SPEEDING UP MCMC BY EFFICIENT DATA SUBSAMPLING

Mattias Villani

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

Credits and Caveats

- **Joint work with**
  - **Matias Quiroz**, Stockholm University and Sveriges Riksbank
  - **Robert Kohn**, University of New South Wales, Sydney

- Work in progress! Results are preliminary.

Background and motivation

- **MCMC** - main tool for Bayesian computations for decades.
- Painfully **slow on large datasets,** especially when the **likelihood is costly** to evaluate.
- How big is **Big Data**? Depends on model complexity.

BACKGROUND AND MOTIVATION

- **MCMC** - main tool for Bayesian computations for decades.
- Painfully **slow on large datasets,** especially when the **likelihood is costly** to evaluate.
- How big is **Big Data**? Depends on model complexity.
- **Approximate methods** abound, all with drawbacks.
  - **Variational Bayes** (VB) [bad approx of posterior spread etc]
  - **Approximate Bayesian Computation** (ABC) [summary statistics?]
  - **Integrated Nested Laplace Approximation** (INLA) [applicable?]
- **Sequential Monte Carlo** (SMC)

Background and motivation

- **MCMC** - main tool for Bayesian computations for decades.
- Painfully **slow on large datasets,** especially when the **likelihood is costly** to evaluate.
- How big is **Big Data**? Depends on model complexity.
- **Approximate methods** abound, all with drawbacks.
  - **Variational Bayes** (VB) [bad approx of posterior spread etc]
  - **Approximate Bayesian Computation** (ABC) [summary statistics?]
  - **Integrated Nested Laplace Approximation** (INLA) [applicable?]
- **Sequential Monte Carlo** (SMC)
- But wait! Can we **speed up MCMC**?
- Focus: **Generic MCMC** for problems with
  - **Tall data** - **many observations**
  - Models with **time-consuming likelihood evaluations** per subject (numerical solution to partial diff eq, game theory etc)

MCMC with a unbiased likelihood estimator

- **Aim**: the **posterior** density

$$p(\theta|y) \propto p(y|\theta)p(\theta)$$

- The **full likelihood** $p(y|\theta)$ is very **costly to evaluate**.

- **Unbiased estimator** $\hat{p}(y|\theta, u)$ of the likelihood is available

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

- $u \sim p(u)$ are auxilliary variables used to compute $\hat{p}(y|\theta, u)$.
- Examples:
  - Importance sampling: u are the particles
  - Here: u are indicators for the subset of observations

MCMC WITH A UNBIASED LIKELIHOOD ESTIMATOR

- The joint density

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

has the correct marginal density $p(\theta|y)$.

MCMC WITH A UNBIASED LIKELIHOOD ESTIMATOR

- The joint density

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

has the correct marginal density $p(\theta|y)$.

- Metropolis-Hastings at iteration $j+1$:
  - propose $\theta^* \sim q(\theta^*|\theta_j)$.
  - propose $u^* \sim p(u)$
  - accept the (u^*, θ^*)-pair with probability

$$\min\left[1, \frac{\hat{p}(y|\theta^*, u^*)p(\theta^*)}{\hat{p}(y|\theta_j, u_j)p(\theta_j)} \frac{q(\theta_j|\theta^*)}{q(\theta^*|\theta_j)}\right]$$

MCMC WITH A UNBIASED LIKELIHOOD ESTIMATOR

- The joint density

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

has the correct marginal density $p(\theta|y)$.

- Metropolis-Hastings at iteration $j+1$:
  - propose $\theta^* \sim q(\theta^*|\theta_j)$.
  - propose $u^* \sim p(u)$
  - accept the (u^*, θ^*)-pair with probability

$$\min\left[1, \frac{\hat{p}(y|\theta^*, u^*)p(\theta^*)}{\hat{p}(y|\theta_j, u_j)p(\theta_j)}\frac{q(\theta_j|\theta^*)}{q(\theta^*|\theta_j)}\right]$$

- This MH chain has $p(\theta|y)$ as its invariant distribution, irrespective of the variance of $\hat{p}(y|\theta, u)$ [Andrieu and Robert, AnnStat2009]
- Punchline: It's OK to replace the likelihood with an unbiased estimate.

