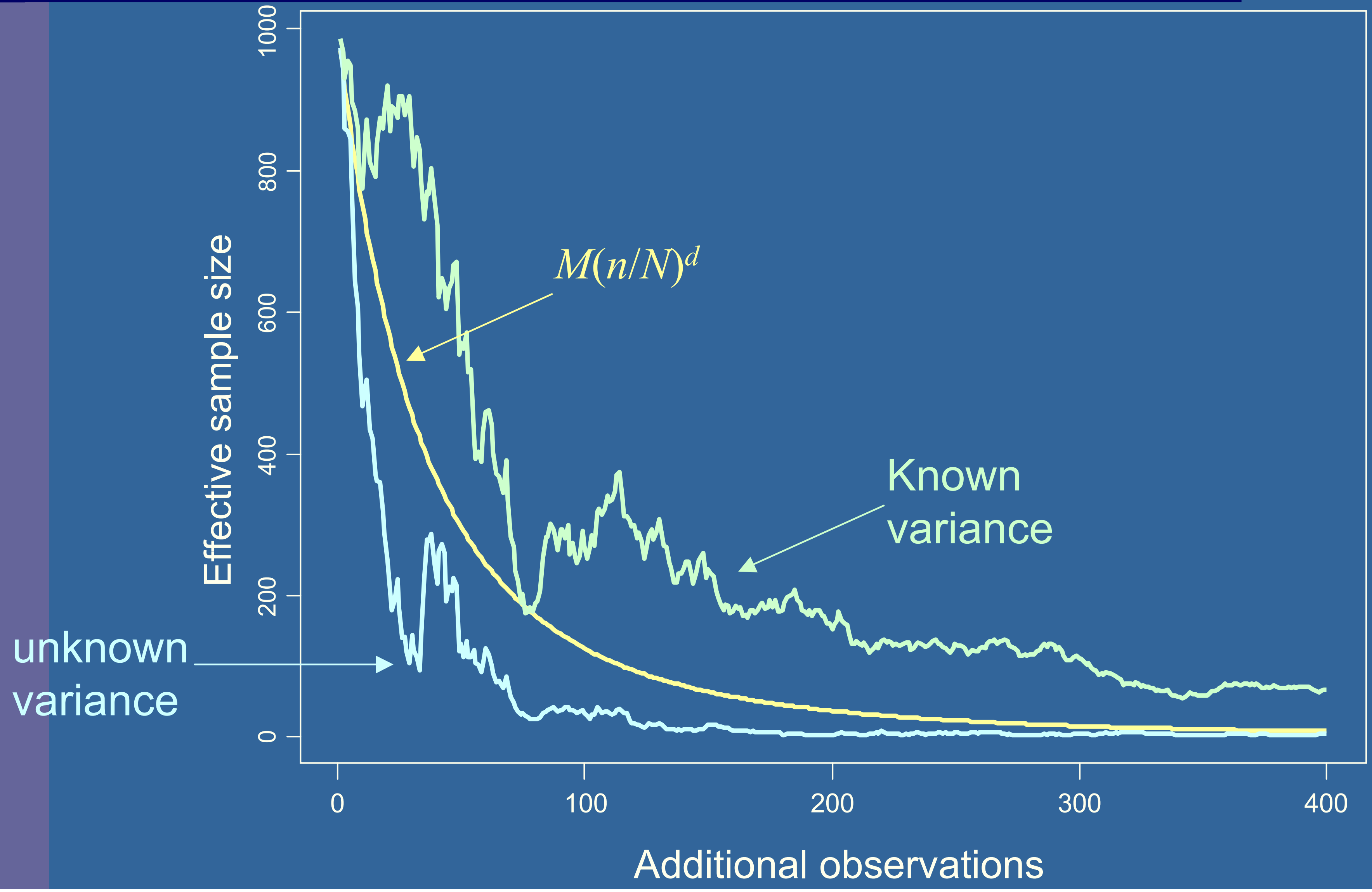
Effective sample size



$$\mathrm{Var}\left(\frac{\sum_{i=1}^{M} w_i\theta_i}{\sum_{i=1}^{M} w_i}\right) = \mathrm{Var}\left(\frac{1}{ESS}\sum_{i=1}^{ESS}\theta_i\right)$$

$$ESS = \frac{\left(\sum_{i=1}^{M} w_i\right)^2}{\sum_{i=1}^{M} w_i^2} \approx M\left(\frac{n}{N}\right)^d$$

