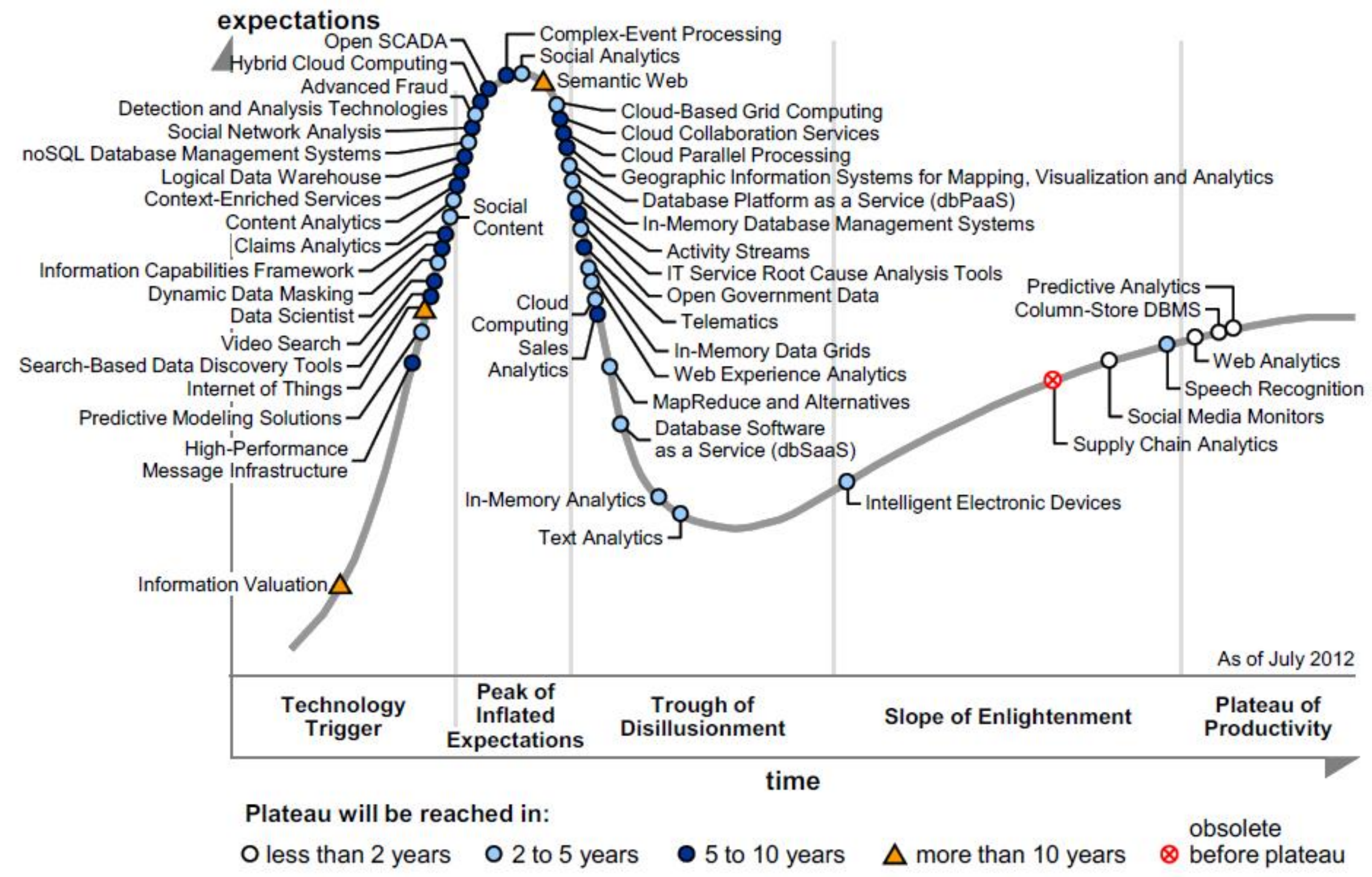
Probabilistic Models for Big Data

Alex Davies and Roger Frigola
University of Cambridge
13th February 2014

The State of Big Data

Figure 1. Hype Cycle for Big Data, 2012



Source: Gartner (July 2012)

Why probabilistic models for Big Data?

