Sampling in linear-Gaussian inverse problems
(is faster than regularized inversion)

Colin Fox fox@physics.otago.ac.nz
Richard A. Norton, J. Andrés Christen
Al Parker
Topics ...

- Sampling from multivariate Gaussians is numerical linear algebra
  - direct methods
  - iterative methods

- Sampling in linear-Gaussian hierarchical models (posterior NOT Gaussian)
  - A problem in image deblurring
  - Regularized inversion
  - Bayesian hierarchical model
  - Marginal then conditional sampling (new)
    * MCMC on posterior
    * Block Gibbs sampling
    * One block method
  - Timing

THM :: MCMC is barely needed, sampling can be faster than regularized deconvolution
Multivariate normal distributions

\[ \pi \left( x; \mu = A^{-1} b, \Sigma = A^{-1} \right) = \sqrt{\frac{\det(A)}{2\pi^n}} \exp \left\{ -\frac{1}{2} x^T A x + b^T x \right\} \]

Particularly interested in cases where precision matrix $A$ is sparse (GMRF) and $n$ large
Direct methods :: mutually conjugate vectors

**Definition 1** A set of non-zero vectors \( s^{(i)} \), \( i = 0, 1, \ldots, m - 1 \), is mutually conjugate w.r.t. \( A \) (SPD) if

\[
s^{(i)^T} A s^{(j)} = 0 \quad \forall i \neq j
\]

Note that \( s^{(i)^T} A s^{(i)} > 0 \) \( \forall i \), as \( A \) is strictly positive definite. Denote \( s^{(i)^T} A s^{(i)} = d_i \), hence, if \( S \in R^{m \times m} \) is the matrix with columns \( s^{(i)} \), \( i = 0, 1, \ldots, m - 1 \) and \( D = \text{diag} \left( d_0, d_1, \ldots, d_{m-1} \right) \) (also pd), then

\[
S^T A S = D
\]

Hence if \( z \sim N \left( 0, I_m \right) \), \( y = \sqrt{D^{-1}} z \sim N \left( 0, D^{-1} \right) \), and \( x = S y \sim N \left( 0, \Sigma \right) \). This follows since \( S^T A S = D \) so \( A^{-1} = S D^{-1} S^T \).

The expression for \( x \) can be written \( x = \sum_{i=0}^{m-1} \left( z_i / \sqrt{d_i} \right) s^{(j)} \). Since the \( z_i \) are i.i.d. \( \sim N \left( 0, 1 \right) \), this shows a sample from \( \pi \) may be generated using a sequence of standard normal random numbers.
Some examples you know of

Example 2  Cholesky factorization of $\Sigma$. Since $RR^T = \Sigma$, $R^TAR = I$, so the columns of $R$ are mutually conjugate w.r.t. $A$.

Example 3  Cholesky factorization of $A$. Since $LL^T = A$, $(L^T)^{-1}L^{-1} = A^{-1}$, so $(L^T)^{-1} = R$ and so the columns of $(L^T)^{-1}$ are mutually conjugate w.r.t. $A$.

Example 4  Eigen decomposition of $A$ or $\Sigma$. Consider

$$A = UDU^T$$

where $U$ is unitary (columns are normalised eigenvectors) and $D$ is diagonal (eigenvalues on diagonal). Then $U^TAU = D$, so the eigenvectors are mutually conjugate w.r.t. $A$.

These examples show that the algorithms based on the Cholesky and eigen factorizations are examples of the more general notion of mutually conjugate w.r.t. $A$, and also establishes that these algorithms do indeed sample from $\pi$. 
Generate conjugate vectors via CG

Algorithm 1 (CG sampler from $N(0, A^{-1})$). Given $b$ and $x^0$, let $r^0 = b^0 - Ax^0$, $p^0 = r^0$, $d_0 = p^{(0)T}Ap^0$, $y^0 = x^0$, and $k := 1$. Specify some stopping tolerance $\epsilon$.

