Mini-Batch Gibbs Sampler

Vadim Smolyakov, Qiang Liu, John W. Fisher III
Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory

March 2, 2016

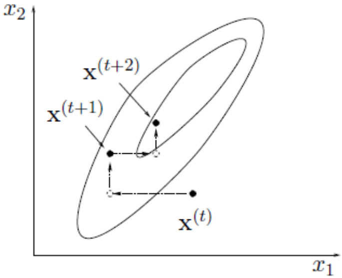
Outline

- Big Data Challenges: big data size, feature dimension, model complexity, computational complexity
- Approximate Scalable Inference: variational and Monte Carlo methods

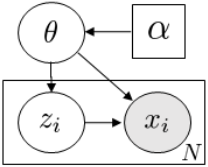
- A variety of scalable algorithms: stochastic, streaming and distributed Monte Carlo and Variational Bayes algorithms.
- Examples: SGLD, approximate MH, streaming VB, sequential MC, distributed VB, consensus MC
- Common goal: to estimate $P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$

- We focus on stochastic Monte Carlo algorithms
- In particular: the Gibbs sampler
- Basic Idea:

$$x_1^{t+1} \sim p(x_1|x_2^t)$$
$$x_2^{t+1} \sim p(x_2|x_1^{t+1})$$

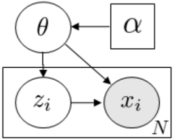


- Collapsed Gibbs sampler samples in a lower dimensional space since some of the latent variables are integrated out.
- A Gibbs sampler acceptance is 1 and it doesn't require a proposal distribution.
- In adaptive MCMC, one can change sampling parameters to increase efficiency, we focus on the mini-batch size.

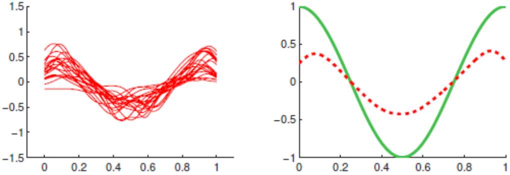


- The choice of mini-batch size controls the frequency of local and global updates
- Posterior distribution: $P(\theta, z|x; \alpha) = P(\theta; \alpha) \prod_{n=1}^{N} P(z_i|\theta) P(x_i|z_i, \theta)$
- Basic idea: replace big N with mini-batch m that minimizes sample variance
- Can be re-written as exponential family: $P(\theta, z) \propto \exp\{\psi(\theta) + \sum_i s(\theta, z_i)\}$
- The full conditional Gibbs sampler updates iterate between:
  $\{z_i\} \sim P(z_i|\theta) \propto \exp\{s(\theta, z_i)\}$
  $\theta \sim P(\theta|\{z_i\}) \propto \exp\{\psi(\theta) + \sum_{i=1}^{N} s(\theta, z_i)\}$

- Consider the following update frequencies:
  $\theta \rightarrow z_1 \rightarrow \theta \rightarrow z_2 \rightarrow \theta \rightarrow z_3 \rightarrow \ldots$
  $\theta \rightarrow z_1 \rightarrow z_2 \rightarrow \theta \rightarrow z_3 \rightarrow z_4 \rightarrow \ldots$
  $\theta \rightarrow \theta \rightarrow z_1 \rightarrow \theta \rightarrow \theta \rightarrow z_2 \rightarrow \ldots$

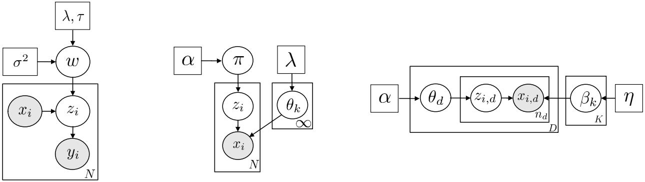


- Frequent updates of θ increase the number of θ-samples and therefore reduce the variance:
  $E[(\bar{f} - \hat{f})^2] = \frac{1}{n^2} E[\sum_{i=1}^{n} (f_i - \hat{f})^2] + \frac{1}{n^2} \sum_{s \neq t} E[(f_s - \hat{f})(f_t - \hat{f})]$
- On the other hand larger mini-batch sizes result in lower auto-correlation ρ_t and therefore greater information content per θ-sample.



- $\mathrm{MSE} = \mathbb{E}\left[(\hat{\theta} - \theta^*)^2\right] = \mathrm{VAR}\left[\hat{\theta}\right] + (\mathbb{E}[\hat{\theta}] - \theta^*)^2$
- Goal: to minimize variance: $\mathrm{VAR}[\hat{\theta}] \approx \frac{\sigma^2}{n}[1 + 2\sum_{t=1}^{\infty} \rho_t] = \frac{\sigma^2}{n}\tau_{int}(m)$
- Given a fixed time budget T, the number of θ-samples we can get is: $n = \frac{T}{mw_z + w_\theta}$, where m is the mini-batch size, w_z is local update time and w_θ is global update time.
- Thus, we can re-write the variance objective as: $\min_m (mw_z + w_\theta)\tau_{int}(m)$

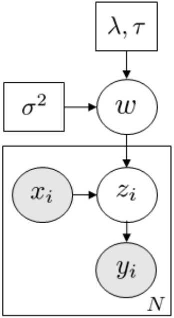
- Adaptive Batch Size Algorithm:
  Define the mini-batch range $M = \{m_1, m_2, ..., m_M\}$ and the number of samples n
  For $m = m_1, m_2, ..., m_M$
  Run MCMC chain after burnin with batch size equal to m for n iterations
  Record n samples of θ and compute τ_{int}
  Compute the objective: $f(m) = (mw_z + w_\theta)\tau_{int}(m)$
  End For
  Return $m^* = \arg\min_m f(m)$.



