Subsampling Strategies for Speeding Up MCMC

Mattias Villani

Division of Statistics and Machine Learning
Department of Computer and Information Science
Linköping University

Outline of the talk

- Bayesian inference
- MCMC
- Distributed MCMC
- MCMC with unbiased likelihood estimators
- MCMC with data subsampling

Credits and Disclaimer

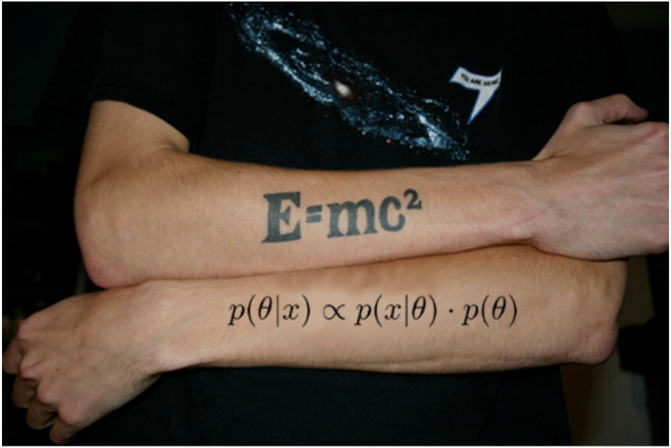
Disclaimer: I will only cover a small (biased) subset of methods. This area is under **active development**.

My own activity in this area is joint work with my PhD student **Matias Quiroz** and **Robert Kohn**.

Why should you care?

- **Big data**. Data sets are getting bigger and bigger.
- **Bayesian inference** is the way to go.
- Bayesian inference is usually implemented using MCMC.
- **MCMC** can be very **slow** on large data sets.
- The **likelihood can be costly** to evaluate (also on small data).
- Approximate solutions (**VB**, **ABC**, **INLA**, **EP**, **BO**,...) generally come without bounds on the error.

Great theorems make great tattoos



Bayesian inference - advantages

- **Probabilistic**. Nothing is ad hoc or black magic. It's a **theory**.
- Result in terms of complete probability distributions, e.g. **predictive distribution**.
- **Marginalization** of nuisance parameters. Predictive distributions include parameter uncertainty.
- Natural way to **handle model uncertainty** and selection. **Model averaging**.
- Natural connection to **decision making**. Maximize posterior expected utility.
- **Prior information** helps. **Flexible** ML models are impossible without **smoothness priors**.

MCMC - THE BASIC IDEA

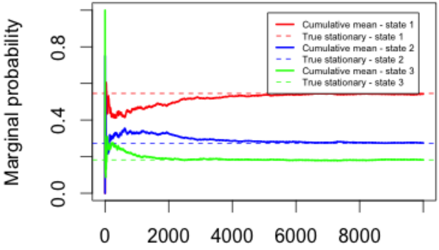
- Explore complicated joint posterior distributions $p(\theta|\mathbf{y})$ by **simulation**.
- Set up **Markov chain** for θ with $p(\theta|\mathbf{y})$ as **stationary distribution**.
- Initialize $\theta^{(0)}$. Simulate Markov chain $\theta^{(i)}|\theta^{(i-1)}$ for $i = 1, 2, ..., N$.
- The Markov chain eventually forgets its initial value $\theta^{(0)}$ and starts to produce draw from its stationary (invariant) distribution $p(\theta|\mathbf{y})$.
- Draw are **autocorrelated ...**
- ...but sample averages $(N^{-1}\sum_{i=1}^{N}\theta^{(i)})$ still converge to posterior expectations $(E(\theta|\mathbf{y}))$.
- High autocorrelation means fewer **effective draws**

$$Var(\bar{\theta}) = \frac{\sigma^2}{N}\left(1 + 2\sum_{k=1}^{\infty}\rho_k\right)$$

MCMC - THE BASIC IDEA

- Example: Discrete $\theta \in \{0, 1, 2\}$. $p(\theta|\mathbf{y}) = (0.545, 0.272, 0.181)$.
- Markov **transition matrix**

$$P = \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.2 & 0.6 & 0.2 \\ 0.3 & 0.3 & 0.4 \end{pmatrix}$$



- Markov chains with **continuous state space** when θ is continuous. **Transition kernel**: $Pr\left(\theta^{(i-1)} \rightarrow \text{Region in } \theta\text{-space}\right)$.