Estimating the likelihood by subsampling

- Define:
  - $L(\theta) = p(y|\theta) = \prod_{k=1}^{n} p(y_k|\theta)$. **Likelihood**.
  - $\ell(\theta) = \ln L(\theta)$. **Log-likelihood**.
  - $\ell_k(\theta) = \ln p(y_k|\theta)$. **Log-likelihood contribution** of ith observation.

Estimating the likelihood by subsampling

- Define:
  - $L(\theta) = p(y|\theta) = \prod_{k=1}^{n} p(y_k|\theta)$. **Likelihood**.
  - $\ell(\theta) = \ln L(\theta)$. **Log-likelihood**.
  - $\ell_k(\theta) = \ln p(y_k|\theta)$. **Log-likelihood contribution** of ith observation.
- Unbiased estimation of the **log-likelihood** using **simple random sampling** (SRS) of size m:

$$\hat{l}(\theta) = \frac{n}{m} \sum_{k \in S(u)} \ell_k(\theta)$$

  where $S(u)$ is the set of m sampled observations, and $u = (u_1, ..., u_n)$ is vector of binary selection indicators.
- Note: same subsampling idea applies also to many non-iid models. Longitudinal data. Time-series with Markov behavior.
- An unbiased estimator of the **likelihood** can be obtain by **bias-correcting** $\exp\left(\hat{\ell}(\theta)\right)$.

Bias-correction

- Let z denote the error in the log-likelihood estimate:

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

- Now, since

$$E \exp\left[\hat{\ell}(\theta)\right] = \exp\left[\ell(\theta)\right] \cdot E\left[\exp\left(z\right)\right],$$

  an unbiased estimator of the likelihood is obtained by

$$\tilde{L}(\theta) \equiv \frac{\exp\left[\hat{\ell}(\theta)\right]}{E\left[\exp\left(z\right)\right]}$$

- Assuming that $z \sim N(0, \sigma_z^2)$ [CLT + big data setting]

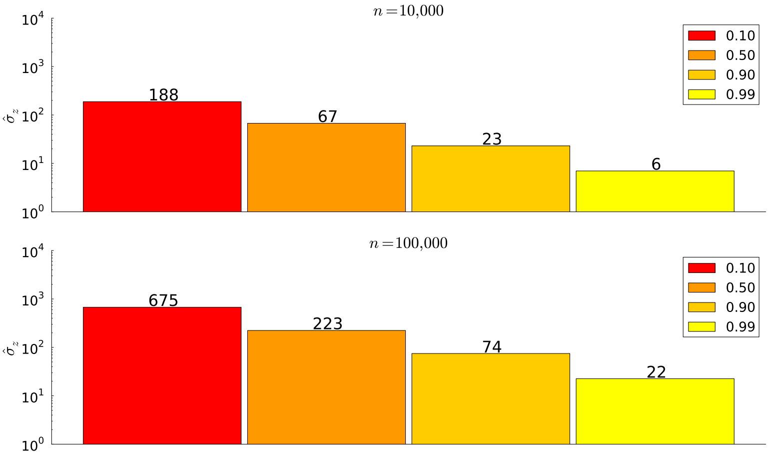
$$\tilde{L}(\theta) \equiv \frac{\exp\left[\hat{\ell}(\theta)\right]}{\exp\left(\sigma_z^2/2\right)}$$

- Other methods: Jackknife, generalized Poisson estimators etc

SIMPLE RANDOM SAMPLING IS NO GOOD

- **Simple random sampling** (SRS) gives a **HUGE variance** of the log-likelihood estimator
- ... so MH convergence is extremely slow (= doesn't work, gets stuck).
- SRS: $Pr(u_k = 1) = \pi_k = m/n$ is the same for all observations.
- Need more **efficient sampling** of data subsets!
- **Main idea** here: π_k should be large when $|\ell_k(\theta)|$ is large.

