Scaling to big data

Bayesian Methodology in Biostatistics (BST 249)

Jeffrey W. Miller

Department of Biostatistics
Harvard T.H. Chan School of Public Health

Outline

Outline

Outline

Consensus Monte Carlo: Motivation

- As data sets grow, it eventually becomes infeasible to hold them in memory on a single machine.
- One way of dealing with this is to split the data into "shards", and process each shard on a different machine.
- However, sharing information across machines in a fully integrated way is difficult because:
  1. between-machine communication is slow, and
  2. it can be tricky to code algorithms involving communication between multiple asynchronous processes.
- If possible, it is simpler to independently process each shard, and combine the results in some way.

Consensus Monte Carlo: Setup

- Basic idea: Run a sampling algorithm on multiple machines, independently, and combine the samples at the end by taking a weighted average.

- Consider a generic i.i.d. model $X_1, \ldots, X_n \overset{\text{iid}}{\sim} p(x|\theta)$ along with a prior $p(\theta)$.

- Assume $\theta \in \mathbb{R}^d$ is a continuous parameter vector.

- Split the data into S subsets, or "shards": for $s = 1, \ldots, S$,

$$x_{A_s} = (x_i : i \in A_s)$$

where $(A_1, \ldots, A_S)$ is a partition of $\{1, \ldots, n\}$.

Consensus Monte Carlo: Method

- For shard s, define

$$q(\theta|x_{A_s}) \propto p(x_{A_s}|\theta)q(\theta)$$

  to be the posterior based on that shard only, using a downweighted prior $q(\theta)$.

- A natural choice of downweighted prior is $q(\theta) \propto p(\theta)^{1/S}$.

- The overall posterior $p(\theta|x_{1:n})$ can then be written as

$$\begin{aligned} p(\theta|x_{1:n}) &\propto p(x_{1:n}|\theta)p(\theta) \\ &= \prod_{s=1}^{S} p(x_{A_s}|\theta)p(\theta)^{1/S} \\ &\propto \prod_{s=1}^{S} q(\theta|x_{A_s}). \end{aligned}$$

Consensus Monte Carlo: Method

- Let $\theta_s^{(1)}, \ldots, \theta_s^{(T)}$ be samples from $q(\theta|x_{A_s})$ (e.g., via MCMC).

- The consensus Monte Carlo samples are defined as

$$\theta_t = \Big(\sum_{s=1}^{S} \Lambda_s\Big)^{-1} \sum_{s=1}^{S} \Lambda_s \theta_s^{(t)}$$

  for $t = 1, \ldots, T$, where Λ_s is a weight matrix that needs to be chosen appropriately.

- A natural choice of weight matrix is the posterior precision under $q(\theta|x_{A_s})$, that is, choose Λ_s to be the inverse of the covariance matrix of $\theta \sim q(\theta|x_{A_s})$.

- The consensus MC method is justified by a Gaussian approximation to $q(\theta|x_{A_s})$, as follows.

Outline

Consensus Monte Carlo: Justification (1/2)

- Under fairly general conditions, the posterior is asymptotically Gaussian as the data grows. Hence, it is natural to suppose that $q(\theta|x_{A_s})$ is approximately Gaussian.

- For the moment, let's suppose $q(\theta|x_{A_s})$ is exactly Gaussian:

$$q(\theta|x_{A_s}) = \mathcal{N}(\theta \mid \mu_s, \Lambda_s^{-1}).$$

- By the properties of Gaussians,

$$\prod_s \mathcal{N}(\theta \mid \mu_s, \Lambda_s^{-1}) \propto \mathcal{N}(\theta \mid \mu, \Lambda^{-1})$$

  where $\Lambda = \sum_s \Lambda_s$ and $\mu = \Lambda^{-1} \sum_s \Lambda_s \mu_s$.

- Then, the overall posterior is also Gaussian:

$$p(\theta|x_{1:n}) \propto \prod_s q(\theta|x_{A_s}) \propto \mathcal{N}(\theta \mid \mu, \Lambda^{-1}).$$

Consensus Monte Carlo: Justification (2/2)

- Now, we see how the consensus MC method yields approximate samples from the posterior.

- If $\theta_s^{(t)} \sim \mathcal{N}(\theta \mid \mu_s, \Lambda_s^{-1})$ independently for $s = 1, \ldots, S$, then

$$\textstyle\sum_s \Lambda_s \theta_s^{(t)} \sim \mathcal{N}\big(\sum_s \Lambda_s \mu_s,\ \sum_s \Lambda_s\big)$$

  by the properties of Gaussians. Therefore,

$$\textstyle\big(\sum_s \Lambda_s\big)^{-1} \sum_s \Lambda_s \theta_s^{(t)} \sim \mathcal{N}\big(\mu, \Lambda^{-1}\big)$$

  where μ and Λ are exactly the same as on the previous slide.

- Since the left-hand side is precisely the consensus MC sample, θ_t, and the right-hand side is the overall posterior, we have

$$\theta_t \sim \mathcal{N}\big(\theta \mid \mu, \Lambda^{-1}\big) = p(\theta | x_{1:n}).$$

Consensus Monte Carlo: Practicalities

- Typically, $q(\theta|x_{A_s})$ is not exactly Gaussian and we only have approximate samples from $q(\theta|x_{A_s})$, so the consensus MC samples are approximate draws from the posterior.

- Further, we typically cannot compute Λ_s, the precision matrix of $q(\theta|x_{A_s})$, exactly.

- If $\dim(\theta)$ is not too big, we can use the samples themselves to estimate Λ_s using the sample precision matrix.