THE METROPOLIS-HASTINGS ALGORITHM

- Initialize $\theta^{(0)}$ and iterate for $i = 1, 2, \ldots$

  1. Sample $\theta_p \sim q\left(\cdot|\theta^{(i-1)}\right)$ (the **proposal distribution**)
  2. Compute the **acceptance probability**

  $$\alpha = \min\left(1, \frac{p(\mathbf{y}|\theta_p)p(\theta_p)}{p(\mathbf{y}|\theta^{(i-1)})p(\theta^{(i-1)})} \frac{q\left(\theta^{(i-1)}|\theta_p\right)}{q\left(\theta_p|\theta^{(i-1)}\right)}\right)$$

  3. With probability α set $\theta^{(i)} = \theta_p$ and $\theta^{(i)} = \theta^{(i-1)}$ otherwise.

Random walk Metropolis algorithm

- Initialize $\theta^{(0)}$ and iterate for $i = 1, 2, \ldots$

  1. Sample $\theta_p \sim N\left(\theta^{(i-1)}, c \cdot \Sigma\right)$ (the **proposal distribution**)
  2. Compute the **acceptance probability**

  $$\alpha = \min\left(1, \frac{p(\mathbf{y}|\theta_p)p(\theta_p)}{p(\mathbf{y}|\theta^{(i-1)})p(\theta^{(i-1)})}\right)$$

  3. With probability α set $\theta^{(i)} = \theta_p$ and $\theta^{(i)} = \theta^{(i-1)}$ otherwise.

Distributed MCMC

- **Big data** = when data don't fit on a single machine. **Multi-machine**.
- **Distributed computing**. Move computation to the data. Map-Reduce style. Minimal communication.
- Assuming data on separate machines are **independent** given θ:

$$p(\theta|\mathbf{y}) = \prod_{m=1}^{M} p(\mathbf{y}_m|\theta)p(\theta)^{1/M}$$

- **Consensus Monte Carlo** on M machines [1]:
  - **Partition the data** $\mathbf{y}$ into $\mathbf{y}_1, ..., \mathbf{y}_M$.
  - **Run** M **separate Monte Carlo** algorithms to sample $\theta_m^{(i)}, ..., \theta_m^{(N)}$ from
  $$p_m(\theta|\mathbf{y}_m) \propto p(\mathbf{y}_m|\theta)p(\theta)^{1/M}, \text{ for } m = 1, ..., M.$$
  - **Combine draws** across machines using weighted averages:
  $$\theta^{(i)} = \left(\sum_m W_m\right)^{-1}\left(\sum_m W_m\theta_m^{(i)}\right)$$

.

Distributed MCMC, cont.

- **Problems** with consensus methods:
  - Only guaranteed to be correct if all $p(\theta|\mathbf{y}_m)$ are Gaussian.
  - Only for continuous θ.
  - Improper priors are problematic.
  - Cannot be applied when θ can change in dimension, e.g. Dirichlet process mixtures.
  - Risk of collapsing multimodal posteriors.

- Recent **extensions**:
  - Combining kernel density estimates rather than draws. [2] Hard when θ is not low-dimensional.
  - Passing low-dim sufficient statistics between machines at runtime.
  - Weierstrass transforms [3]
  - Median of subset posteriors [4]

MCMC WITH AN UNBIASED LIKELIHOOD ESTIMATOR

- The **full likelihood** $p(\mathbf{y}|\theta)$ is **intractable** or very **costly to evaluate**.
- **Unbiased estimator** $\hat{p}(\mathbf{y}|\theta, u)$ of the likelihood is available

$$\int \hat{p}(\mathbf{y}|\theta, u)p(u)du = p(\mathbf{y}|\theta)$$

- $u \sim p(u)$ are auxilliary variables used to compute $\hat{p}(\mathbf{y}|\theta, u)$.
- **Importance sampling/particle filters** for latent variable (x) models

$$p(\mathbf{y}|\theta) = \int p(\mathbf{y}, \mathbf{x}|\theta)dx = \int \frac{p(\mathbf{y}, \mathbf{x}|\theta)}{q_\theta(\mathbf{x})} q_\theta(\mathbf{x})d\mathbf{x}$$

$$\hat{p}(\mathbf{y}|\theta, u) = \frac{1}{m}\sum_{k=1}^{m} \frac{p(\mathbf{y}, \mathbf{x}^{(k)}|\theta)}{q_\theta(\mathbf{x}^{(k)})} \text{ where } \mathbf{x}^{(i)} \overset{iid}{\sim} q_\theta(\cdot)$$

- **Subsampling**: u are indicators for selected observations.
- Let m be the number of u's (particles/subsample size).