Iterate:
1. $\gamma_{k-1} = \frac{r^{(k-1)T}r^{k-1}}{d_{k-1}}$ is the 1-D minimizer of $\phi$ in the direction $x^{k-1} + \gamma p^{k-1}$
2. $x^k = x^{k-1} + \gamma_{k-1}p^{k-1}$
3. Sample $z \sim N(0, 1)$, and set $y^k = y^{k-1} + \frac{z}{\sqrt{d_{k-1}}}p^{k-1}$
4. $r^k = -\nabla_x \phi(x^k) = r^{k-1} - \gamma_{k-1}Ap^{k-1}$ is the residual
5. $\beta_k = -\frac{r^{kT}r^k}{r^{(k-1)T}r^{k-1}}$
6. $p^k = r^k - \beta_kp^{k-1}$ is the next conjugate search direction.
7. $d_k = p^{(k)T}Ap^k$
8. Quit if $\|r^k\|^2 < \epsilon$. Else set $k := k + 1$ and go to step 1.

- Apart from step 3, this is exactly (linear) CG optimization
- $\text{Var}(x^k)$ is the CG polynomial

Parker & F SISC 2012
Best approximation property

Theorem 5 (Parker 2009)

The covariance matrix

$$\text{Var}(x^k|x^0, b^0) = V_k T_k^{-1} V_k^T$$

has $k$ non-zero eigenvalues which are the Lanczos estimates of the eigenvalues of $A^{-1}$, $\sigma_i(\text{Var}(x^k|x^0, b^0)) = \frac{1}{\theta_i^k}$. The eigenvectors of $\text{Var}(x^k|x^0, b^0)$ are the Ritz vectors $V_k v^i$ which estimate the eigenvectors of $A$. When $\|r^k\|_2 = 0$, then

$$\text{Var}(b^k|x^0, b^0) = V_k T_k V_k^T$$

and the $k$ non-zero eigenpairs of $\text{Var}(b^k|x^0, b^0)$ are the Lanczos Ritz pairs $(\theta_i^k, V_k v^i)$.

$\text{Var}(x^k|x^0, b^0)$ approximates $A^{-1}$ and $\text{Var}(b^k|x^0, b^0)$ approximates $A$ in the eigenspaces corresponding to the extreme and well separated eigenvalues of $A$. 

---

Parker & F SISC 2012
Finite precision CG vs Cholesky samples

**Fig. 4.2.** In the left panel is a CG sample $y^k \sim N(0, A^{-1})$ from a $10^4$-dimensional Gaussian over a 2D domain with a second order locally linear precision matrix. The realized variance $\text{Var}(y^k)$ accounts for 80% of the variability in $A^{-1}$. A Cholesky sample is shown in the right panel.

CG (by itself) over-smooths: initialize Gibbs with CG

Parker & F SISC 2012, Schneider & Willsky 2003
Iterative methods :: Gibbs sampling

Gibbs sampling\(^a\) repeatedly samples from (block) conditional distributions

Algorithm 1: One sweep of the component-wise Gibbs sampler targeting \(\pi(x)\)

\[
\text{for } i = 1, \ldots, n \text{ do} \\
\text{sample } z \sim \pi(x_i|x_{-i}) \\
x_i = z \\
\text{end}
\]

Forward and reverse sweep simulates a reversible kernel (self-adjoint wrt \(\pi\))

When is \(x \sim \pi^{(0)}\) is normal, then so is the k-step distribution

\[
A^{(k)} \rightarrow A \quad \Sigma^{(k)} \rightarrow \Sigma
\]

In what sense is “stochastic relaxation” related to “relaxation”?
What decomposition of \(A\) is this performing?

\(^a\)Glauber 1963 (heat-bath algorithm), Turcin 1971, Geman and Geman 1984
Matrix splitting form of stationary iterative methods

Want to solve

\[ Ax = b \]

The splitting \( A = M - N \) converts \( Ax = b \) to \( Mx = Nx + b \)

If \( M \) is nonsingular

\[ x = M^{-1}Nx + M^{-1}b \]

Iterative methods compute successively better approximations by

\[ x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b \]

\[ = Gx^{(k)} + g \]

Many splittings use terms in \( A = L + D + U \). Gauss-Seidel sets \( M = L + D \)

\[ x^{(k+1)} = -D^{-1}Lx^{(k+1)} - D^{-1}L^T x^{(k)} + D^{-1}b \]
Matrix formulation of Gibbs sampling from $\mathcal{N}(0, A^{-1})$

Let $y = (y_1, y_2, ..., y_n)^T$

Component-wise Gibbs updates each component in sequence from the (normal) conditional distributions