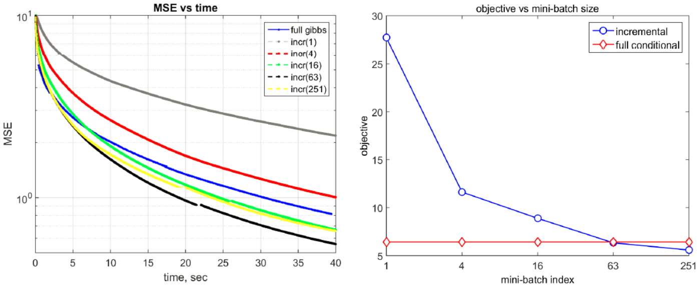
- Bayesian Lasso (left), Dirichlet Process Mixture Model (center), Latent Dirichlet Allocation (right)
- Goal: achieve lower MSE for fixed time budget by optimizing mini-batch size

Mini-Batch Gibbs Sampler
Bayesian Lasso

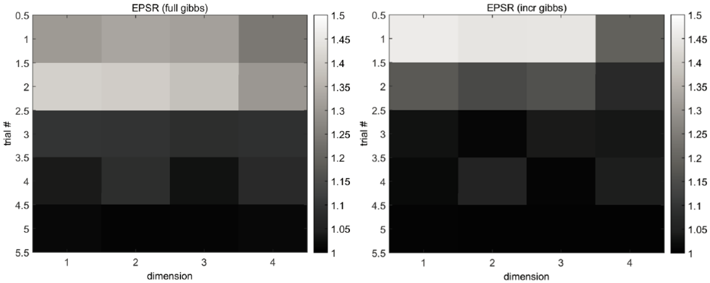
- Bayesian Binary Classification
- Goal: find a separating hyperplane $w \in \mathbb{R}^d$ given training data $D = \{(x_i, y_i) : i = 1 \ldots n\}$
- Generative model:
  $w_i \sim \mathrm{Laplace}(\lambda/\sigma)$ or
  $w_i \sim N(0, \sigma^2 D_\tau), \tau_i \sim \mathrm{Exp}(\lambda^2/2)$
  $z_i \sim N(w^T x_i, \sigma^2 I_d)$
  $y_i = sgn(z_i)$



- Gibbs sampling algorithm consists of local updates:
  $z_i \sim N(w^T x_i, \sigma^2 I)1[z_i \geq 0]$, if $y_i \geq 0$ and $z_i \sim N(w^T x_i; \sigma^2 I)1[z_i < 0]$, otherwise
  and global updates: $w \sim N(A^{-1} X^T y, \sigma^2 A^{-1})$, where $A = X'X + D_\tau^{-1}$.



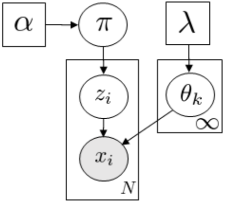
- Goal: achieve lower MSE for fixed time budget by stochastic updates of global parameters.
- UCI a9a dataset: 32,561 data points, 123 features
- Finding: mini-batch size 63 achieves lower MSE for Probit regression with Laplace prior



- Estimated Potential Scale Reduction (EPSR) convergence criterion for full Gibbs sampler (left) and optimum mini-batch Gibbs sampler (right). $\hat{R} = \sqrt{\frac{\mathrm{var}(\psi|\mathrm{y})}{W}}$
- where $\mathrm{var}(\psi|y) = \frac{n-1}{n}W + \frac{1}{n}B$, with B and W representing the in-between chain and within-chain variance, respectively

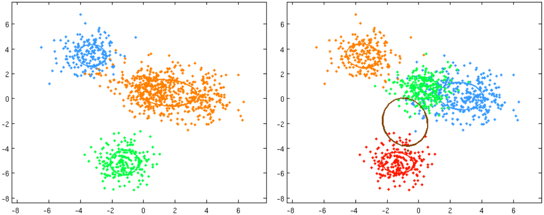
Mini-Batch Gibbs Sampler
Dirichlet Process Mixture Models

- Bayesian Non-parametric Clustering
- Goal: fit an infinite mixture of Gaussians given unlabelled data $\{x_i : i = 1, ..., n\}$
- $G(\theta) = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}(\theta)$