- However, if $\dim(\theta)$ is large, then a regularized or constrained estimate of Λ_s may be preferable.

- E.g., one could use $\Lambda_s = \mathrm{diag}(\lambda_{s1}, \ldots, \lambda_{sd})$ where λ_{sj} is the sample precision of the jth component of θ under $q(\theta|x_{A_s})$.

Consensus Monte Carlo: Algorithm

1. Let $(A_1, \ldots, A_S)$ be a random partition of $\{1, \ldots, n\}$.

2. For each $s = 1, \ldots, S$,
   - Let $\theta_s^{(1)}, \ldots, \theta_s^{(T)}$ be samples from $q(\theta|x_{A_s})$, where
   $$q(\theta|x_{A_s}) \propto p(x_{A_s}|\theta)p(\theta)^{1/S}.$$
   - Let Λ_s be an estimate of the precision matrix of $q(\theta|x_{A_s})$.

3. For $t = 1, \ldots, T$, define
$$\theta_t = \Big(\sum_{s=1}^{S} \Lambda_s\Big)^{-1} \sum_{s=1}^{S} \Lambda_s \theta_s^{(t)}.$$

4. Use $\theta_1, \ldots, \theta_T$ as approximate samples from $p(\theta|x_{1:n})$.

Consensus Monte Carlo: Pros and cons

Pros:

- Parallization: For each s, we can sample from $q(\theta|x_{A_s})$ independently on different machines, using only x_{A_s}.
- Ease of use: Can simply use existing code for sampling from the posterior, as long as you can adjust it to use the downweighted prior $q(\theta) \propto p(\theta)^{1/S}$.
- Theoretical justification: The Gaussian approximation is often reasonable for large datasets.

Cons:

- Insufficient information per shard: Each shard needs to have enough information to infer θ reasonably accurately, but often this is not the case in complex models.
- Limited to continuous parameters $\theta \in \mathbb{R}^d$.
- The Gaussian approximation breaks down in many models such as mixtures, matrix factorization models, and variable selection.

Outline

Consensus Monte Carlo: Small-sample bias issue

- Perhaps surprisingly, small-sample bias can be problematic when using consensus MC on large datasets.
- For many models, there is a bias that would normally go to zero as the data grows.
- However, since the size of each shard is not growing, each shard posterior is potentially biased.
- This bias can remain when aggregating all of the shard posteriors, making consensus MC biased.

Consensus Monte Carlo: Small-sample bias example

- Model: $Y_i \sim \mathcal{N}(\mu, 1)$ and $p(\mu) \propto 1$.
- Data: $Y_i \sim \mathcal{N}(3, 1)$ for $i = 1, \ldots, 10000$.
- Consensus MC using $S = 10$ equally weighted shards.
- Parameter of interest is μ^2.

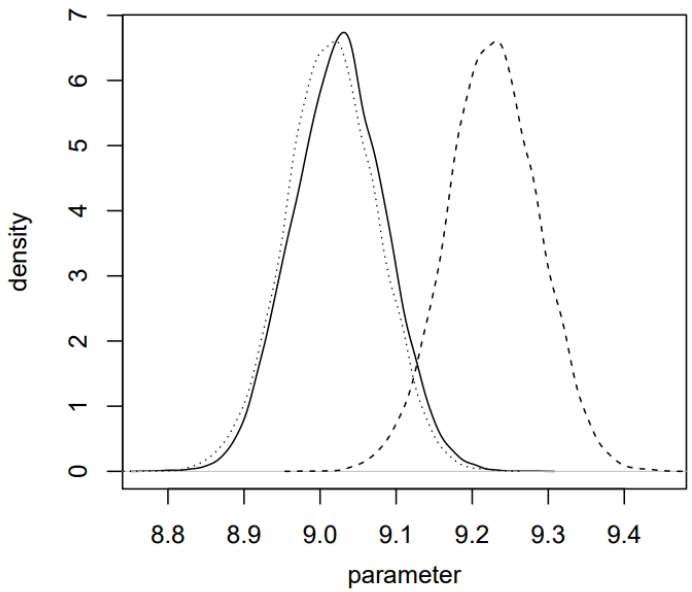


Figure 3: *Example of small sample bias. The solid line is the posterior distribution of μ^2 as described in Section 3.5. The dashed line is the consensus Monte Carlo estimate. The dotted line is the consensus Monte Carlo estimate with a jackknife bias correction applied.*

(figure from Scott et al., 2013)