SRS IS FAR FROM THE OPTIMAL $\sigma_Z \approx 1$



πPS sampling + Horvitz-Thompson estimator

- πPS-**sampling**: $\pi_i \propto |\ell_i(\theta)|$. Sampling **without replacement**.
- **Horvitz-Thompson's estimator** of the log-likelihood

$$\hat{\ell}^{HT}(\theta) = \sum_{k \in S} \frac{\ell_k(\theta)}{\pi_k} = \sum_{k \in F} \frac{\ell_k(\theta)}{\pi_k} u_k$$

- **Asymptotic normality** of z holds (Rosén, 1972).
- Unbiased **estimate of the variance** is

$$\hat{V}[\hat{\ell}^{HT}(\theta)] = \sum_{k \in S} \sum_{l \in S} (1 - \frac{\pi_k \pi_l}{\pi_{kl}}) \frac{\ell_k(\theta)}{\pi_k} \frac{\ell_l(\theta)}{\pi_l}, \qquad (1)$$

where $\pi_{kl} = P(u_k = 1, u_l = 1)$.
- πPS is **time-consuming** [computing π_{kl}, sampling, estimating σ_z^2].

PPS sampling + Hansen-Hurwitz estimator

- **PPS sampling** is like πPS, but **with replacement**. Much faster!
- **Hansen-Hurwitz estimator** of the log-likelihood

$$\hat{\ell}^{HH}(\theta) \quad = \quad \frac{1}{m}\sum_{i=1}^{m}\frac{\ell_{u_i}(\theta)}{p_{u_i}}.$$

$$\hat{V}[\hat{\ell}^{HH}(\theta)] = \frac{1}{m(m-1)}\sum_{i=1}^{m}\left(\frac{\ell_{u_i}(\theta)}{p_{u_i}} - \hat{\ell}^{HH}(\theta)\right)^2$$

PPS sampling + Hansen-Hurwitz estimator

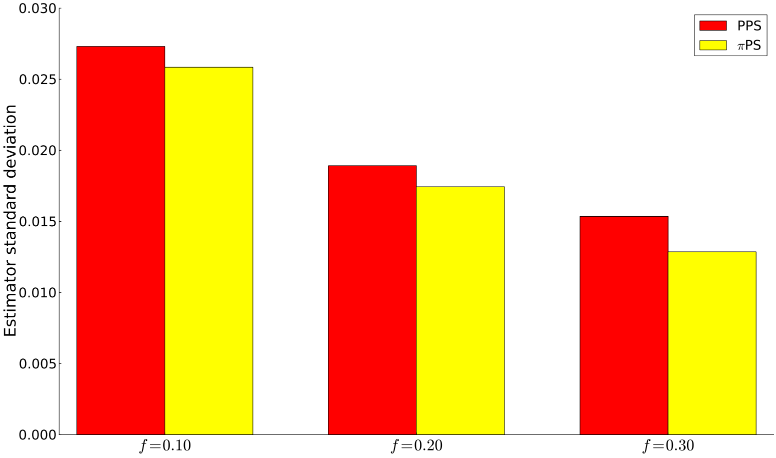
- **PPS sampling** is like πPS, but **with replacement**. Much faster!
- **Hansen-Hurwitz estimator** of the log-likelihood

$$\hat{\ell}^{HH}(\theta) \quad = \quad \frac{1}{m}\sum_{i=1}^{m}\frac{\ell_{u_i}(\theta)}{p_{u_i}}.$$

$$\hat{V}[\hat{\ell}^{HH}(\theta)] = \frac{1}{m(m-1)}\sum_{i=1}^{m}\left(\frac{\ell_{u_i}(\theta)}{p_{u_i}} - \hat{\ell}^{HH}(\theta)\right)^2$$

- **Asymptotic normality** of z holds (Rosén, 1972).
- The p_i need to be **good** proxies of $|\ell_i(\theta)|$.
- Any **surrogate/approximate model** can be used.
- What if no surrogate model is available? Need an **general method for approximating** $\ell_i(\theta)$.