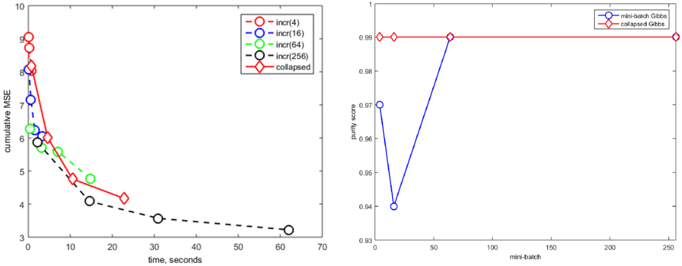


- Generative model:
  $\pi \sim \mathrm{GEM}(\alpha)$
  $z_i \sim \mathrm{Cat}(\pi)$
  $\theta_k \sim H(\lambda)$
  $x_i \sim F(\theta_{z_i})$

- The Gibbs sampling algorithm consists of local updates: $p(z_i = k|z_{-i}) \propto \frac{N_k}{N+\alpha-1} p(x_i|x_{k\setminus i})$ and $p(z_i = \mathrm{new}|z_{-i}) \propto \frac{\alpha}{\alpha+N-1} p(x_i|x_{k\setminus i})$ and global updates: μ_k and Σ_k based on the new z_i.



- DPMM using collapsed Gibbs sampler (left) and mini-batch Gibbs sampler (right)
- $N = 1e3$ points in R^2, $\alpha = 1$, $K_{gt} = 5$

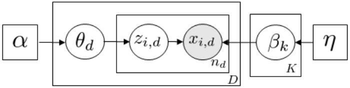


- MSE vs iterations plot (left) and purity scores vs mini-batch size plot (right)
- $N = 1e3$ points in R^2, $\alpha = 1$, $K_{gt} = 5$
- Purity score: $\sum_i \frac{N_i}{N} p_i$, where $N_i = \sum_{j=1}^{C} N_{ij}$, $p_i = \max_j p_{ij}$, and $p_{ij} = N_{ij}/Ni$, N_{ij} is the number of objects in cluster i that belong to class j.

Mini-Batch Gibbs Sampler

Latent Dirichlet Allocation

- Topic Modeling
- Goal: extract thematic information out of textual data
- Generative model:
  $\theta_d \sim \mathrm{Dir}(\alpha_1, ..., \alpha_K)$
  $z_{id} \sim \mathrm{Cat}(\theta_d)$
  $\beta \sim \mathrm{Dir}(\eta_1, ..., \eta_V)$
  $x_{id} \sim \mathrm{Cat}(\beta_k)$



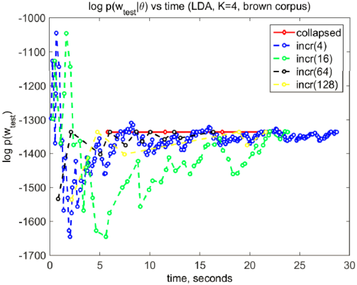
- The gibbs sampling algorithm consists of local updates:

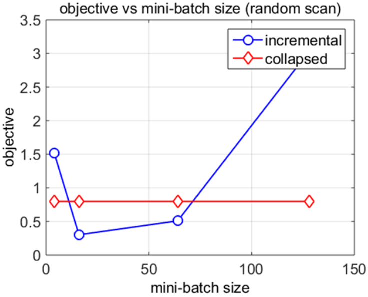
$$p(z_i = j | z_{-i}, w) \propto \frac{n^{(w_i)}_{-i,j} + \beta_i}{n^{(\cdot)}_{-i,j} + \sum_i \beta_i} \times \frac{n^{(d_i)}_{-i,j} + \alpha_j}{n^{(d_i)}_{-i,\cdot} + \sum_j \alpha_j}$$

and global updates: β_k based on the new z_i.

Mini-Batch Gibbs Sampler

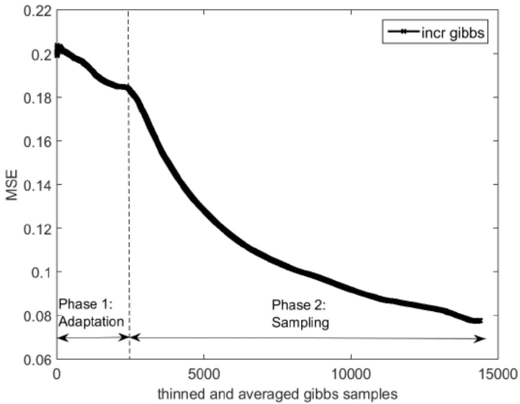
Latent Dirichlet Allocation





- LDA perplexity (left) and objective function (right)
- Brown corpus with $K = 4$ topics, $V = 6K$ dictionary, and $D = 250$ documents
- $\text{Perplexity}(w_{test}) = \exp\{-\frac{1}{D_{test}} \sum_d \frac{1}{n_d} \sum_{w \in n_d} \log p(w_{test})\}$

Mini-Batch Gibbs Sampler
Discussion



- Optimum mini-batch size $m = \{1, 2, ..., M\}$ is selected during adaptation phase and used during the sampling phase
- Suited for models with a hierarchical structure (local and global parameters)
- Less likely to stuck in local optima due to stochastic updates of global parameters
- Relies on commonly used MCMC diagnostic functions such as integrated autocorr