MCMC WITH A UNBIASED LIKELIHOOD ESTIMATOR

- But is it OK to use a noisy estimate $\hat{p}(\mathbf{y}|\theta, u)$ of the likelihood?
- The joint density

$$\tilde{p}(\theta, u|\mathbf{y}) = \frac{\hat{p}(\mathbf{y}|\theta, u)p(\theta)p(u)}{p(\mathbf{y})}$$

has the correct marginal density $p(\theta|y)$ if $\hat{p}(y|\theta, u)$ is **unbiased**

$$p(y|\theta) = \int \hat{p}(\mathbf{y}|\theta, u)p(u)du$$

- This is easily seen from

$$\int \tilde{p}(\theta, u|\mathbf{y})du = \frac{p(\theta)}{p(y)} \int \hat{p}(\mathbf{y}|\theta, u)p(u)du = \frac{p(\theta)p(\mathbf{y}|\theta)}{p(\mathbf{y})} = p(\theta|\mathbf{y})$$

The Pseudo-Marginal MH algorithm

- Initialize $\left(\theta^{(0)}, u^{(0)}\right)$ and iterate for $i = 1, 2, \ldots$

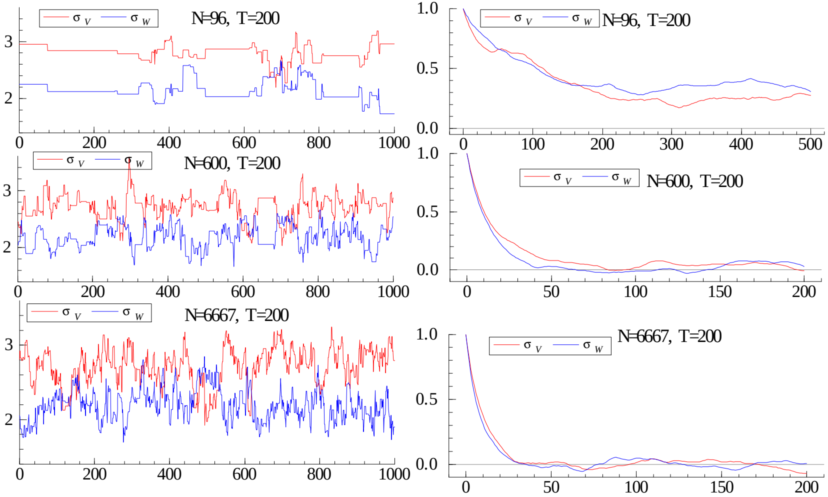
  1. Sample $\theta_p \sim q\left(\cdot|\theta^{(i-1)}\right)$ and $u_p \sim p_\theta(u)$ to obtain $\hat{p}(y|\theta_p, u)$
  2. Compute the **acceptance probability**

  $$\alpha = \min\left(1, \frac{\hat{p}\left(\mathbf{y}|\theta_p, u_p\right) p(\theta_p)}{\hat{p}\left(\mathbf{y}|\theta^{(i-1)}, u^{(i-1)}\right) p(\theta^{(i-1)})} \frac{q\left(\theta^{(i-1)}|\theta_p\right)}{q\left(\theta_p|\theta^{(i-1)}\right)}\right)$$

  3. With probability α set $\left(\theta^{(i)}, u^{(i)}\right) = (\theta_p, u_p)$ and $\left(\theta^{(i)}, u^{(i)}\right) = \left(\theta^{(i-1)}, u^{(i-1)}\right)$ otherwise.