PPS HAS ROUGHLY THE SAME VARIANCE AS πPS



Approximating $l_i(\theta)$ - Gaussian process

- Wanted: **approximation of the log-likelihood contribution**:

$$d \to \ell(\theta; d)$$

for any data point $d = (y, x)$ and parameter vector θ.

Approximating $l_i(\theta)$ - Gaussian process

- Wanted: **approximation of the log-likelihood contribution**:

$$d \to \ell(\theta; d)$$

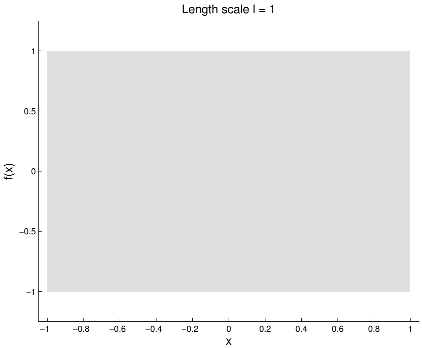
  for any data point $d = (y, x)$ and parameter vector θ.

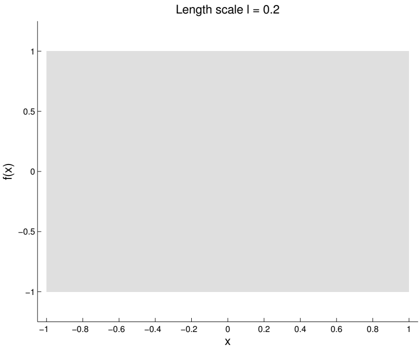
- Given θ, assume a **noise-free Gaussian Process (GP) prior** over d-space:

$$\ell(\theta; d) \sim GP\left[0, k(d, d')\right]$$

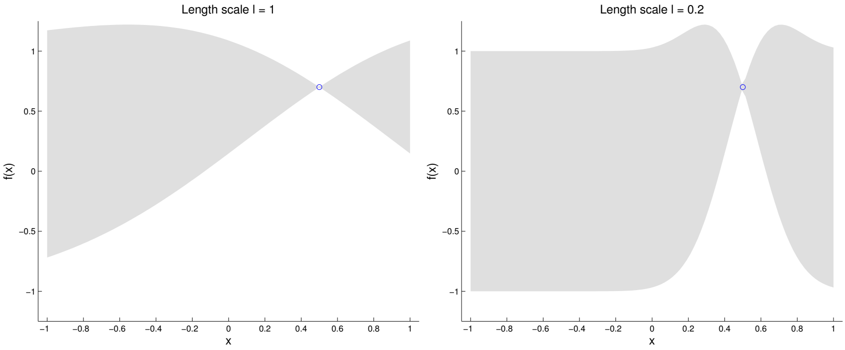
- Compute $\ell(\theta; d)$ for all $d \in V$, a **small fixed subset** of the data.
- **Update the GP prior** using $\ell_V(\theta) = \{\ell(\theta; d)\}_{d \in V}$ to a GP posterior.