Consensus Monte Carlo: Small-sample bias correction

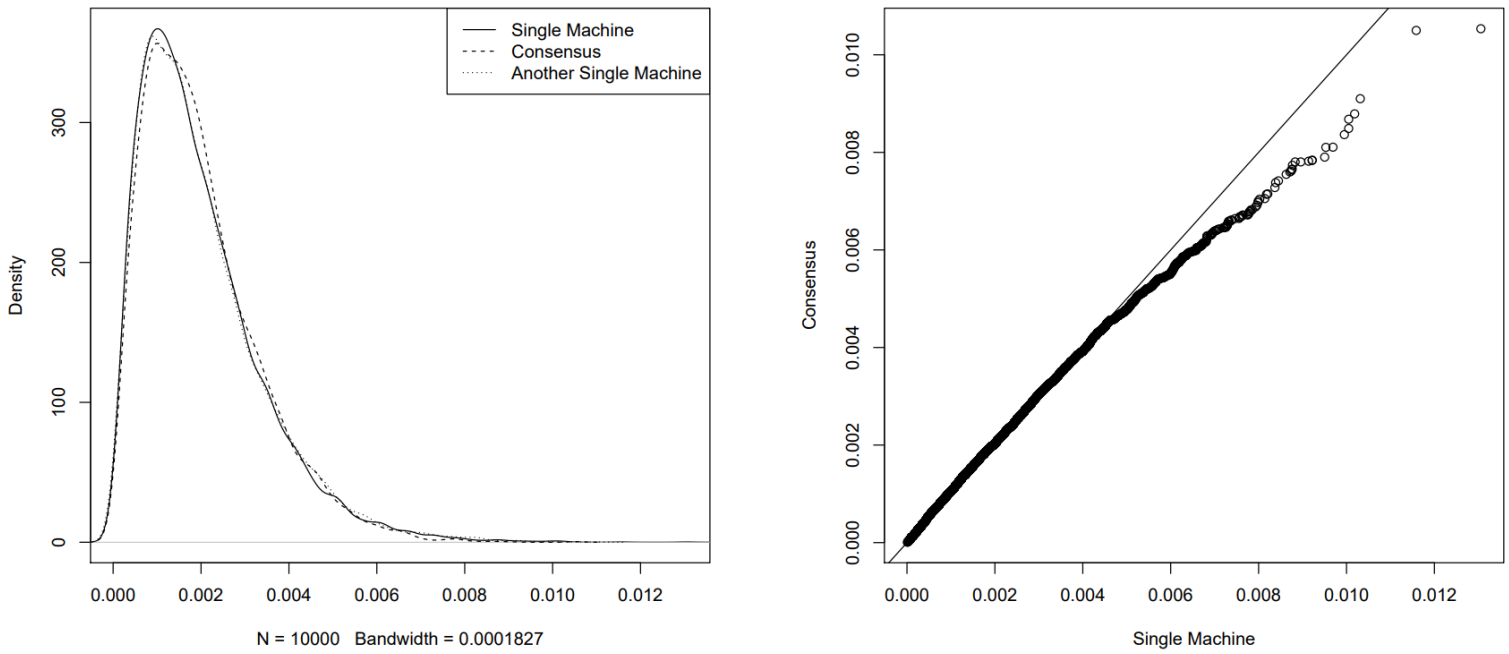
- The consensus MC paper proposes using the jackknife to correct this bias.
- Suppose the posterior mean exhibits a bias of B/n on average, for a sample of size n.
- The basic idea is to estimate the bias by comparing the posterior mean for two different sample sizes.

Outline

Consensus Monte Carlo: Binomial simulation example

- Model: $X_1, \ldots, X_n \sim \text{Bernoulli}(\theta)$ and $\theta \sim \text{Beta}(1, 1)$.
- Data: $x_1 = 1$ and $x_2 = \cdots = x_n = 0$, where $n = 1000$.
- Consensus MC using $S = 100$ equally sized shards.
- For the downweighted prior on each shard, using a $\text{Beta}(0.01, 0.01)$ prior.
- Note that this is different than using $p(\theta)^{1/S}$ as the shard prior.

Consensus Monte Carlo: Binomial example



(a) (b)

Figure 4: *(a) Posterior draws from binomial data. (b) A qq plot showing that the tails of the consensus Monte Carlo distribution in panel (a) are slightly too light.*

(figure from Scott et al., 2013)

Consensus Monte Carlo: Binomial simulation example

- Now, consider a modified version of this example with unequal shard sizes.
- Model: $X_1, \ldots, X_n \sim \text{Bernoulli}(\theta)$ and $\theta \sim \text{Beta}(1, 1)$.
- Data simulated as $X_i \sim \text{Bernoulli}(0.01)$.
- Consensus MC using 5 shards of sizes 100, 20, 20, 70, 500.
- For the downweighted prior on each shard, compare using $\text{Beta}(1, 1)$ versus $\text{Beta}(1/5, 1/5)$.