1. “If you don’t have to worry about overfitting, your model is likely too small.” (Welling)
2. For scientific problems, prior knowledge is usually available.
3. Not all data points inform all latent variables.
4. We can have big data with small N (e.g. 20 experiments, 10 TB per experiment.)
5. How do you regularize models with complex dependencies with billions of parameters.

Trueskill

Xbox Live ranking system. Predict player skill

2,000,000 matches per day

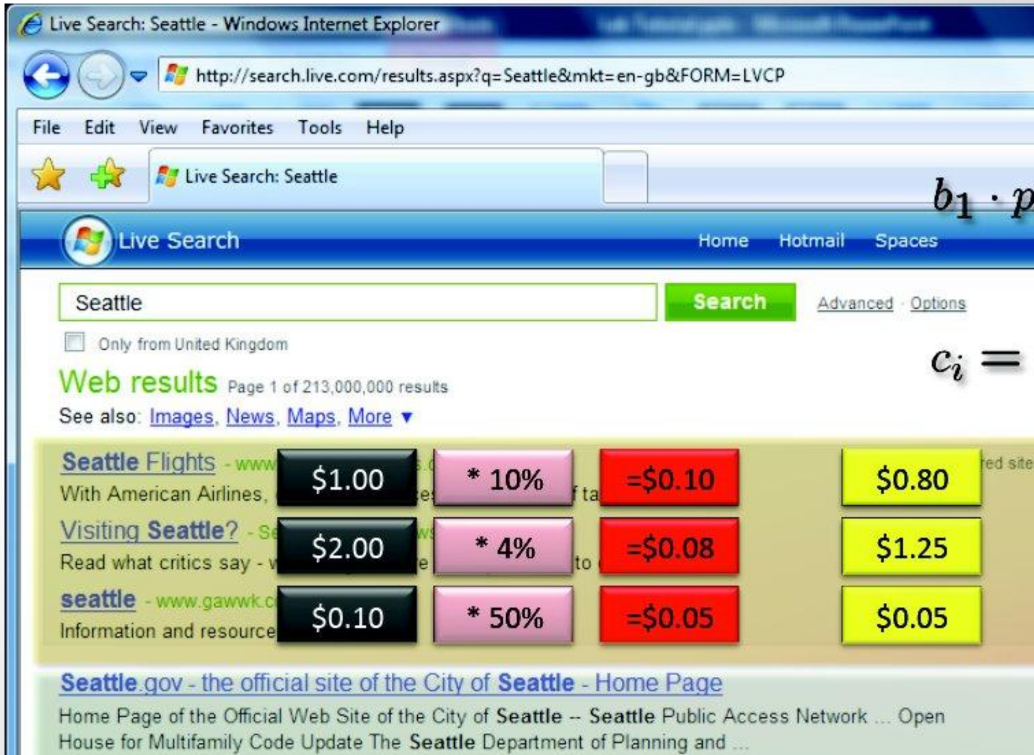
10,000,000 players

Bing advertising

Predict expected click revenue for ads.

Requires ~10ms response times.

$832 million in annual revenue. Every percent accuracy matters.



Seattle.gov - the official site of the City of Seattle - Home Page

Home Page of the Official Web Site of the City of Seattle -- Seattle Public Access Network ... Open House for Multifamily Code Update The Seattle Department of Planning and ...

Main considerations for the Big Data revolution

- Trading off Accuracy/Time/Data size.
- Large scale infrastructure.

Trading off Accuracy-Time-Data Size

For a given computational budget, can we guarantee a level of inferential accuracy as data grows in size?

Two main approaches to achieve this:

- **Divide and conquer**: divide in subproblems and piece solutions together.
- **Algorithmic weakening**: hierarchy of algorithms ordered by computational complexity.