Learning a noise-free Gaussian process

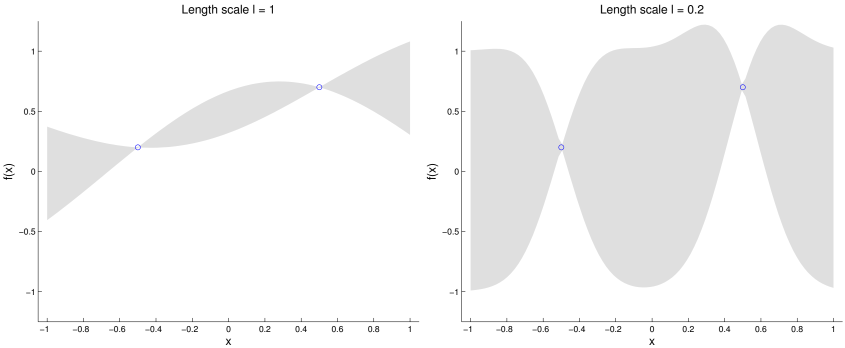




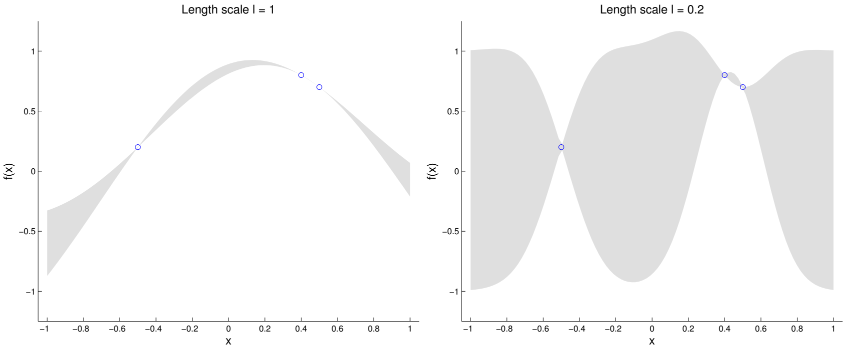
Learning a noise-free Gaussian process



Learning a noise-free Gaussian process



LEARNING A NOISE-FREE GAUSSIAN PROCESS



APPROXIMATING $\ell_i(\theta)$ - GAUSSIAN PROCESS, CONT.

- Use the GP to **predict** $\ell(\theta; d)$ for all $d \in V^c$

$$\hat{\ell}_{V^c}(\theta) = K(d_{V^c}, d_V)K(d_V, d_V)^{-1}\ell_V(\theta),$$

where $K(d_V, d_V)$ is the covariance matrix for the data points in V.

Approximating $\ell_i(\theta)$ - Gaussian process, cont.

- Use the GP to **predict** $\ell(\theta; d)$ for all $d \in V^c$

$$\hat{\ell}_{V^c}(\theta) = K(d_{V^c}, d_V)K(d_V, d_V)^{-1}\ell_V(\theta),$$

  where $K(d_V, d_V)$ is the covariance matrix for the data points in V.

- The **kernel hyperparameters** are chosen to minimize the prediction errors on all $d \in V^c$ for some $\theta = \hat{\theta}$ (e.g. posterior mode). **Before MCMC**.

- **Important**: $K(d_{V^c}, d_V)$ and $K(d_V, d_V)^{-1}$ are **computed once, before the MCMC**.

- In each MCMC iteration $\hat{\ell}_{V^c}(\theta)$ is obtained by two matrix-vector multiplications. Fast!

Approximating $\ell_i(\theta)$ - Thin-plate surfaces

- For large datasets, GPs can be computationally demanding.
- Approximate GPs for large data exist, and likely to improve over time.
- Alt. approach for large data: **regularized thin-plate spline surfaces**.
- The **knot locations** are chosen by kmeans + boundary
- **Shrinkage** λ (or Λ) chosen to minimize the prediction errors for all $d \in V^c$.
- **Predicting** any $d \in V^c$

$$\hat{\ell}_{V^c}(\theta) = W_{V^c}(W_V' W_V + \lambda I)^{-1} W_V' \ell_V(\theta),$$