ESS deterioration

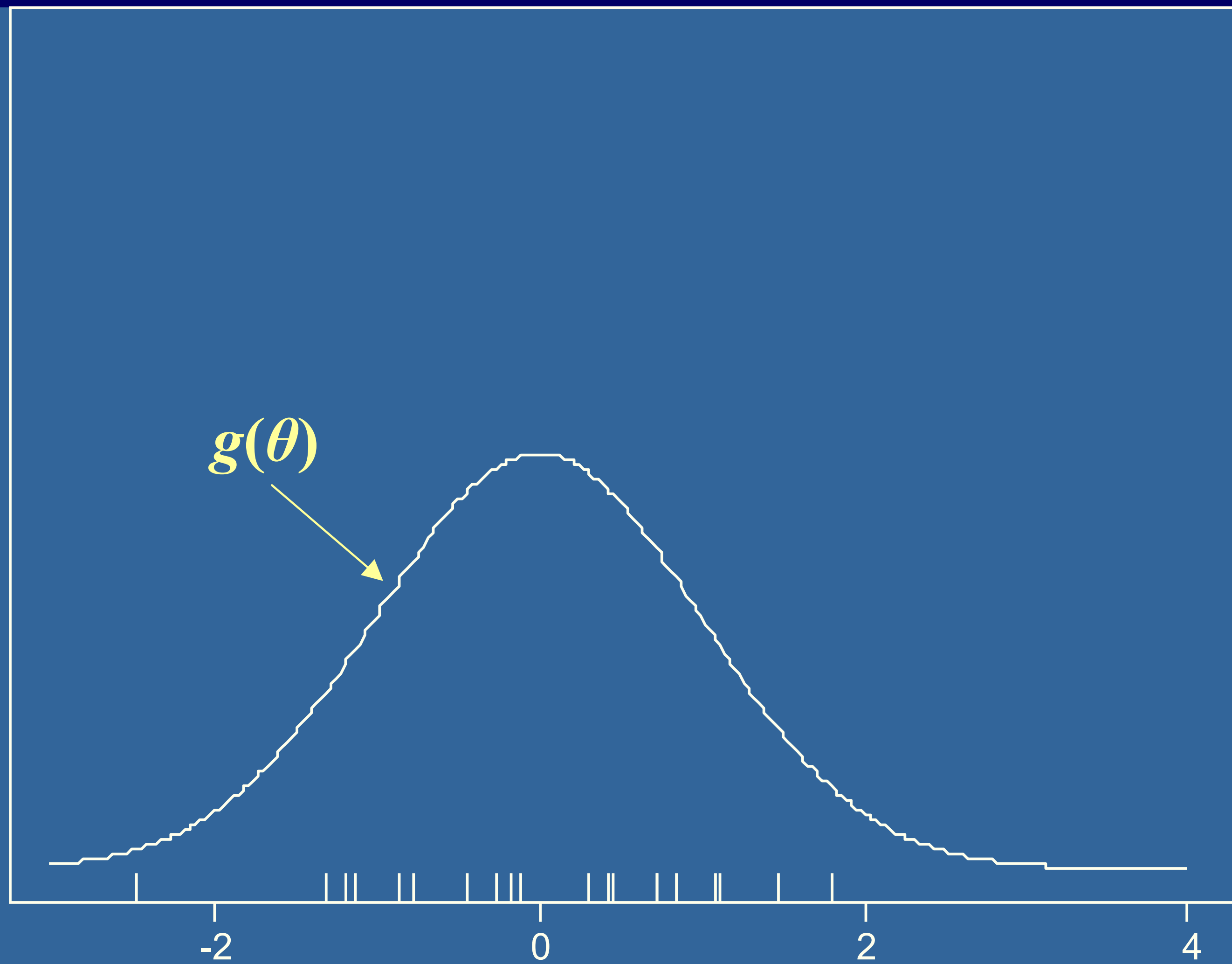


Gilks and Berzuini rejuvenation

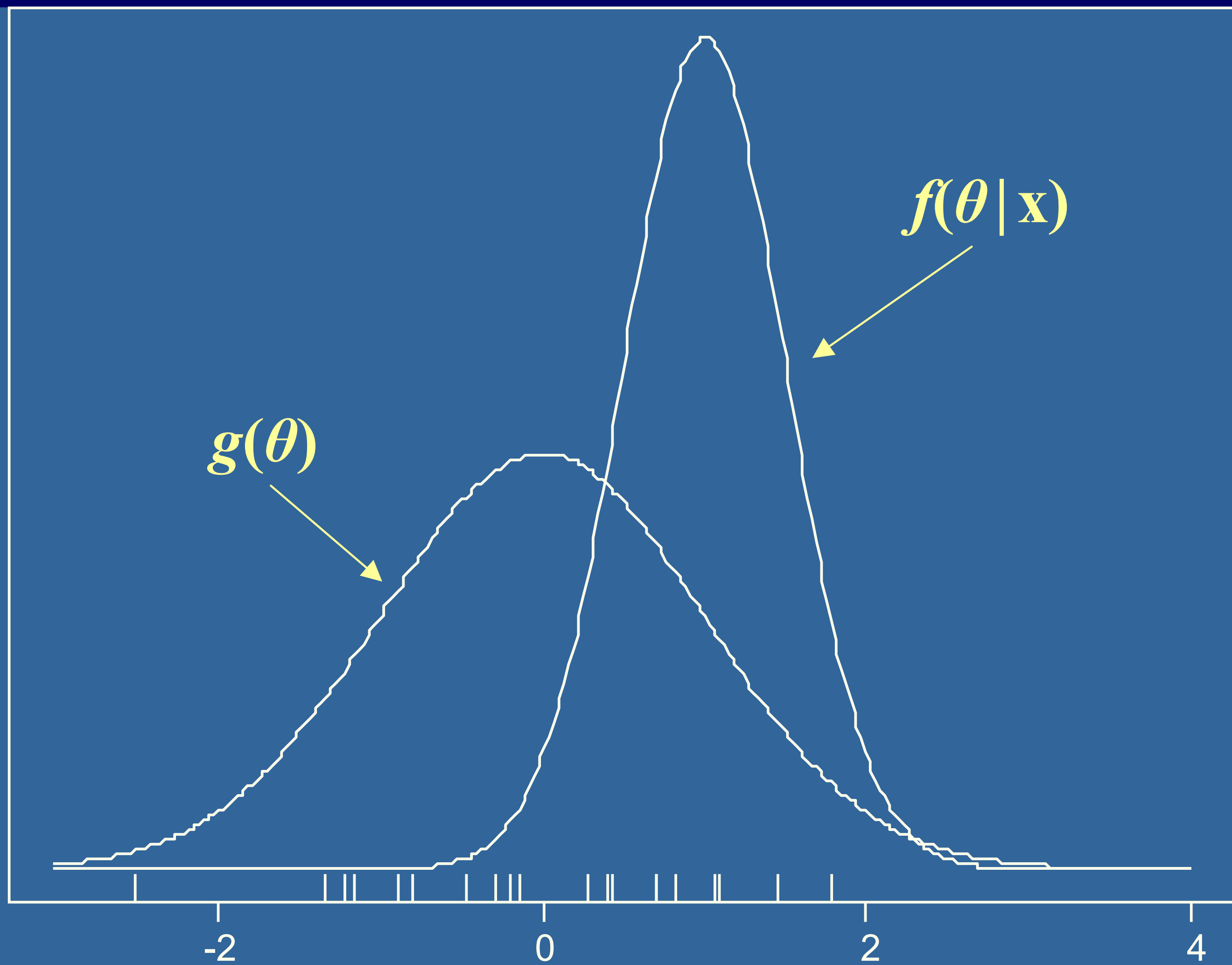
$\theta_1, \ldots, \theta_M$ are *particles* and the weights *filter* the θ_i with little posterior mass

- Get initial sample from $f(\theta | x_1, \ldots, x_n)$
- While ESS is large enough incorporate new observations using importance reweighting
- Sample with replacement from $\theta_1, \ldots, \theta_M$ with probability proportional to w_i
- Rejuvenate: For each θ_i do a single Metropolis step