- This MH has $\tilde{p}(\theta, u|\mathbf{y})$ as stationary distribution with marginal $p(\theta|\mathbf{y})$.
- This result holds **irrespective of the variance** of $\hat{p}(y|\theta, u)$.
- **It's OK to replace the likelihood with an unbiased estimate! [5]**

The number of particles in a state-space model



Optimal m - keep the variance around 1

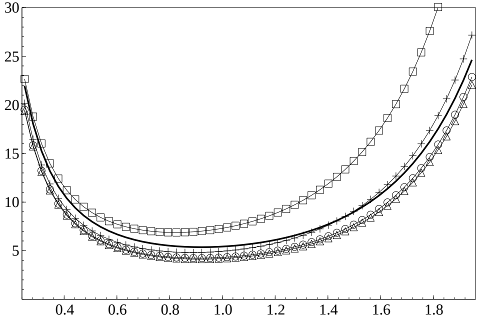
- **Large** m $\Rightarrow$ **costly** $\hat{p}(\mathbf{y}|\theta, u)$, but **efficient MCMC**.
- **Small** m $\Rightarrow$ **inexpensive** $\hat{p}(\mathbf{y}|\theta, u)$, but **inefficient MCMC**.
- Define the estimation error

$$z = \ln \hat{p}(y|\theta, u) - \ln p(y|\theta, u)$$

- Assumptions:
  - z is independent of θ
  - z is Gaussian
- **Optimal** m:
  - For good proposals for θ, use $\sigma_z \approx 1$.
  - For bad proposals for θ, use $\sigma_z \approx 1.7$.
  - Targeting $\sigma_z \approx 1.7$ when $\sigma_z \approx 1$ is optimal is much worse than targeting $\sigma_z \approx 1$ when $\sigma_z \approx 1.7$ is optimal.
  - A good compromise: $\sigma_z \approx 1.2$. [6] [7]

Optimal m - keep the variance around 1

- Computing time as a function of σ_z



- Squares - $IF = 1$
- Crosses - $IF = 4$
- Circles - $IF = 20$
- Triangles - $IF = 80$
- Solid - Perfect proposal

Estimating the likelihood by subsampling

- **Log-likelihood** for independent observations:

$$\ell(\theta) = \ln p(y_1, ..., y_n|\theta) = \sum_{i=1}^{n} \ln p(y_i|\theta)$$

- **Log-likelihood contribution** of ith observation:

$$\ell_i(\theta) = \ln p(y_i|\theta)$$

- Applicable as long as we have **independent pieces of data**:
  - **Longitudinal data**. Subjects are independent, the observations for a given subject are not.
  - **Time series** with kth order Markov structure: $y_t|y_{t-1}, ..., y_{t-k}$.
  - **Textual data**. Documents are independent. Words within documents are not.
- Estimating the log-likelihood (a sum) is like estimating a population total. **Survey sampling**.

Simple random sampling does not work

- **Simple random sampling** (**SRS**) with replacement. At the jth draw:

$$\Pr(u_j = k) = \frac{1}{n},\; k = 1, ..., n \text{ and } j = 1, ..., m$$

- Let $u = (u_1, ..., u_m)$ record the sampled observations.
- **Unbiased estimator** of the **log-likelihood**

$$\hat{\ell}_{SRS}(\theta) = \frac{n}{m} \sum_{j=1}^{m} \ell_{u_j}(\theta)$$

- $\hat{\ell}_{SRS}(\theta)$ is **extremely variable**, even when the sampling fraction m/n is large.
- **PMCMC gets stuck** as soon as $\hat{\ell}_{SRS}(\theta)$ is sampled in the extreme right tail.
- Sampling without replacement does not help.

PPS sampling

- The problem with SRS is that all elements have the same inclusion probability, $p_k = \Pr(u_j = k) = \frac{1}{n}$.
- Some $\ell_k(\theta)$ are much larger than other, and the risk of missing those in the sample inflates the variance of $\hat{\ell}_{SRS}(\theta)$.
- **Probability proportional-to-size** (**PPS**) sampling uses a **size proxy** $w_k(\theta)$ to sample large elements with larger probabilities

$$p_k \propto w_k(\theta)$$

- Hansen-Hurwitz estimator

$$\hat{\ell}_{HH}(\theta) = \frac{1}{m} \sum_{j=1}^{m} \frac{\ell_{u_j}(\theta)}{p_{u_j}}$$

- $\hat{\ell}_{HH}(\theta)$ is unbiased for the **log-likelihood** $\ell(\theta)$. Easy-to-compute unbiased estimator of $\mathrm{Var}(\hat{\ell}_{HH}(\theta))$.
- m can be easily chosen **adaptively** at runtime.[8]

Ways to obtain the proxy for $\ln p(y_i|\theta)$

- **Surrogate models**
  - Problem-specific approximation to the data density.
- **Cruder numerical solution** of the likelihood
  - Larger tolerance in numerical integration
  - Larger steps length when solving PDEs
  - Fewer Newton steps in optimization
- **Surface fit** (thin-plate spline or Gaussian process)
  - Use small subset of data to fit surface $(y, \mathbf{x}) \to \ell(\theta; y, \mathbf{x})$
  - Predict $\ell(\theta; y, \mathbf{x})$ for remaining observations using the fitted surface.
  - Tune surface smoothness parameters before MCMC. Predictions by matrix-vector product. Fast.