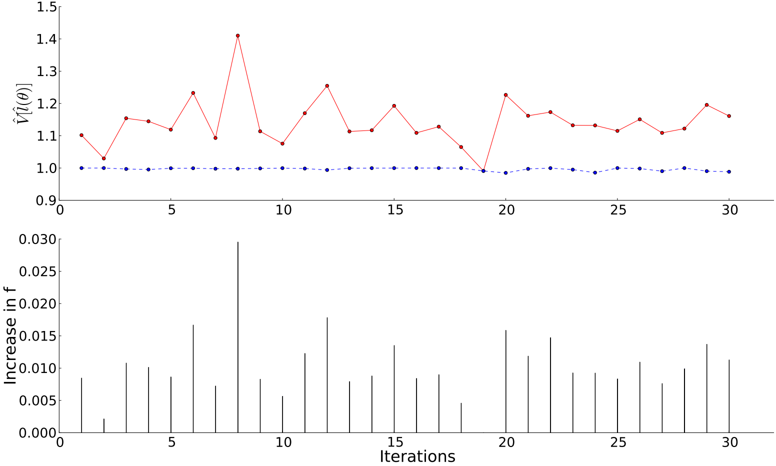
where W_V and W_{V^c} are basis-expansion matrices in d-space.

Adaptive sampling fraction

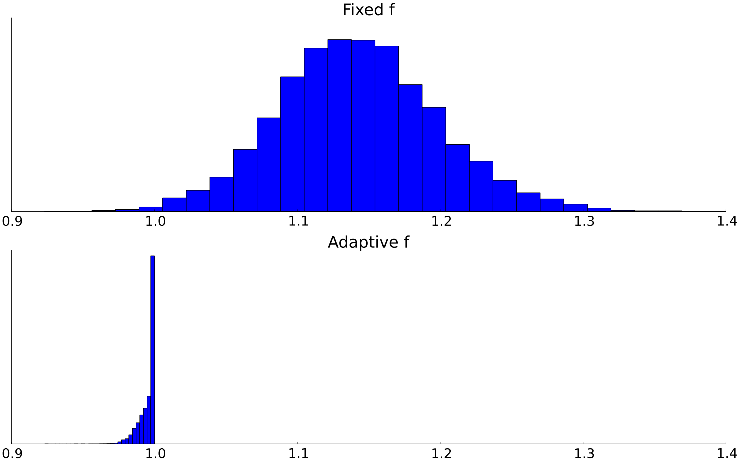
- Variance of $\hat{\ell}(\theta)$ (σ_Z^2) should be close to unity for optimal efficiency/computing time trade-off (Doucet, Pitt and Kohn, 2012).
- Sampling fraction $f = m/n$ can be chosen adaptively in each MCMC draw.
- If $\sigma_z^2 > 1$, increase sampling fraction to $f = m^*/n$, where m^* is a guess of the sample size needed to reach some $\sigma_Z^2 = v_{max}$.
- For PPS we have a good guess by backing out m from the variance formula

$$m^* = \frac{1}{v_{max}(m-1)} \sum_{i=1}^{m} \left(\frac{\ell_{u_i}(\theta)}{p_{u_i}} - \hat{\ell}^{HH}(\theta) \right)^2$$

Adaptive sampling fraction, cont.



Adaptive sampling fraction, cont.



Firm bankruptcy and excess cash holding

- **Bivariate probit** with **endogenity**

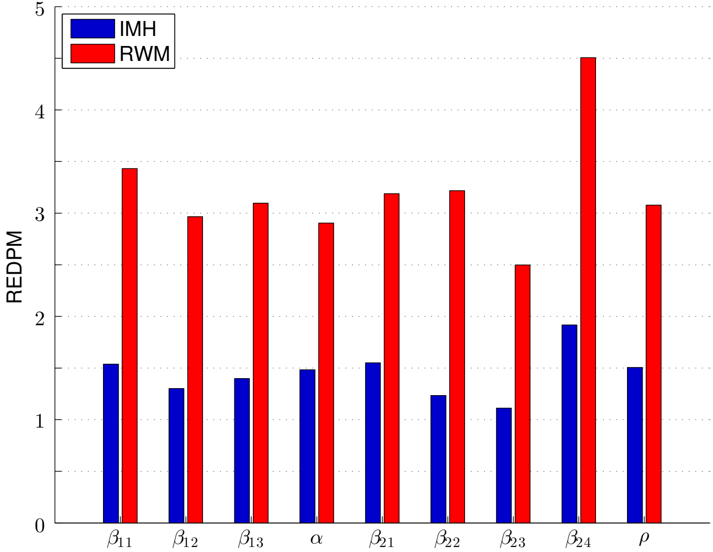
$$
\begin{aligned}
y_1^* &= \beta_{10} + \beta_{11} \cdot x_1 + \beta_{12} \cdot x_2 + \alpha \cdot y_2 + \varepsilon_1 \\
y_2^* &= \beta_{20} + \beta_{21} \cdot x_1 + \beta_{22} \cdot x_3 + \beta_{23} \cdot x_4 + \varepsilon_2 \\
y_1 &= I(y_1^* > 0) \\
y_2 &= I(y_2^* > 0)
\end{aligned}
$$