One ‘sweep’ over all $n$ components can be written

$$y^{(k+1)} = -D^{-1}Ly^{(k+1)} - D^{-1}L^Ty^{(k)} + D^{-1/2}z^{(k)}$$

where: $D = \text{diag}(A)$, $L$ is the strictly lower triangular part of $A$, $z^{(k-1)} \sim \mathcal{N}(0, I)$

$$y^{(k+1)} = Gy^{(k)} + c^{(k)}$$

$c^{(k)}$ is iid 'noise' with zero mean, finite covariance

Goodman & Sokal 1989
Matrix formulation of Gibbs sampling from $\mathbf{N}(0, \mathbf{A}^{-1})$

Let $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$

Component-wise Gibbs updates each component in sequence from the (normal) conditional distributions

One ‘sweep’ over all $n$ components can be written

$$\mathbf{y}^{(k+1)} = -\mathbf{D}^{-1}\mathbf{L}\mathbf{y}^{(k+1)} - \mathbf{D}^{-1}\mathbf{L}^T\mathbf{y}^{(k)} + \mathbf{D}^{-1/2}\mathbf{z}^{(k)}$$

where: $\mathbf{D} = \text{diag}(\mathbf{A})$, $\mathbf{L}$ is the strictly lower triangular part of $\mathbf{A}$, $\mathbf{z}^{(k-1)} \sim \mathbf{N}(0, \mathbf{I})$

$$\mathbf{y}^{(k+1)} = \mathbf{G}\mathbf{y}^{(k)} + \mathbf{c}^{(k)}$$

$c^{(k)}$ is iid 'noise' with zero mean, finite covariance

Spot the similarity to Gauss-Seidel iteration for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$

$$\mathbf{x}^{(k+1)} = -\mathbf{D}^{-1}\mathbf{L}\mathbf{x}^{(k+1)} - \mathbf{D}^{-1}\mathbf{L}^T\mathbf{x}^{(k)} + \mathbf{D}^{-1}\mathbf{b}$$

Goodman & Sokal 1989; Amit & Grenander 1991
Gibbs converges $\iff$ linear solver converges

**Theorem 6** Let $A = M - Ng$, $M$ invertible. The stationary linear solver

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b$$

$$= Gx^{(k)} + M^{-1}b$$

converges, if and only if the random iteration

$$y^{(k+1)} = M^{-1}Ny^{(k)} + M^{-1}c^{(k)}$$

$$= Gy^{(k)} + M^{-1}c^{(k)}$$

converges in distribution. Here $c^{(k)} \iid \pi_n$ has zero mean and finite variance

**Proof.** Both converge iff $\varrho(G) < 1$ \blacksquare

Any convergent splitting generates a convergent (generalized) Gibbs sampler

Error polynomials identical, hence convergence rates identical, hence acceleration methods transferable

Young 1971 Thm 3-5.1, Duflo 1997 Thm 2.3.18-4, Goodman & Sokal, 1989, Galli & Gao 2001

F Parker 2013
Gibbs samplers and equivalent linear solvers

Optimization ...

Gauss-Seidel
Cheby-GS
CG/Lanczos

Sampling ...

Gibbs
Cheby-Gibbs
Lanczos

Parker F SISC 2012
Chebyshev accelerated Gibbs sampling :: $d = 100$

$$[A]_{ij} = 10^{-4} \delta_{ij} + \begin{cases} 
  n_i & \text{if } i = j \\
  -1 & \text{if } i \neq j \text{ and } ||s_i - s_j||_2 \leq 1 \\
  0 & \text{otherwise}
\end{cases}$$

$\approx 10^4$ times faster
The inverse problem is to recover the ‘true’ unblurry image. We use the satellite (upper right) as PSF $k$, so \textit{semi-blind} deconvolution.

$$y = k \ast x + \eta = Ax + \eta$$

In the continuous setting this is the prototypical ill-posed inverse problem; $h$ is square integrable \implies $A$ is Hilbert-Schmidt \implies compact
Discrete problem reflects ill-posedness
Regularized Fourier deconvolution

Evaluate convolution by a circular convolution – no need to zero pad as image is black (hence periodic) at boundaries, giving efficient evaluation by FFT\(^a\).

\[ Ax = x \ast k \]

Introduce regularizing functional and regularizing parameter \( \lambda \). Solution \( \hat{x}_\lambda \) satisfies

\[ \hat{x}_\lambda = \arg \min_x \left\{ \| Ax - y \|^2 + \lambda x^T L x \right\} \]

or the normal (generalized deconvolution) equations

\[ \left( A^T A + \lambda L \right) \hat{x}_\lambda = A^T y \]