(Jordan, 2013, Bernoulli 19(4), 1378–1390)

Stochastic Variational Inference

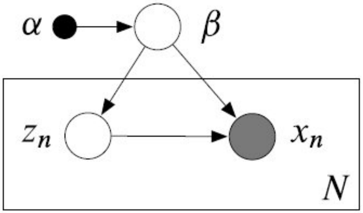
- Bringing the power of stochastic gradient descent to variational inference

- For a complete understanding of SVI, three components are required
  - Stochastic Gradient Descent
  - Natural Gradients
  - Variational Inference

Stochastic Variational Inference

SVI applies to models of the following form:

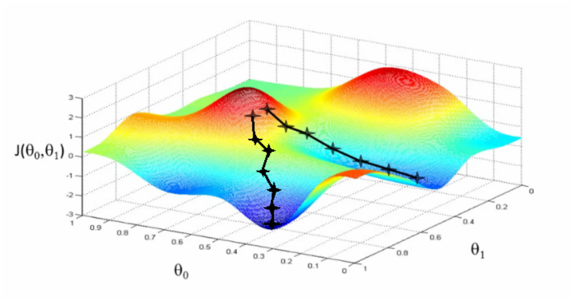
- Local and global latent parameters



- Conjugate distributions

Stochastic Gradient Descent

The poster child of large scale optimization.



Stochastic Gradient Descent

The trade-offs of large scale learning:

$$\epsilon = \mathbb{E}\left[E(\bar{f}_n) - E(f_n)\right] + \mathbb{E}\left[E(f_n) - E(f^*_{\mathcal{F}})\right] + \mathbb{E}\left[E(f^*_{\mathcal{F}}) - E(f^*)\right] \quad (1)$$

- f^* is the optimal function
- $f^*_{\mathcal{F}}$ is the best function in our function class
- f_n is the empirical risk minimizing function
- $\bar{f}_n$ is the approximation to f_n from early stopping of optimization

Stochastic Gradient Descent

The trade-offs of large scale learning:

$$\epsilon = \mathbb{E}\left[E(\bar{f}_n) - E(f_n)\right] + \mathbb{E}\left[E(f_n) - E(f^*_{\mathcal{F}})\right] + \mathbb{E}\left[E(f^*_{\mathcal{F}}) - E(f^*)\right] \tag{2}$$

$$\epsilon = \epsilon_{\text{optimization}} + \epsilon_{\text{estimation}} + \epsilon_{\text{approximation}}$$

- f^* is the optimal function
- $f^*_{\mathcal{F}}$ is the best function in our function class
- f_n is the empirical risk minimizing function
- $\bar{f}_n$ is the approximation to f_n from early stopping of optimization

STOCHASTIC GRADIENT DESCENT

	GD	SGD
Time per iteration	n	1
Iterations to accuracy ρ	$\log \frac{1}{\rho}$	$\frac{1}{\rho}$
Time to accuracy ρ	$n \log \frac{1}{\rho}$	$\frac{1}{\rho}$
Time to excess error ϵ	$\frac{1}{\epsilon^\alpha} \log^2 \frac{1}{\epsilon}$	$\frac{1}{\epsilon}$

Line 4 comes from mystical frequentist bounds with $\alpha \in [1, 2]$. It assumes strong convexity or “certain assumptions” and also that we have optimally traded off between the three forms of error.

Stochastic Gradient Descent

But convergence is easy to assure. All that is required is the following constraints on the step size schedule:

$$\sum \rho_t = \infty$$

$$\sum \rho_t^2 < \infty$$

Stochastic Variational Inference

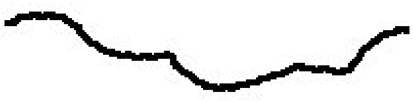
- Bringing the power of stochastic gradient descent to variational inference
- For a complete understanding of SVI, three components are required
  - Stochastic Gradient Descent ✓
  - Natural Gradients
  - Variational Inference

NATURAL GRADIENTS

Natural gradients are a sensible choice of gradients that can be used when optimizing probability distributions. Naturally.