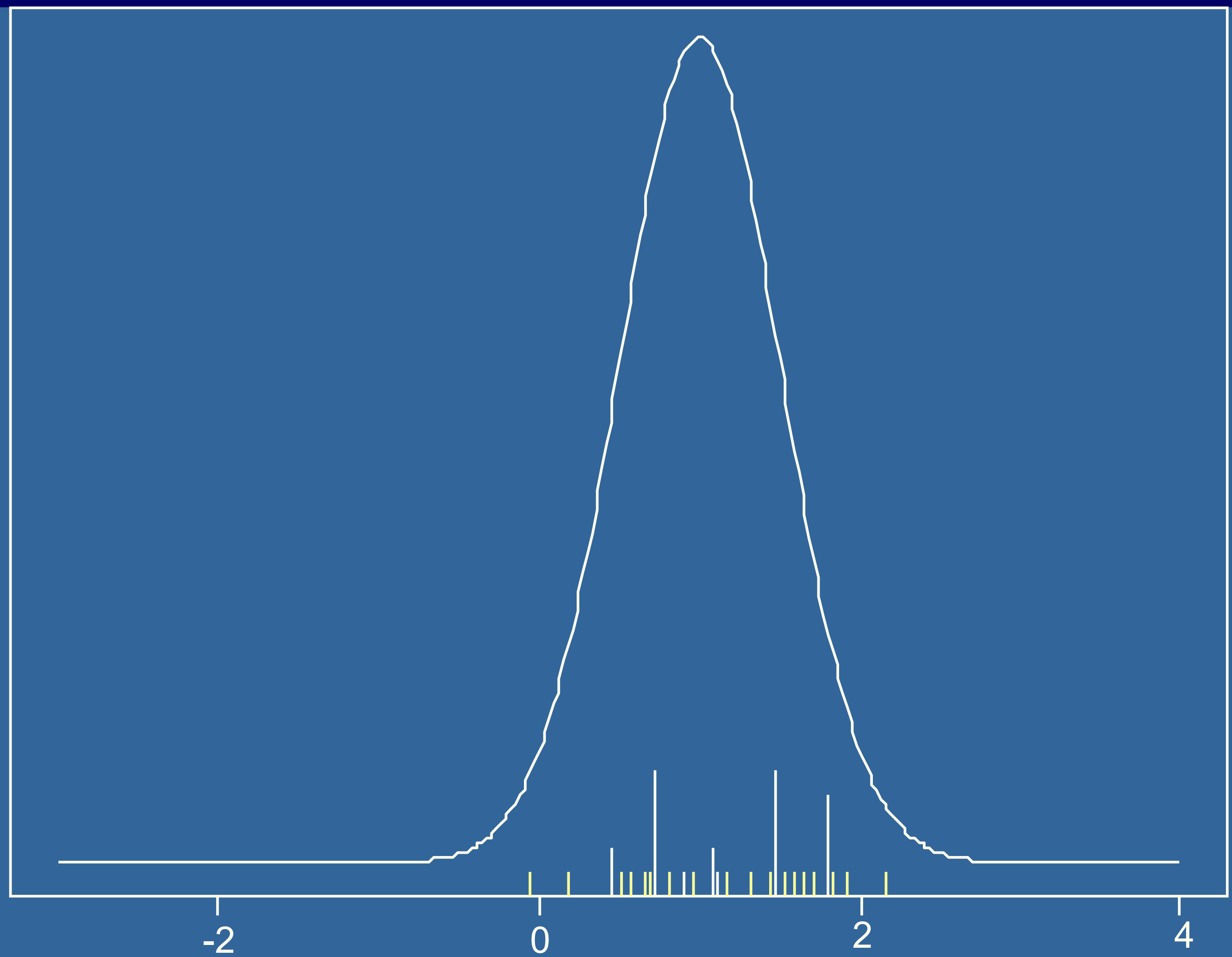
Sample from $g(\theta)$



Reweight, resample to get $f(\theta | \mathbf{x})$



Rejuvenate



Frequency of rejuvenation

Observations absorbed at refresh k