We use \( L \) is graph Laplacian on nearest-neighbour pixel lattice, with periodic boundaries\(^b\)

\[ Lx \equiv x \ast \begin{pmatrix} -1 \\ -1 & 4 & -1 \\ -1 \end{pmatrix} \]

\(^a\)Numerical Recipes, section 13.1 \(^b\)Bardsley SISC 2014
Choose regularizing parameter by L-curve criterion

\[ \lambda \]

Determining \( \lambda \) requires \( \approx 200 \) iterations/solves

0.517 seconds (0.507s for 200 solves) Matlab R2012b, Lenovo X230, Intel CORE i5

\( ^a \)P. C. Hansen (1992)
Bayesian hierarchical model

All unknowns treated as random variables, use same components as regularization

\[ y | x, \theta \sim N \left( Ax, (\gamma I)^{-1} \right) \]  \hspace{1cm} \text{(likelihood)}

\[ x | \theta \sim N \left( \mu, (\delta L)^{-1} \right) \]  \hspace{1cm} \text{(prior)}

\[ \theta = (\gamma, \delta) \sim \pi(\theta) \]  \hspace{1cm} \text{(hyperprior)}

where \( \gamma \) is precision of measurements, \( \delta \) is lumping constant in true image.

This is a common model in statistics: \( y = \text{observed data}, x = \text{latent field}, \theta = \text{hyperparameter} \)

Since

\[ \pi(y | x, \theta) = \frac{\gamma^{n/2}}{\sqrt{2\pi}} \exp \left\{ -\frac{\gamma}{2} \| Ax - y \|^2 \right\} \quad \text{and} \quad \pi(x | \theta) = \frac{\delta^{n/2} \sqrt{\det L}}{\sqrt{2\pi}} \exp \left\{ -\frac{\delta}{2} x^T L x \right\} \]

by Bayes rule

\[ \pi(x | y, \theta) \propto \exp \left\{ -\frac{\gamma}{2} \left( \| Ax - y \|^2 - \frac{\delta}{\gamma} x^T L x \right) \right\} \]

so \( x_{\text{MAP}} = \hat{x}_\lambda \) when \( \lambda = \delta/\gamma \) (though expectations average over \( \theta \) )
Posterior inference

The focus of inference is the posterior distribution

$$\pi(x, \theta | y) = \frac{\pi(y | x, \theta) \pi(x, \theta)}{\pi(y)}$$

Solutions and uncertainties are the posterior expectation of some function $\phi$ of $x$,

$$E[\phi(x)] = \int \phi(x) \pi(x, \theta | y) \, dx \, d\theta$$

which implicitly averages over the nuisance parameter $\theta$.

When $(x, \theta)^{(1)}, \ldots, (x, \theta)^{(N)} \sim \pi(x, \theta | y)$ are iterates of an ergodic Markov chain

$$E[\phi(x)] \approx \frac{1}{N} \sum_{i=1}^{N} \phi(x^{(i)})$$

with convergence guaranteed by a CLT.

Kipnis Varadhan Central limit theorem for additive functionals of reversible Markov processes and applications to simple exclusions Comm Math Phys 1986
Random-walk MCMC over the posterior

Most common is to form posterior distribution:

\[
\pi(x, \theta | y) = \frac{\pi(y | x, \theta) \pi(x | \theta) \pi(\theta)}{\pi(y)}
\]

The normalizing constant \( \pi(y) = \int \int \pi(y, x, \theta) \, dx \, d\theta \) is typically not available

Most applications perform MCMC with Metropolis-Hastings dynamics using random walk proposals (e.g. AM, MALA, HMC). This generates a Markov chain that converges in distribution to the posterior.

Convergence can be very slow due to correlations within the latent field \( x \), and between \( x \) and the hyperparameters \( \theta \).