Consensus Monte Carlo: Binomial example

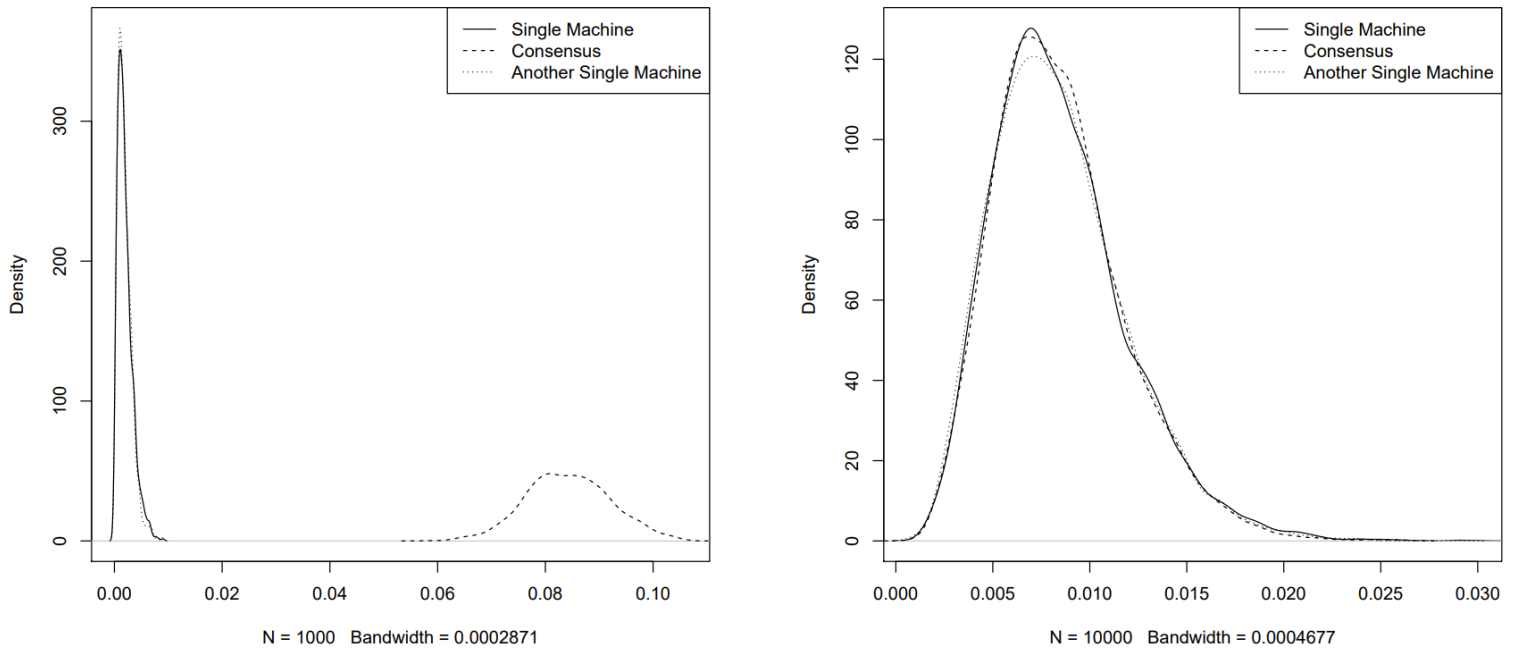


Figure 5: *The consensus Monte Carlo distribution (a) performs badly when each worker receives a uniform prior, and (b) performs well with imbalanced shards, when properly weighted.*

(figure from Scott et al., 2013)

Consensus Monte Carlo: Gaussian simulation example

- Model: $X_1, \ldots, X_n \sim \mathcal{N}(\mu, \Sigma)$.

- Data simulated as $X_i \sim \mathcal{N}(\mu, \Sigma)$ where $\mu = (1, 2, 3, 4, 5)^{\mathrm{T}}$ and

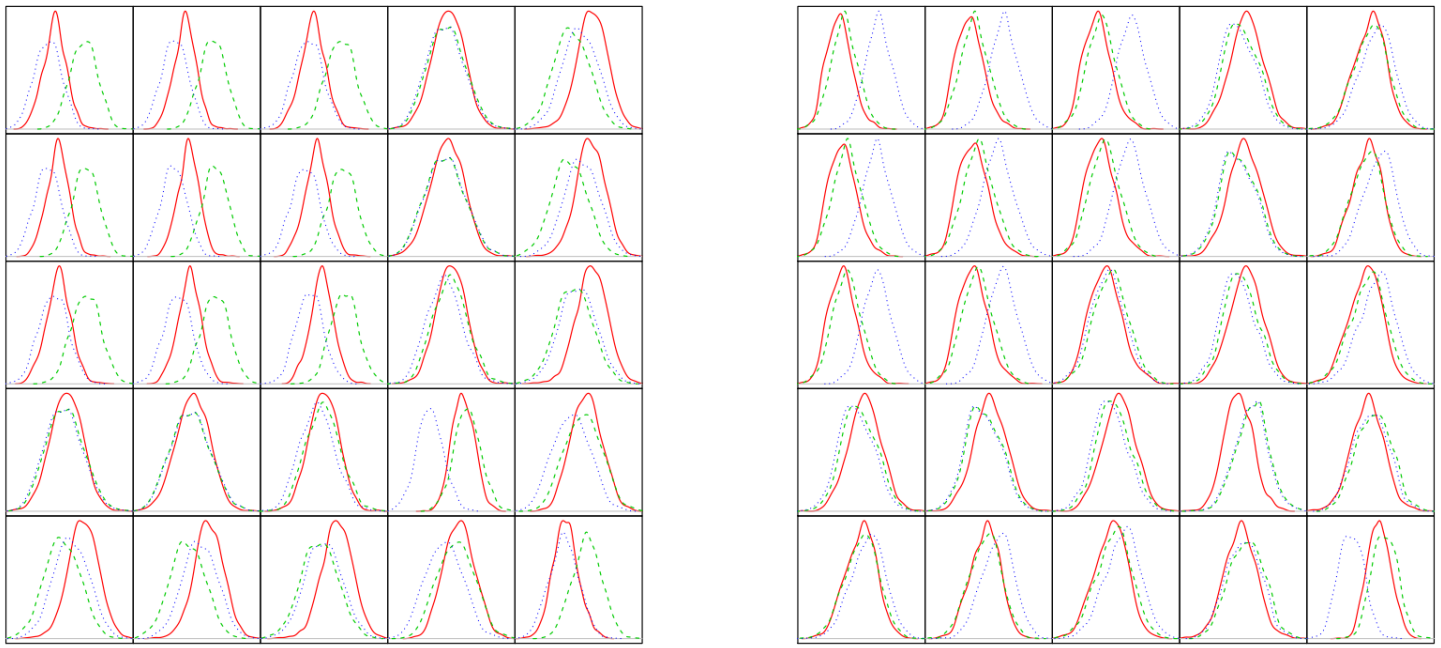
$$\Sigma = \begin{pmatrix} 1.00 & 0.99 & 0.98 & 0.00 & -0.70 \\ 0.99 & 1.00 & 0.97 & 0.00 & -0.75 \\ 0.98 & 0.97 & 1.00 & 0.00 & -0.60 \\ 0.00 & 0.00 & 0.00 & 1.00 & 0.00 \\ -0.70 & -0.75 & -0.60 & 0.00 & 1.00 \end{pmatrix}.$$

- Three scenarios: Consensus MC using $S = 100$ shards of sizes (a) 50, (b) 100, and (c) 1000 samples each.