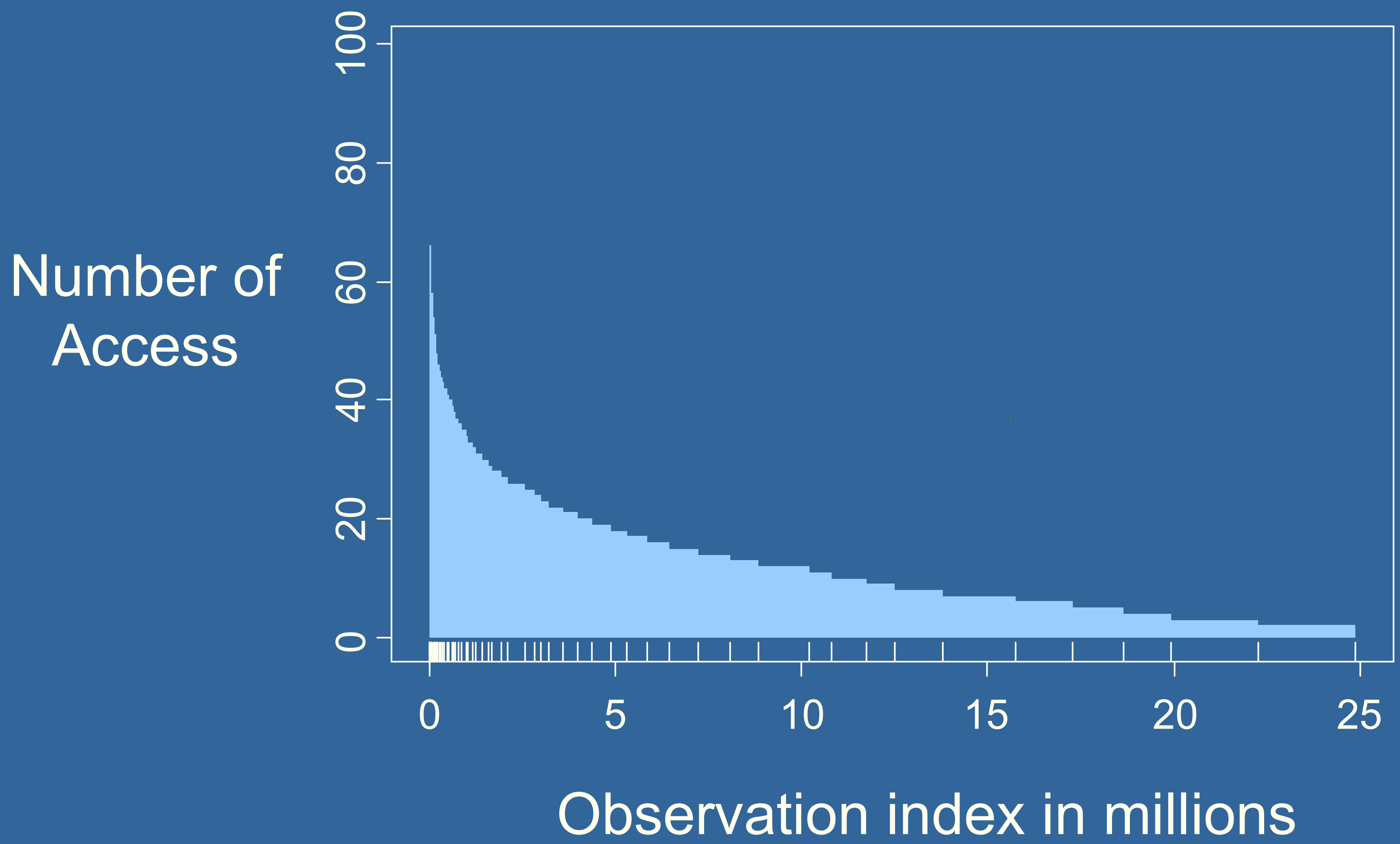
$$= n\left(\frac{1}{p}\right)^{\frac{k}{d}}$$

$$= \text{Initial dataset size} \times \left(\frac{1}{\text{percent decrease in ESS}}\right)^{\frac{k}{d}}$$