Natural Gradients

With gradient descent, we want to optimize through some "nice" space with respect to the function we are optimizing. It would be nice if it was pretty smooth and well behaved, and the length-scale over which the function tends to vary stays relatively stable.



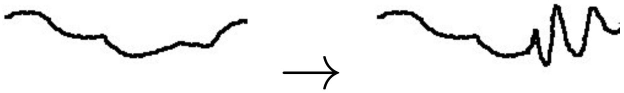
Natural Gradients

When optimizing a probability distribution, our loss function is a function of the distribution. But we represent our distributions in terms of their parameters. Hopefully our loss function is "nice" wrt the probability distribution, but unless we have very specifically chosen the parameterization for the loss function we are optimizing, optimizing in parameter space will probably make things worse.

$$\theta \to p(\theta) \to f(p(\theta))$$

Natural Gradients

When optimizing a probability distribution, our loss function is a function of the distribution. But we represent our distributions in terms of their parameters. Hopefully our loss function is “nice” wrt the probability distribution, but unless we have very specifically chosen the parameterization for the loss function we are optimizing, optimizing in parameter space will probably make things worse.

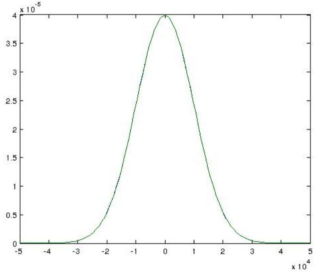


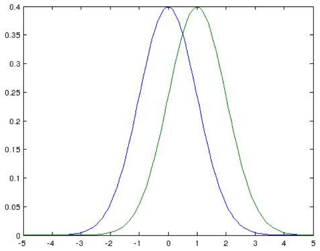
NATURAL GRADIENTS

Example: Normal distribution with standard parameterization

NATURAL GRADIENTS

A step size of 10 in parameter space takes us from $\mathcal{N}(0, 10000) \rightarrow \mathcal{N}(10, 10000)$ while a step size of 1 in parameter space can take us from $\mathcal{N}(0, 1) \rightarrow \mathcal{N}(1, 1)$





Natural Gradients

When we are optimizing a probability distribution, we want to take nice consistent steps in “probability space”. ie from one step to another, we want to have moved a consistent distance. A sensible measure of distance for probability distributions is the symmetrized KL divergence.

Natural Gradients

So how about ....
Instead of finding the direction

$$\arg\max_{d\theta} f(\theta + d\theta) \text{ where } \|d\theta\| < \epsilon$$

we go for

$$\arg\max_{d\theta} f(\theta + d\theta) \text{ where } D_{KL}^{sym}(\theta, \theta + d\theta) < \epsilon$$

NATURAL GRADIENTS

So how about ....
Instead of finding the direction

$$\arg\max_{d\theta} f(\theta + d\theta) \text{ where } \|d\theta\| < \epsilon$$

we go for

$$\arg\max_{d\theta} f(\theta + d\theta) \text{ where } D_{KL}^{sym}(\theta, \theta + d\theta) < \epsilon$$

To do this, we can make a linear transformation of our space so that $\|d\tilde{\theta}\| = D_{KL}^{sym}(\theta, \theta + d\theta)$

Natural Gradients

It turns out that you can do this by using the inverse of the *Fisher information matrix* **G** as the linear transformation.

$$\hat{\nabla} f = \mathbf{G}^{-1} \nabla f \tag{3}$$

Now we have a sensible optimization step that's independent of the parameterization of our distribution.

Stochastic Variational Inference

- Bringing the power of stochastic gradient descent to variational inference
- For a complete understanding of SVI, three components are required
  - Stochastic Gradient Descent ✓
  - Natural Gradients ✓
  - Variational Inference

Variational Inference

Approximate the full distribution with one in a “nice” class, that we can calculate with easily.

$$Q(\theta|\phi) \approx P(\theta|X)$$

Variational Inference

Approximate the full distribution with one in a “nice” class, that we can calculate with easily.

$$Q(\theta|\phi) \approx P(\theta|X)$$

We need to find the value of ϕ that makes $Q(\theta|\phi)$ as close as possible to $P(X|\theta)$.