  where ε_1 and ε_2 are standard Gaussian with correlation ρ.
- Variables:
  - y_1 = Bankrupt, y_2 = Excess cash
  - x_1 = Profit, x_2 = leverage, x_3 = fixed assets, x_4 = firm size.
- Cash has many troublesome outliers $\Rightarrow$ Better with binary Excess cash.
- **Time-consuming likelihood** (bivariate normal integral).
- Special case of a **Gaussian copula model**.

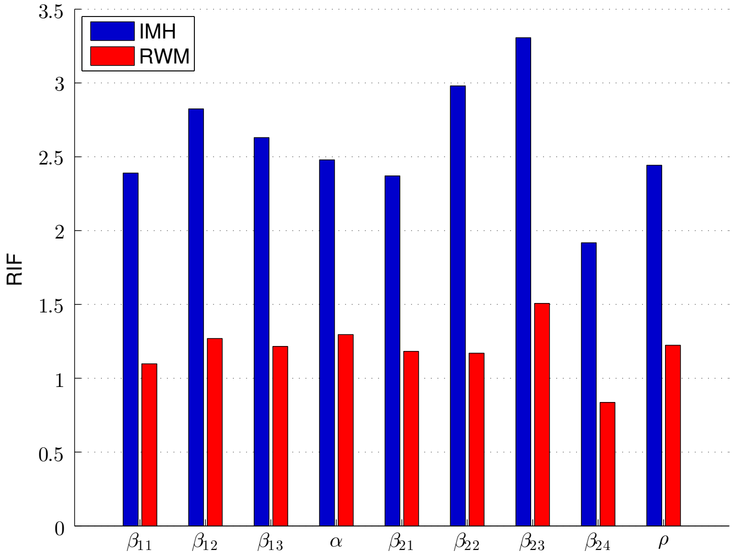
Firm bankruptcy data

- Dataset used has **half a million observations**.
- Observations within the firm are assumed independent conditional on time-varying covariates.
- Extension to random effects is possible.
- 5% of data is used for fitting thin-plate approximation.
- 8% of data sampled by PPS on average.
- 10,000 post burn-in draws.

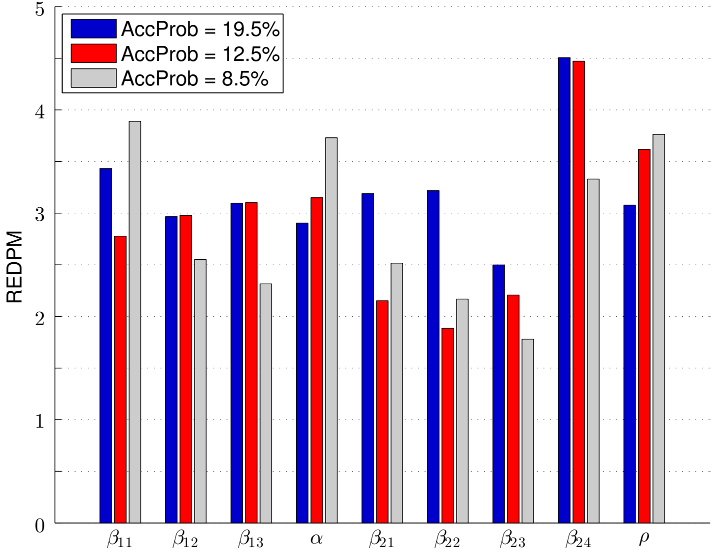
Comparing the Effective Draws Per Minute