- The consensus MC approximation for μ is nearly exact, making it less interesting as a test case.

- The posterior on Σ is more interesting.

Consensus Monte Carlo: Gaussian simulation example



(a) (b)

Figure 6: *(a) Posterior distribution of Σ based on 100 workers, with 50 observations per worker. Red (solid) line is the single machine algorithm. Green (dashed) line is the consensus Monte Carlo estimate. Blue (dotted) line is the bias corrected consensus estimate. (b) Same, but with 100 observations per worker. With 1000 observations per worker all plots overlap essentially perfectly.*

(figure from Scott et al., 2013)

Consensus Monte Carlo: Logistic regression example

- Model: $Y_i \sim \text{Bernoulli}(\text{logit}^{-1}(\beta^{\mathsf{T}} x_i))$.
- Data: $p = 5$ covariates, $n = 10000$ samples

y	n	x_1	x_2	x_3	x_4	x_5
266	2755	1	0	0	1	0
116	2753	1	0	0	0	0
34	1186	1	0	1	0	0
190	717	1	1	0	1	0
61	1173	1	0	1	1	0
37	305	1	1	1	0	0
68	301	1	1	1	1	0
119	706	1	1	0	0	0
18	32	1	0	0	0	1
13	17	1	0	1	1	1
18	24	1	0	0	1	1
8	10	1	1	0	1	1
2	2	1	1	1	0	1
7	13	1	0	1	0	1
2	2	1	1	1	1	1
3	4	1	1	0	0	1

(a)

	x_1	x_2	x_3	x_4	x_5
frequency	1	.2	.3	.5	.01
coefficient	-3	1.2	-.5	.8	3

(b)

Table 1: *(a) Data for the logistic regression in Figure 7. (b) The probability that each variable is active, and the true logistic regression coefficients used in the simulation.*

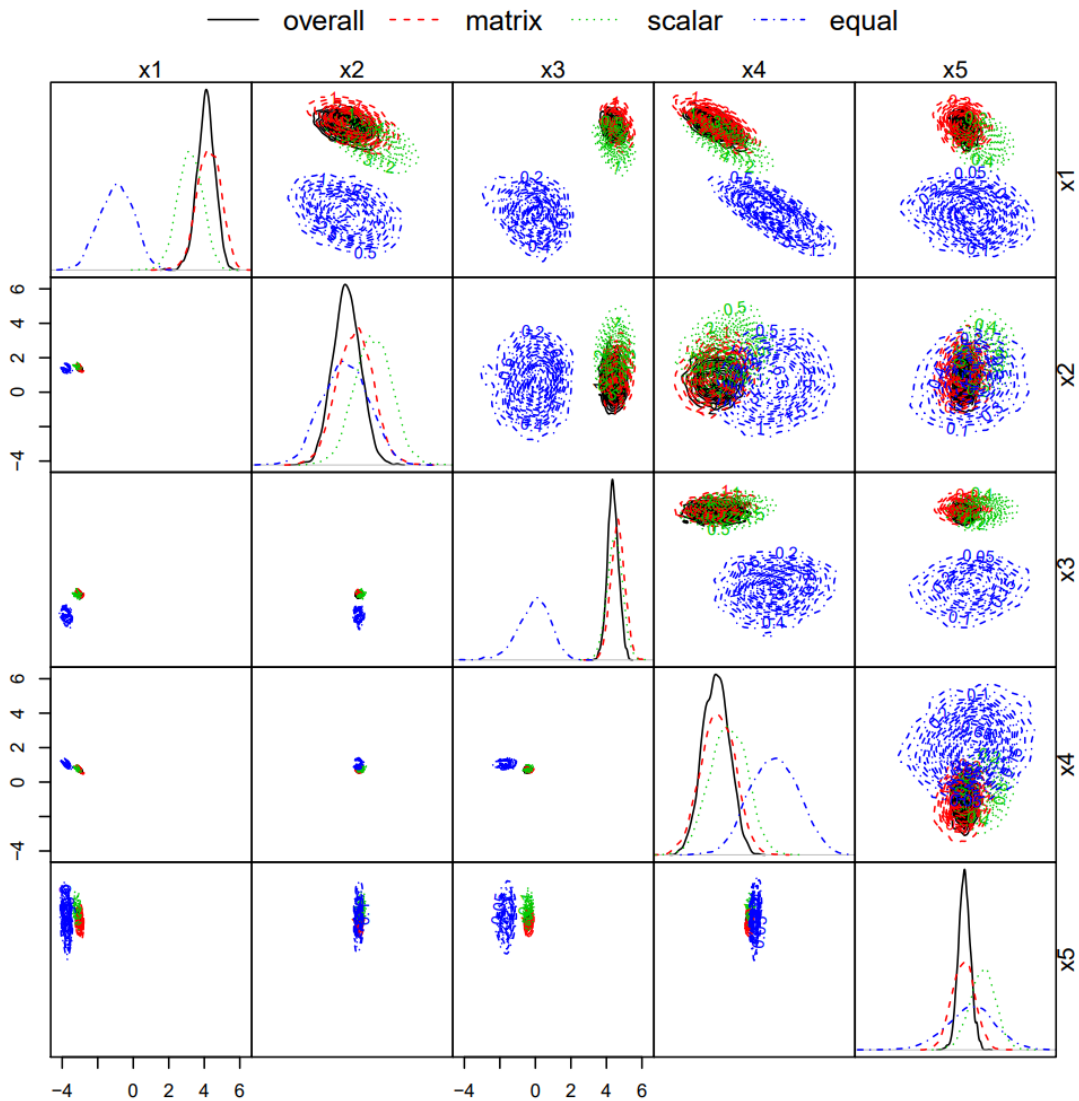
(table from Scott et al., 2013)