Variational Inference

We already know the go-to measure for distance between distributions, KL divergence. So we want to minimize:

$$D_{KL}\left(Q(\theta|\phi), P(\theta|X)\right)$$

Variational Inference : KL-divergence

We already know the go-to measure for distance between distributions, KL divergence. So we want to minimize:

$$D_{KL}\left(Q(\theta|\phi), P(\theta|X)\right) = \mathbb{E}_Q\left[\log Q(\theta|\phi)\right] - \mathbb{E}_Q\left[\log P(\theta|X)\right]$$

Variational Inference : KL-divergence

We already know the go-to measure for distance between distributions, KL divergence. So we want to minimize:

$$D_{KL}\left(Q(\theta|\phi), P(\theta|X)\right) = \mathbb{E}_Q\left[\log Q(\theta|\phi)\right] - \mathbb{E}_Q\left[\log P(\theta|X)\right]$$

$$D_{KL}\left(Q, P\right) = \underbrace{\mathbb{E}_Q\left[\log Q(\theta|\phi)\right] - \mathbb{E}_Q\left[\log P(\theta, X)\right]}_{\text{-ELBO}} + \underbrace{\log P(X)}_{\text{Marginal likelihood}}$$

Variational Inference : KL-divergence

$$D_{KL}(Q,P) = \underbrace{\mathbb{E}_Q\left[\log Q(\theta|\phi)\right] - \mathbb{E}_Q\left[\log P(\theta,X)\right]}_{\text{-ELBO}} + \underbrace{\log P(X)}_{\text{Log marginal likelihood}}$$

1. Since $\log P(X)$ is independent of ϕ, we can minimize the KL by maximizing the ELBO
2. Since the KL is always positive, the ELBO is a lower-bound on the log marginal likelihood

Variational Inference : KL-divergence

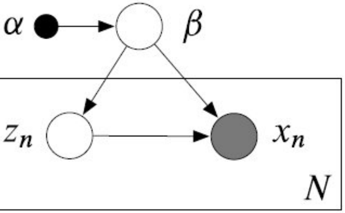
To optimize the parameters of the variational distribution ϕ, optimize the ELBO using your favourite gradient method.

$$\mathcal{L}(\phi) = \mathbb{E}_Q \left[\log P(\theta, X)\right] - \mathbb{E}_Q \left[\log Q(\theta|\phi)\right]$$

Stochastic Variational Inference

- Bringing the power of stochastic gradient descent to variational inference
- For a complete understanding of SVI, three components are required
  - Stochastic Gradient Descent ✓
  - Natural Gradients ✓
  - Variational Inference ✓

Stochastic Variational Inference



The derivative of the ELBO in a conjugate, global and local variable model:

$$\nabla_\lambda \mathcal{L}(\phi) = \mathbf{G}(\mathbb{E}_Q[\eta(x, z, \alpha)] - \lambda) \tag{4}$$

Stochastic Variational Inference

The derivative of the ELBO in a conjugate, global and local variable model:

$$\nabla_\lambda \mathcal{L}(\phi) = \mathbf{G}(\mathbb{E}\left[\eta(x, z, \alpha)\right] - \lambda) \tag{5}$$

That **G** looks like a total hassle to calculate. (Quadratic in number of variational parameters).

Stochastic Variational Inference

The derivative of the ELBO in a conjugate, global and local variable model:

$$\nabla_\lambda \mathcal{L}(\phi) = \mathbf{G}(\mathbb{E}\left[\eta(x, z, \alpha)\right] - \lambda) \tag{6}$$

That **G** looks like a total hassle to calculate. (Quadratic in number of variational parameters).

$$\begin{aligned}\tilde{\nabla}_\lambda \mathcal{L}(\phi) &= \mathbf{G}^{-1}\mathbf{G}(\mathbb{E}\left[\eta(x, z, \alpha)\right] - \lambda) && (7)\\ &= \mathbb{E}\left[\eta(x, z, \alpha)\right] - \lambda && (8)\end{aligned}$$