Bias-correction

- Consider **subsampling estimators of the log-likelihood** of the form

$$\hat{\ell}_{HH}(\theta) = \frac{1}{m}\sum_{j=1}^{m}\frac{\ell_{u_j}(\theta)}{p_{u_j}}$$

  where $p_k = \Pr(u = k)$ are the selection probabilities.
- Let z denote the **error** in the log-likelihood estimate:

$$\hat{\ell}_{HH}(\theta) = \ell(\theta) + z$$

  and $\sigma_z^2 = \mathrm{Var}(z)$.
- Assume $z \sim N(0, \sigma_z^2)$ and that σ_z^2 known. Then

$$\exp\left[\hat{\ell}_{HH}(\theta) - \sigma_z^2/2\right]$$

  is unbiased for the likelihood.
- What if z is not Gaussian and σ_z^2 is estimated unbiasedly by

$$\hat{\sigma}_z^2 = \frac{1}{m(m-1)}\sum_{j=1}^{m}\left(\frac{\ell_{u_j}(\theta)}{p_{u_j}} - \hat{\ell}_{HH}(\theta)\right)^2$$

MCMC WITH A BIASED LIKELIHOOD ESTIMATOR

- **Biased likelihood estimator**:

$$\hat{p}_m(y|\theta, u) = \exp\left[\hat{\ell}_{HH}(\theta) - \hat{\sigma}_z^2/2\right]$$

- Define:
  - Perturbed likelihood: $p_m(y|\theta) = \int \hat{p}_m(y|\theta, u)p(u)du$
  - Perturbed marginal data density: $p_m(y) = \int p_m(y|\theta)p(\theta)d\theta$
  - **Perturbed posterior**: $p_m(\theta|y) = p_m(y|\theta)p(\theta)/p_m(y)$.
- A PMCMC scheme targeting

$$\tilde{\pi}_m(\theta, u|\mathbf{y}) = \frac{\hat{p}_m(\mathbf{y}|\theta, u)p(\theta)p(u)}{p(\mathbf{y})}$$

has $p_m(\theta|y)$ as invariant distribution.

THEOREM

$$\frac{|p_m(\theta|y) - p(\theta|y)|}{p(\theta|y)} \leq \frac{C}{\sqrt{m}}$$

Application: Firm bankruptcy

- **Discrete-time Weibull survival model**.
- The **hazard** probability for firm i at time period j

$$h_t(x_{ij}) \quad = \quad 1 - \exp\left(-\lambda\left(t_{ij}^{\rho} - t_{i(j-1)}^{\rho}\right)\right)$$

where

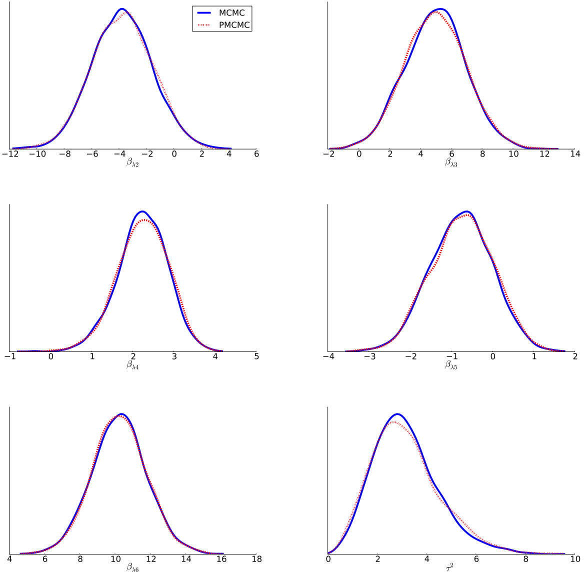
$$\log(\lambda) = \gamma_i + x_{ij}^T\beta_\lambda \text{ and } \log(\rho) = x_{ij}^T\beta_\rho, \text{ with } \gamma_i \overset{iid}{\sim} N(0, \tau^2)$$