Consensus Monte Carlo: Logistic regression example

- Comparing 3 weighting schemes:
  - "equal": weight each shard posterior equally
  - "matrix": weight using inverse of sample covariance matrix
  - "scalar": weight using inverse of diagonal of sample covariance matrix

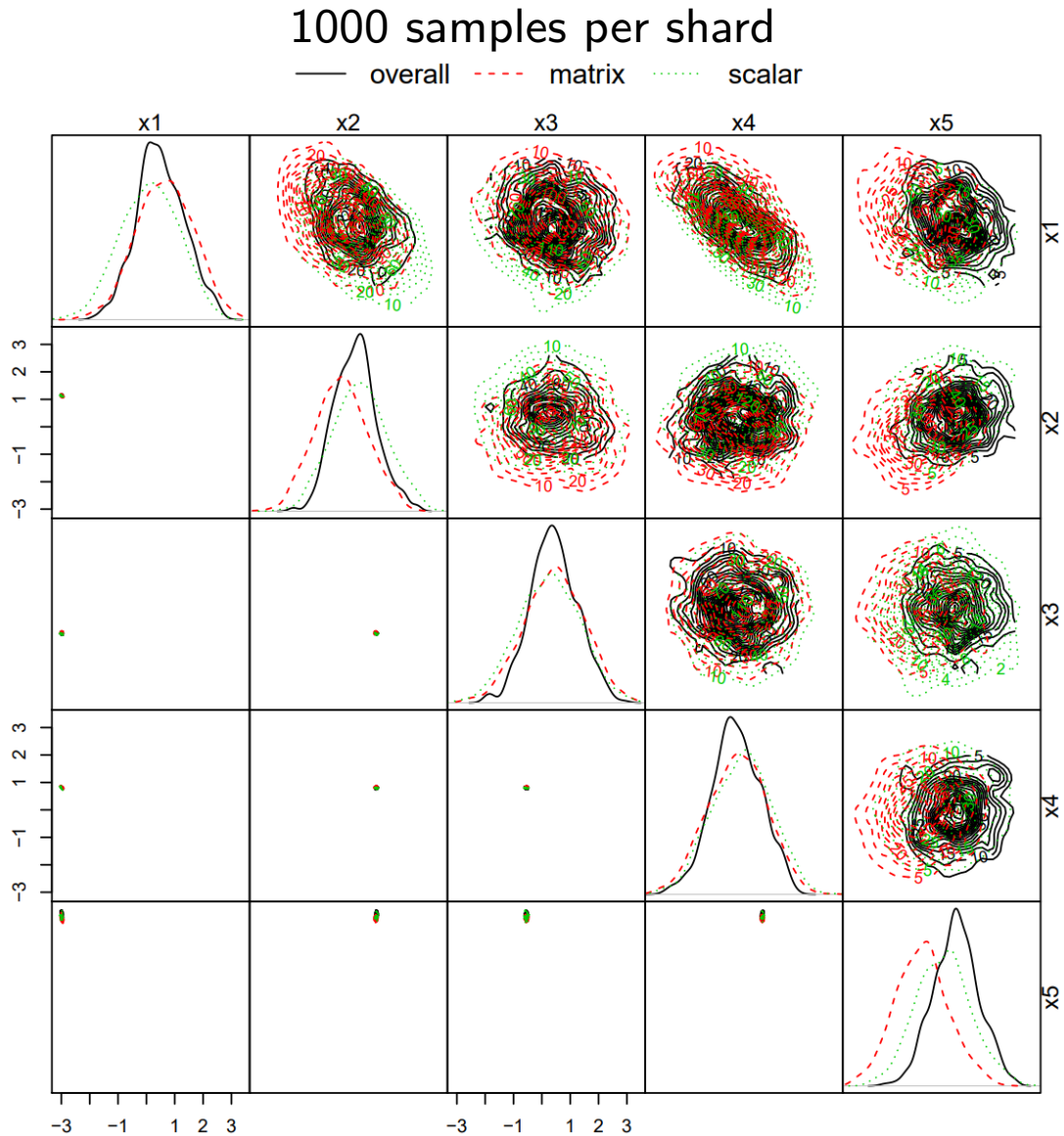
Consensus Monte Carlo: Logistic regression example