Often need \( \sim 10^5 \) iterations for suitable convergence – to generate a sample for Monte Carlo estimates

Easy to implement – but gives MCMC it’s reputation for being very computationally expensive.
Some thoughts on Monte Carlo

John von Neumann (1951): *Any one who considers [Monte Carlo methods] is ... in a state of sin.*

Alan Sokal (1996): *Monte Carlo is an extremely bad method; it should be used only when all alternative methods are worse.*

Hammersley & Handscomb (1964): *... it will usually pay to scrutinize each part of a Monte Carlo experiment to see whether that part cannot be replaced by exact theoretical analysis ...*
Marginal then conditional sampling

First sample from the marginal posterior distribution over hyperparameters $\theta$

$$\pi (\theta | y) = \int \pi (x, \theta | y) \, dx$$

then from the full conditional distribution over $x$

---

**Algorithm 2**: MTC sampling from the marginal for $\theta$ then the full conditional for $x$

1. draw $\theta \sim \pi (\theta | y)$
2. draw $x \sim \pi (x | y, \theta)$

---

**Lemma 7**  Algorithm 2 generates a sample from the posterior distribution, i.e.,

$$(x, \theta) \sim \pi (x, \theta | y).$$

**Proof**. The density function over $x$ and $\theta$ is $\pi (x | y, \theta) \pi (\theta | y) = \pi (x, \theta | y)$ □

Independent $\theta$ gives independent $(x, \theta)$
Marginal posterior for $\theta$

Lemma 8

$$\pi (\theta | y) = \frac{\pi (y | \theta, x) \pi (x | \theta) \pi (\theta)}{\pi (x | \theta, y) \pi (y)}$$

Proof. $\pi (x, y, \theta) = \pi (x | \theta, y) \pi (y | \theta) \pi (\theta)$ and $\pi (x, y, \theta) = \pi (y | x, \theta) \pi (x | \theta) \pi (\theta)$. Since $\pi (y) \neq 0$, the result follows. \blacksquare

Can eliminate $x$ to give

(general Gaussian-linear model: $\Sigma = \text{noise covariance}$, $Q = \text{prior precision}$)

$$\pi (\theta | y) \propto \sqrt{\frac{\det (\Sigma^{-1}) \det (Q)}{\det (Q + A^T \Sigma^{-1} A)}} \exp \left\{ -\frac{1}{2} (y - A\mu)^T \Sigma^{-1} A \left[ (A^T \Sigma^{-1} A)^{-1} - (A^T \Sigma^{-1} A + Q)^{-1} \right] A^T \Sigma^{-1} (y - A\mu) \right\} \pi (\theta).$$

(1)
**Full conditional for** \( x \)

For the linear-Gaussian hierarchical model the full conditional for \( x \) is the multivariate normal

\[
x|y, \theta \sim N \left( \mu_{x|y,\theta}, Q_{x|y,\theta}^{-1} \right)
\]

where

\[
\mu_{x|y,\theta} = \mu + \left( Q + A^T \Sigma^{-1} A \right)^{-1} A^T \Sigma^{-1} (y - A \mu)
\]

\[
Q_{x|y,\theta} = Q + A^T \Sigma^{-1} A.
\]

I have omitted the dependence of matrices on \( \theta \) for brevity.

For the Jupiter example

\[
x|\theta, y \sim N \left( (A^T A + (\delta/\gamma) L)^{-1} A^T y, (\gamma A^T A + \delta L)^{-1} \right)
\]

An independent sample from this distribution may be computed in \( O(n \log n) \) operations by solving (the generalized deconvolution eqns)

\[
\left( \gamma A^T A + \delta L \right) x = \gamma A^T y + w
\]

where \( w = v_1 + v_2 \) with independent \( v_1 \sim N \left( 0, \gamma A^T A \right) \) and \( v_2 \sim N \left( 0, \delta L \right) \)
**Block Gibbs sampling**

Cycles through sampling from full conditional distributions (aka Glauber, heat bath)

**Algorithm 3:** Gibbs sampling algorithm with blocking of the latent field

at state $x, \theta = (\gamma, \delta)$

- draw $x|\gamma, \delta, y \sim N \left( (A^T A + (\delta/\gamma)L)^{-1} A^T y, (\gamma A^T A + \delta L)^{-1} \right)$
- draw $\gamma|x, \delta, y \sim \Gamma \left( \frac{n}{2} + \lambda \gamma, \frac{1}{2} \|Ax - y\|^2 + \beta \gamma \right)$
- draw $\delta|x, \gamma, y \sim \Gamma \left( \frac{n}{2} + \lambda \delta, \frac{1}{2} \|Ax - y\|^2 + \beta \delta \right)$

Most of the computational cost is the $O(n \log n)$ draw from the large Gaussian latent field