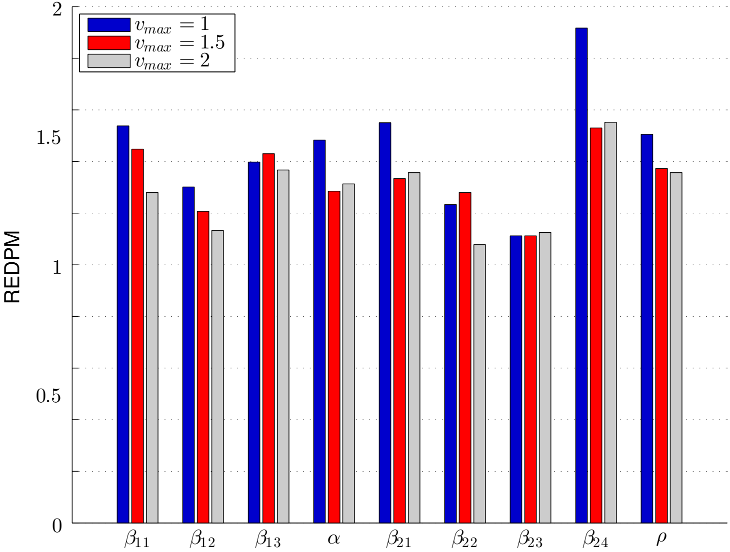
Inefficiency factor



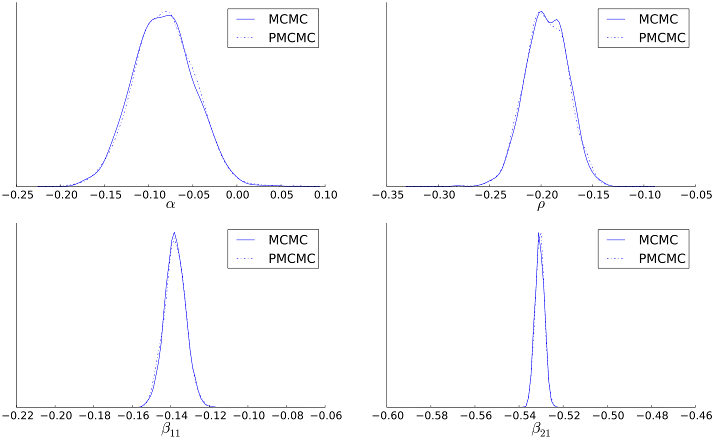
Scaling of the Random Walk Proposal



Targeting different σ_Z^2 - IMH



Marginal posteriors



Posterior summary

	Posterior mean	2.5%	97.5%
Parameters in y_1^*			
β_{11} (Intercept)	-2.543	-2.570	-2.517
β_{12} (Earnings)	-0.138	-0.148	-0.127
β_{13} (Leverage)	0.304	0.292	0.316
α (Excess cash)	-0.083	-0.151	-0.015
Parameters in y_2^*			
β_{21} (Intercept)	-0.017	-0.020	-0.013
β_{22} (Earnings)	-0.531	-0.535	-0.527
β_{23} (Tangible)	0.230	0.226	0.234
β_{24} (Size)	-0.263	-0.267	-0.259
ρ (Correlation)	-0.195	-0.235	-0.155

Conclusions

- We have proposed a general framework for **Pseudo-MCMC** based on **efficient data subsampling**.
- Bias-corrected log-likelihood estimator from **PPS sampling** combined with the **Hansen-Hurwitz estimator**.
- **Gaussian Process** or Regularized **thin-plate spline surface** for computing **efficient PPS-weights**.
- More efficient draws per minute in a bivariate probit **application to financial data**. Biggest gain for weaker proposals.
- **Future work**:
  - **more examples**
  - **improved PPS-weights**, especially for problems with many covariates.
  - **other sampling schemes**