Stochastic Variational Inference

The *natural* derivative of the ELBO in a conjugate, global and local variable model:

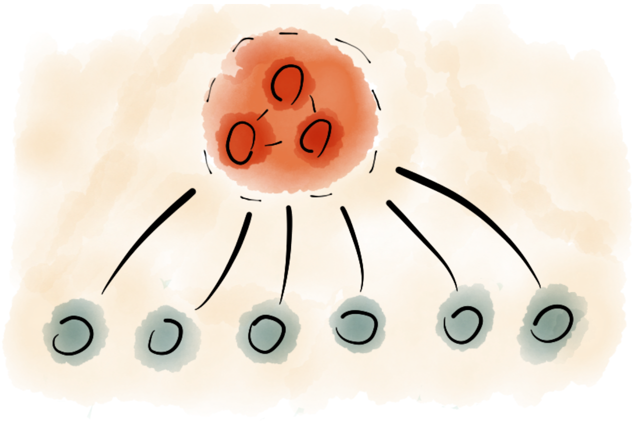
$$\begin{aligned}\tilde{\nabla}_\lambda \mathcal{L}(\phi) &= \mathbb{E}\left[\eta(x, z, \alpha)\right] - \lambda \qquad (9)\\ &= \alpha + N(\mathbb{E}_{\phi_i}[t(x_i, z_i)], 1) - \lambda \qquad (10)\end{aligned}$$

(t are the sufficient statistics)

Stochastic Variational Inference

Breather slider.

Gaussian Processes with inducing points



Gaussian Processes with inducing points

SVI applies to models of the following form:

- Local and global parameters ✓
- Conjugate distributions ✓

Stochastic Variational Inference for GPs

In order to apply stochastic variational inference to GPs, the inducing points from sparse GPs become our global variational parameters. Demo time!

LARGE SCALE GRAPHICAL MODELS

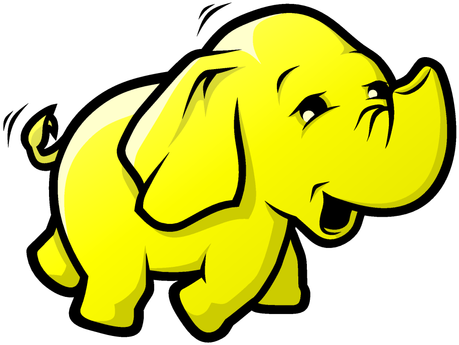
Many of the hardest things about scaling a system to truly big data are actually engineering challenges that come from distributed computation.

1. Shared memory distribution
2. Communciation
3. Synchronization
4. Fault tolerance

This shouldn't be our job.

Large scale graphical models

Hadoop was the first excellent programming paradigm for big data.



LARGE SCALE GRAPHICAL MODELS

You use a restricted programming structure, Map-Reduce, and everything else just magically works.

Map-Reduce turns out to be very useful for *Data processing* tasks but not so much for *Machine learning*.

Large scale graphical models

You use a restricted programming structure, Map-Reduce, and everything else just magically works.

Map-Reduce turns out to be very useful for *Data processing* tasks but not so much for *Machine learning*.

LARGE SCALE GRAPHICAL MODELS

An alternative to Map-Reduce is Bulk Synchronous Processing (BSP), which is implemented in Giraph and GraphLab

LARGE SCALE GRAPHICAL MODELS

The power of these frameworks are in their simplicity. Rather than specifying a map and a reduce function you simply specify a node function. At each iteration a node can perform local computations, recieve messages along it's edges from the previous round of computation and send message out along it's edges.

Page rank code

```
class PageRankVertex < Vertex
  def compute
    if superstep >= 1
      sum = messages.inject(0) {|total,msg| total += msg; total }
      @value = (0.15 / num_nodes) + 0.85 * sum
    end

    if superstep < 30
      deliver_to_all_neighbors(@value / neighbors.size)
    else
      halt
    end
  end
end
```

LARGE SCALE GRAPHICAL MODELS

Summary: If you can write an algorithm that iteratively computes locally on nodes in a graph, you can use Giraph and GraphLab and scale to billions of nodes without every knowing a thing about parallel architecture.