- Five explanatory variables (profits, leverage, liquidity, firm age, sales)
- The **log-likelihood** for n firms

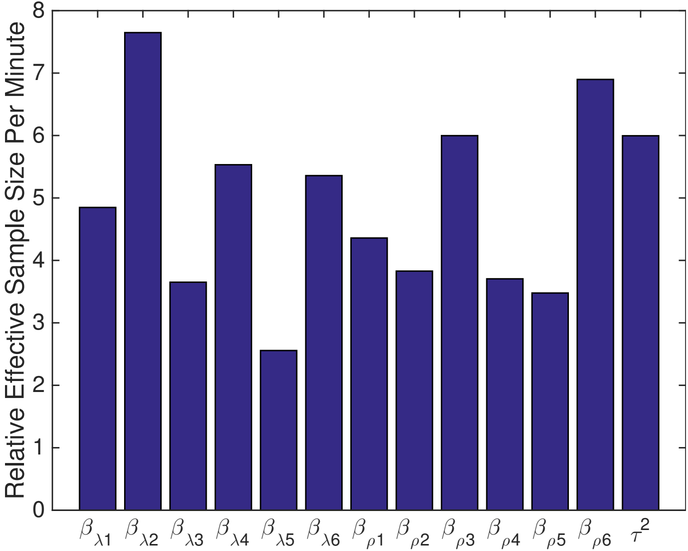
$$\log p(\mathbf{y}_1, ..., \mathbf{y}_n | \beta_\gamma, \beta_\rho) \quad = \quad \sum_{i=1}^{n} \log\left(\int p(\mathbf{y}_i | \beta_\gamma, \beta_\rho, \gamma_i) p(\gamma_i) d\gamma_i\right)$$

where $\mathbf{y}_i = (y_{i1}, \ldots y_{in_i})$.
- $n = 2000$. $m = 20$.

WEIBULL REGRESSION - PPS SAMPLING APPROACH



WEIBULL REGRESSION - PPS SAMPLING APPROACH



The difference estimator

- Let $w_k(\theta)$ be an cheap approximation of $\ell_k(\theta)$. Trivial decomposition:

$$\begin{aligned}\ell(\theta) &= \sum_{k \in F} w_k(\theta) + \sum_{k \in F} \left[\ell_k(\theta) - w_k(\theta)\right] \\ &= \sum_{k \in F} w_k(\theta) + \sum_{k \in F} d_k(\theta)\end{aligned}$$

- $\sum_{k \in F} w_k(\theta)$ is known.
- $\sum_{k \in F} d_k(\theta)$ can be estimated by sampling like any population total.
- If $w_k(\theta)$ is a decent proxy for $\ell_k(\theta)$, the **differences** $d_k(\theta)$ should be **roughly equal in size**. SRS works!

- **PPS**: size proxy $w_k(\theta)$ is used to sample large elements.
- **Difference estimator**: $w_k(\theta)$ is used to normalize the size of the elements. No fancy sampling scheme needed. [9]

Obtaining the proxy by data clustering

- Idea: $\ell(\theta; y, \mathbf{x})$ and $\ell(\theta; y', \mathbf{x}')$ are likely to be similar when $(y, \mathbf{x})$ and $(y', \mathbf{x}')$ are close.

- Approximate $\ell(\theta; y, \mathbf{x}) \approx \ell(\theta; y_c, \mathbf{x}_c)$ where $(y_c, \mathbf{x}_c)$ is the **nearest cluster centroid**.

- Even better: use **Taylor expansion** around $\ell(\theta; y_c, \mathbf{x}_c)$ as proxy.

- Difference estimator can be computed using computations only at the C centroids. Scalable.

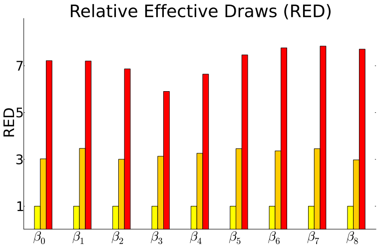
- Curse of dimensionality can be dealt with by **dimension reduction** (clustering in PCA space). [9]

Application: multi-period logistic regression

- Predicting firm bankruptcy using multi-period logistic regression

$$p(y_k|x_k,\beta) = \left(\frac{1}{1+\exp(x_k^T\beta)}\right)^{y_k}\left(\frac{1}{1+\exp(-x_k^T\beta)}\right)^{1-y_k}$$

- 0.5 million firms with a total of 4.5 million firm-year observations.