100 samples per shard



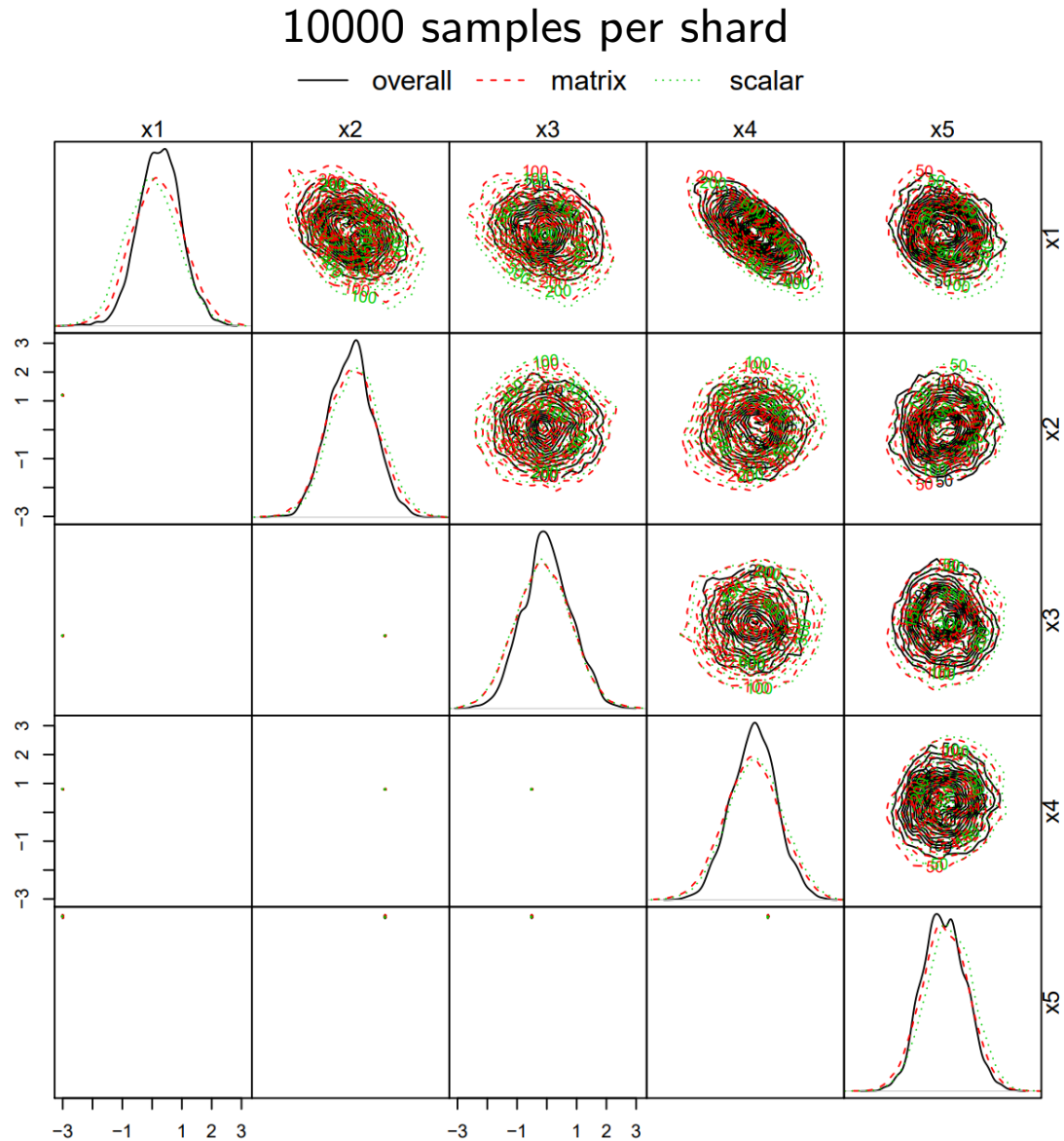
(figure from Scott et al., 2013)

Consensus Monte Carlo: Logistic regression example



(figure from Scott et al., 2013)

Consensus Monte Carlo: Logistic regression example



(figure from Scott et al., 2013)

Outline

Application: Hierarchical Poisson regression on internet ads

- Large distributed data set of internet advertising data.
- $n \approx 24$ million observations across 11000 advertisers.
- $y_{ij} =$ number of times that ad i from advertiser j was clicked.
- $E_{ij} =$ number of times that ad i from advertiser j was shown.
- $x_{ij} =$ small set of predictor variables (ad format, continuous quality score, etc.).