MCMC for Big Data

Conventional wisdom: MCMC not suited for Big Data.

Is MCMC the optimal way to sample a posterior for large data sets and limited computation time?

MCMC

Data y, unknown parameter θ

GOAL: sample posterior

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

where $p(y|\theta) = \prod_{i=1}^{N} p(y_i|\theta)$

by obtaining a set $\{\theta_1, \theta_2, ..., \theta_T\}$ distributed according to $p(\theta|y)$.

$$I = \mathbb{E}_{p(\theta|y)}[f(\theta)] \approx \frac{1}{T}\sum_{t=1}^{T} f(\theta_t), \quad \text{with } \theta_t \sim p(\theta|y)$$

Metropolis-Hastings

Simple algorithm to generate a Markov chain with a given stationary distribution.

Start with θ_0

1. Draw candidate $\theta' \sim q(\theta'|\theta_t)$.
2. Compute acceptance probability

$$P_a = \min(1, \frac{p(\theta'|y)\ q(\theta_t|\theta')}{p(\theta_t|y)\ q(\theta'|\theta_t)})$$

3. Draw $u \sim \mathrm{Uniform}[0,1]$. If $u < P_a$ set $\theta_{t+1} \leftarrow \theta'$ otherwise set $\theta_{t+1} \leftarrow \theta_t$.

PROBLEM: evaluating $p(\theta|y)$ is $\mathcal{O}(N)$.

Bias and Variance

Samples can be used to compute expectations wrt the posterior

$$I = \mathbb{E}_{p(\theta|y)}[f(\theta)]$$
$$I \approx \hat{I} = \frac{1}{T}\sum_{t=1}^{T} f(\theta_t)$$

After burn-in, $\hat{I}$ is an unbiased estimator of I

$$\mathbb{E}_{\text{chains}}[\hat{I}] = I$$
$$\text{Var}_{\text{chains}}[\hat{I}] = \sigma_f^2 \frac{\tau}{T}$$

New Paradigm: Accept Bias - Reduce Variance

If we can sample faster, T grows and the variance reduces.

Risk is useful to study bias/variance trade-off

$$R = \mathbb{E}_{\text{chains}}[(I - \hat{I})^2] = B_\epsilon^2 + V_\epsilon$$

The setting of ϵ that minimises risk depends on available computation time.

Traditional MCMC setting assumes $T \to \infty$, best strategy is to have no bias since variance will go down to zero.

Existing Methods

- Use mini-batches of data and omit MH acceptance test

  Stochastic Gradient Langevin Dynamics (Welling and Teh, 2011) and Stochastic Gradient Fisher Scoring (Ahn, Koratikkara and Welling, 2012)

- Approximate MH Acceptance test

  (Koratikkara, Chen and Welling, 2014) and (Bardenet, Doucet and Holmes, 2014)

Approximate MH Acceptance Test

Acceptance probability

$$P_a = \min(1, \frac{p(\theta'|y)\ q(\theta_t|\theta')}{p(\theta_t|y)\ q(\theta'|\theta_t)})$$

$\mathcal{O}(N)$ computation to obtain one bit of information.

Options:

- Use unbiased estimators of the likelihood that use only a subset of data $\rightarrow$ high variance.
- Reformulate MH acceptance test as a statistical decision problem.

Approximate MH Acceptance Test

Equivalent reformulation of MH acceptance test

1. Draw $u \sim \mathrm{Uniform}[0, 1]$.
2. Compute

$$\mu_0 = \frac{1}{N} \log\left(u \frac{p(\theta_t)\ q(\theta'|\theta_t)}{p(\theta')\ q(\theta_t|\theta')}\right)$$

$$\mu = \frac{1}{N} \sum_{i=1}^{N} l_i, \quad l_i = \log \frac{p(y_i|\theta')}{p(y_i|\theta_t)}$$

3. If $\mu > \mu_0$ set $\theta_{t+1} \leftarrow \theta'$ otherwise set $\theta_{t+1} \leftarrow \theta_t$.

This looks like a hypothesis test!