- Yellow = MCMC on full data set.
- Orange = Difference estimator, updating u every draw.
- Red = Orange = Difference estimator, updating u every 50th draw.

IDA Machine Learning Seminars

liu.se/machinelearning

Linköping University

Department of Computer and Information Science

Swedish web site

Search | Search IDA.LiU.se | Search

A — Z

LiU ▶ IDA ▶ Research ▶ Machine Learning ▶ Seminars ▶ MoreSeminars ▶ Machine Learning Seminars

Machine Learning

- **People**
- **Seminars**
- **Publications**
- **Education**

IDA Machine Learning Seminars - Fall 2014

Wednesday, September 17, 3.15 pm, 2014.

Sequential Decision Making: Experiment Design, Big Data and Reinforcement Learning
Christos Dimitrakakis, Computer Science and Engineering at Chalmers University of Technology.
Abstract: Gone are the days when statisticians used to work with fixed, laboriously compiled and labelled datasets. Nowadays data collection is frequently active and therefore must be adaptive. This talk will give an overview of the field of sequential decision making and how it relates to experiment design, active learning and the general problem of reinforcement learning. The main technical problems encountered are how to plan and learn efficiently. The first problem requires efficient optimisation algorithms, while the second requires good models that are easy to update online and can deal with large amounts of data.
Location: **Visionen**
Organizer: **Mattias Villani**

Wednesday, October 15, 3.15 pm, 2014.

Inducing Semantic Representations from Text with Little or No Supervision
Ivan Titov, Institute for Logic, Language and Computation at University of Amsterdam.
Abstract: Inducing meaning representations from text is one of the key objectives of NLP. Most existing statistical semantic analyzers rely on large human-annotated datasets, which are expensive to create and exist only for a very limited number of languages. Even then, they are not very robust, cover only a small proportion of semantic constructions appearing in the labeled data, and are domain-dependent. We investigate Bayesian models which do not use any labeled data but induce semantic representations from unannotated texts. Unlike semantically-annotated data, unannotated texts are plentiful and available for many languages and many domains which makes our approach particularly promising. We show that these models induce linguistically-plausible semantic representations, significantly outperform current state-of-the-art approaches, and yield

- S. L. Scott, A. W. Blocker, F. V. Bonassi, H. A. Chipman, E. I. George, and R. E. McCulloch, "Bayes and big data: The consensus monte carlo algorithm," in *EFaBBayes 250 conference*, vol. 16, 2013.
- W. Neiswanger, C. Wang, and E. Xing, "Asymptotically exact, embarrassingly parallel mcmc," *arXiv preprint arXiv:1311.4780*, 2013.
- X. Wang and D. B. Dunson, "Parallelizing mcmc via weierstrass sampler," *arXiv preprint arXiv:1312.4605*, 2013.
- S. Minsker, S. Srivastava, L. Lin, and D. B. Dunson, "Robust and scalable bayes via a median of subset posterior measures," *arXiv preprint arXiv:1403.2660*, 2014.
- C. Andrieu and G. O. Roberts, "The pseudo-marginal approach for efficient monte carlo computations," *The Annals of Statistics*, pp. 697–725, 2009.
- M. K. Pitt, R. dos Santos Silva, P. Giordani, and R. Kohn, "On some properties of markov chain monte carlo simulation methods based on

the particle filter," *Journal of Econometrics*, vol. 171, no. 2, pp. 134–151, 2012.

A. Doucet, M. Pitt, G. Deligiannidis, and R. Kohn, "Efficient implementation of markov chain monte carlo when using an unbiased likelihood estimator," *forthcoming in Biometrika*, 2012.

M. Quiroz, M. Villani, and R. Kohn, "Speeding up mcmc by efficient data subsampling," *arXiv preprint arXiv:1404.4178*, 2014.

M. Quiroz, M. Villani, and R. Kohn, "Scalable mcmc for large data problems using data subsampling and the difference estimator," *Manuscript*, 2015.