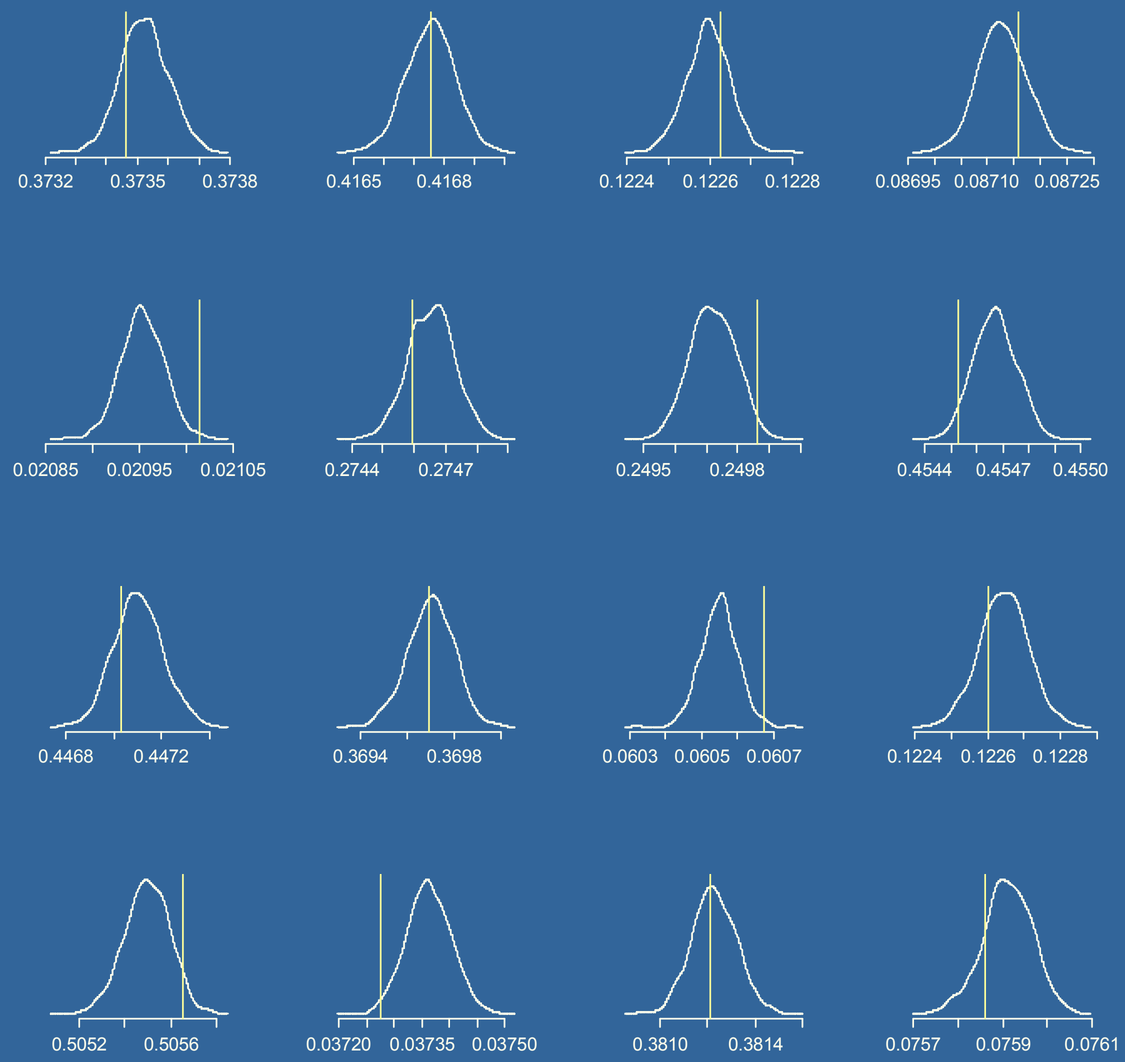
Example: Mixture of transition models

- Set of sequences of about 5 to 20 states visited by each observation
- Each sequence was generated by one of two first order probability transition matrices
- We do not know the transition probabilities nor the cluster assignments
- Properties
  - 25 million observations
  - 1 Gb of data
  - allowed only 1,000 sequences in memory

Number of accesses



Cluster 1 transition matrix



Conclusions

- Requires one good Metropolis-Hastings run up front with a small dataset
- Greatly reduces data access requirements
- Number of data accesses does not depend on M
- Chopin (2002) Biometrika article offers a similar strategy with interesting measures of sample quality

Bayesian Analysis of Massive Datasets Via Particle Filters

Greg Ridgeway
RAND Statistics Group
http://i-pensieri.com/gregr

David Madigan
Rutgers University
http://stat.rutgers.edu/~madigan