Approximate MH Acceptance Test

Approximate μ with a random subset of the data

$$\mu \approx \bar{l} = \sum_{j=1}^{n} l_j, \quad l_j = \log \frac{p(y_j|\theta')}{p(y_j|\theta_t)} \tag{1}$$

$$s_l = \sqrt{(\bar{l^2} - (\bar{l})^2)\frac{n}{n-1}}, \quad \text{Standard deviation of } l \tag{2}$$

$$s_{\bar{l}} = \frac{s_l}{\sqrt{n}}\sqrt{1 - \frac{n-1}{N-1}}, \quad \text{Standard deviation of } \bar{l} \tag{3}$$

Test statistic

$$t = \frac{\bar{l} - \mu_0}{s_{\bar{l}}}$$

If *n* is large enough for Central Limit Theorem to hold, and $\mu = \mu_0$ *t* follows a standard Student-t dist. with n-1 DOF.

Approximate MH Acceptance Test

If $1 - \Phi_{n-1}(|t|) < \epsilon$, μ is statistically significantly different from μ_0. (where Φ_{n-1} is a Student-t cdf)

Then, if $1 - \Phi_{n-1}(|t|) < \epsilon$, the approximate MH test becomes: if $\bar{l} > \mu_0$ set $\theta_{t+1} \leftarrow \theta'$ otherwise set $\theta_{t+1} \leftarrow \theta_t$.

If the test is not significant at a level ϵ we draw more samples and recompute t.

Recap: Approximate MH Acceptance Test

Can often make confident decisions even when $n < N$.

Saves time that can be used to draw longer chains and hence reduce variance.

- High $\epsilon \Rightarrow$ High bias, fast
- Low $\epsilon \Rightarrow$ Low bias, slow (high *n* required)

ϵ is a knob to trade off bias and variance.

Optimal Sequential Test Design

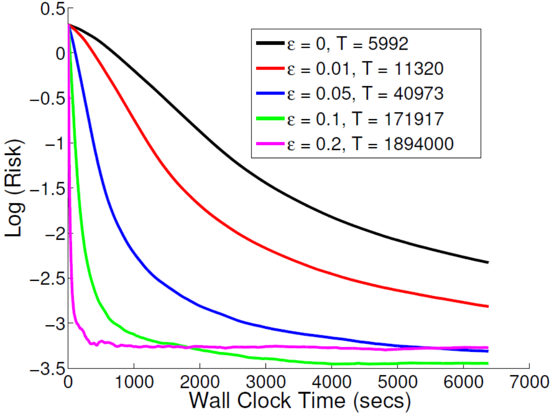
How to choose the parameters of the algorithm?

- Choose initial mini-batch size ≈ 500 for Central Limit Theorem to hold.
- Keep ϵ as small as possible while maintaining low average data use.

Experiments: Independent Component Analysis

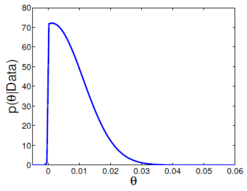
Posterior over unmixing matrix.

Risk in mean Amari distance to “ground truth”.

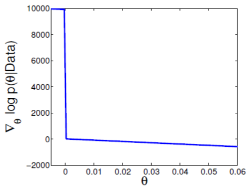


Experiments: Stochastic Gradient Langevin Dynamics (SGLD) + Approximate MH Test

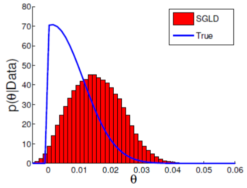
Linear regression with Laplace prior. SGLD can be thrown off track in areas of large gradients and low density.



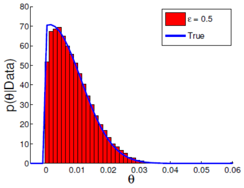
(a) Posterior density



(b) Gradient of log posterior



(c) SGLD



(d) SGLD + MH, $\epsilon = 0.5$.