Application: Hierarchical Poisson regression on internet ads

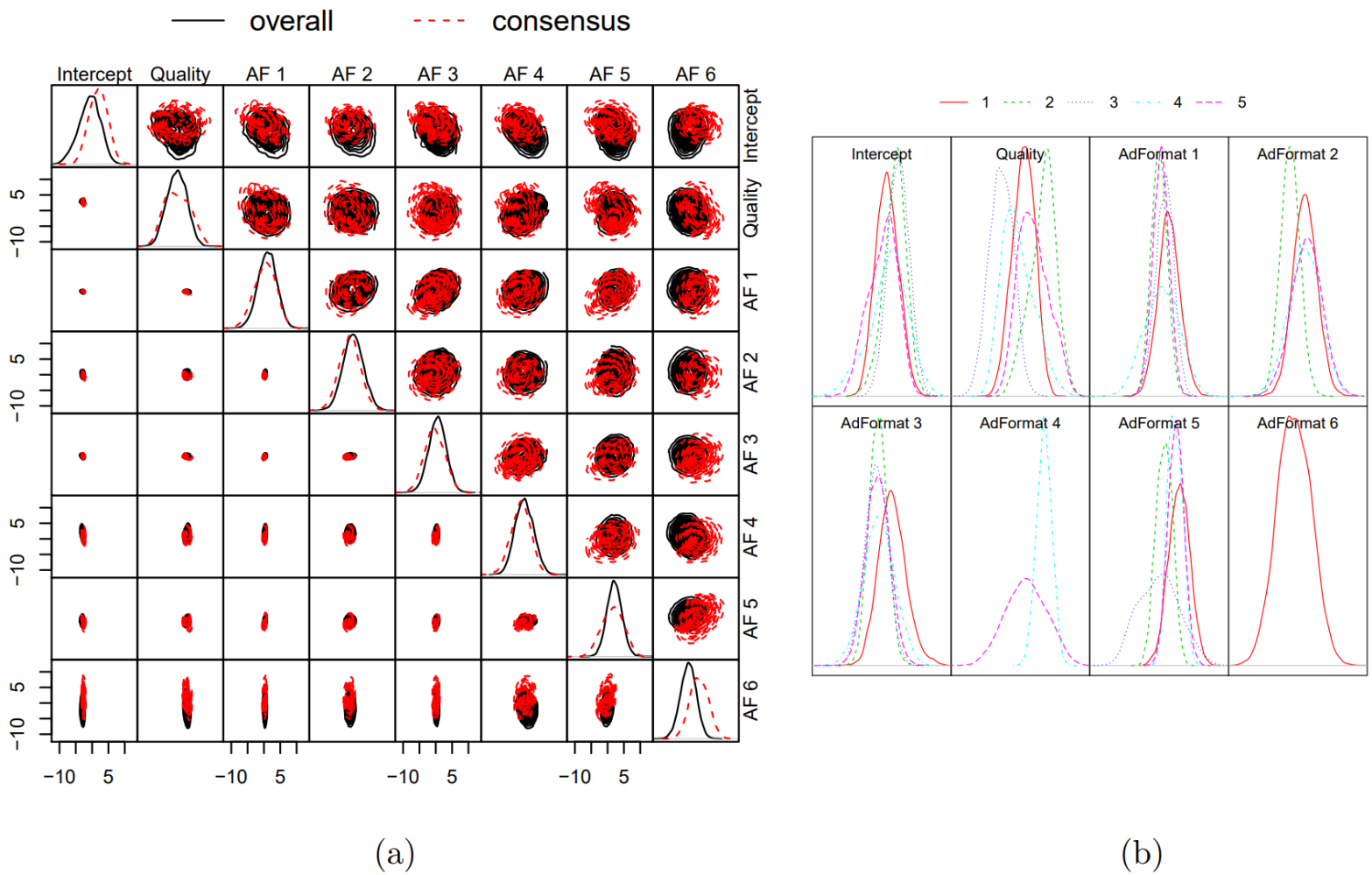
- Model:

$$
\begin{aligned}
Y_{ij} &\sim \text{Poisson}(E_{ij} \exp(\beta_j^{\mathsf{T}} x_{ij})) \\
\beta_j &\sim \mathcal{N}(\mu, \Sigma) \\
\mu &\sim \mathcal{N}(0, \Sigma/\kappa) \\
\Sigma^{-1} &\sim \text{Wishart}(I, \nu).
\end{aligned}
$$

- Consensus Monte Carlo using 867 shards with between 10000 to 50000 observations each. The scalar precision weights were used.

- For comparison, the full posterior was computed using only 5 shards because it took very long to compute.

Application: Hierarchical Poisson regression on internet ads



(a) (b)

Figure 9: *Posterior draws of μ based on the first 5 shards described in Section 4.4. (a) Posterior draws from the single-machine and consensus Monte Carlo algorithms. (b) Draws from the five worker machines.*

(figure from Scott et al., 2013)

Application: Hierarchical Poisson regression on internet ads

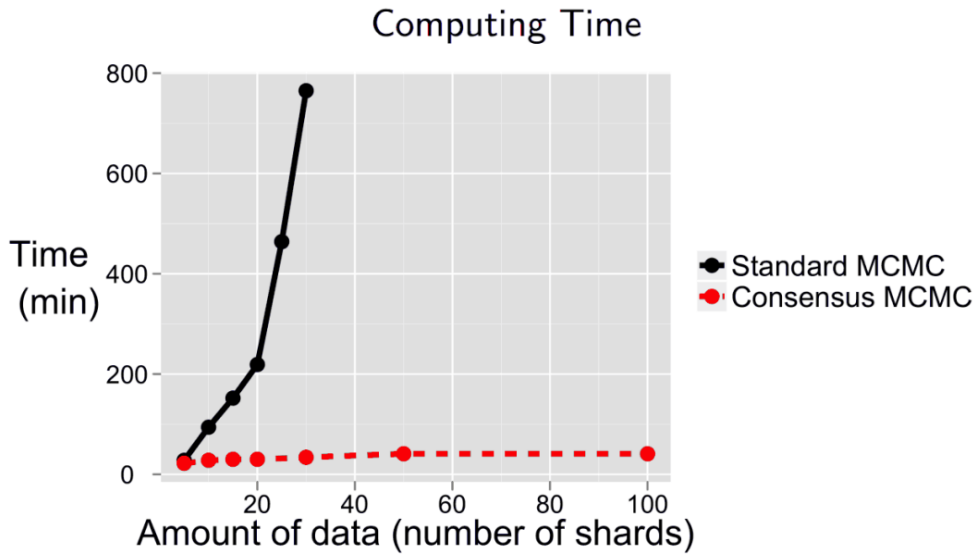


Figure 10: *Time required to complete 10,000 MCMC draws with different numbers of shards under the single machine and consensus Monte Carlo algorithm.*

(figure from Scott et al., 2013)

References and supplements

- Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., & McCulloch, R. E. (2016). Bayes and big data: The consensus Monte Carlo algorithm. International Journal of Management Science and Engineering Management, 11(2), 78-88.