*Conjugate priors* $\pi(\gamma) = \Gamma(\alpha_\gamma, \beta_\gamma)$ and $\pi(\delta) = \Gamma(\alpha_\delta, \beta_\delta)$ make full conditionals available: (shape and scale parameters chosen, as Bardsley, to be “uninformative”)

$$\lambda|x, \delta, y \sim \Gamma \left( \frac{n}{2} + \alpha_\lambda, \frac{1}{2} \|Ax - y\|^2 + \beta_\lambda \right)$$

$$\delta|x, \lambda, y \sim \Gamma \left( \frac{n}{2} + \alpha_\delta, \frac{1}{2} \|Ax - y\|^2 + \beta_\delta \right)$$
One-block sampler

Algorithm 4: One-block algorithm

at state $x, \theta$

draw $\theta' \sim q(\theta'|\theta)$

draw $x' \sim \pi(x'|\theta', y)$

accept $(x', \theta')$ w.p. $\alpha((x, \theta) \rightarrow (x', \theta')) = 1 \land \frac{\pi(x', \theta'|y) \pi(x|\theta, y) q(\theta'|\theta)}{\pi(x, \theta|y) \pi(x'|\theta', y) q(\theta'|\theta)}$

otherwise reject

Transition kernel for the hyperparameter $\theta$ targets $\pi(\theta|y)$ since

$$\frac{\pi(x', \theta'|y) \pi(x|\theta, y) q(\theta'|\theta)}{\pi(x, \theta|y) \pi(x'|\theta', y) q(\theta'|\theta)} = \frac{\pi(\theta'|y) q(\theta'|\theta)}{\pi(\theta|y) q(\theta'|\theta)}.$$

Makes larger steps in marginal for $\theta$, rather than conditional $\theta|x$. 
Marginal then conditional (MTC) sampler

Implement MTC sampler by performing MH MCMC on $\pi(\theta|y)$ to get (effectively) independent sample $\theta \sim \pi(\theta|y)$, then draw $x \sim \pi(x|\theta, y)$ to get (effectively) independent sample $(x, \theta)$ from the full posterior distribution. Evaluate MH ratio using $\pi(\theta|y)$

**Algorithm 5:** Metropolis-Hastings algorithm on $\pi(\theta|y)$

at state $\theta$

draw $\theta' \sim q(\theta'|\theta)$

accept $\theta'$ w.p. $\alpha(\theta \rightarrow \theta') = 1 \wedge \frac{\pi(\theta'|y) q(\theta|\theta')}{\pi(\theta|y) q(\theta'|\theta)}$

otherwise reject

General MCMC only over low-dimensional space of hyperparameters
high correlations between $\theta$ and $x$ are irrelevant

General linear-Gaussian model requires ratios of determinants of $\Sigma^{-1}$, $Q$ and $Q + A^T \Sigma^{-1} A$, and differences of arguments of the exponential.
\( \mathcal{O}(1) \) evaluation of ratio of determinants

For our Jupiter example the marginal posterior for \( \theta \) simplifies to

\[
\pi(\theta | y) \propto \delta^{n/2} \exp \left( -\frac{1}{2} g(\lambda) - \frac{\gamma}{2} f(\lambda) \right) \pi(\theta)
\]

\[
\lambda = \frac{\delta}{\gamma}, \quad f(\lambda) = (A^T y)^T ((A^T A)^{-1} - (A^T A + \lambda L)^{-1})(A^T y) \quad \text{and} \quad g(\lambda) = \log |A^T A + \lambda L|
\]

A range of techniques can be used to give accurate approximation and efficient computation.
Autocorrelation of $\lambda = \delta / \gamma$ for all four sampling algorithms
Posterior sample and histograms

Left; image component of a posterior sample, using the MTC Option 2 algorithm
Right; marginal posterior histograms for $\gamma$, $\delta$ and $\delta/\gamma$
Computational efficiency

computing cost per effective sample \(\text{CCES} = \frac{\tau T}{N}\)

\(T\) is the total (on-line) compute time

\(N\) is the length of chain

integrated autocorrelation time \(\tau = 1 + 2 \sum_{k=1}^{\infty} \rho_k\)

length of chain with same variance reducing power as one independent sample

Table 1: Compute times and CCES (in seconds) for \(10^4\) steps of sampling algorithms

<table>
<thead>
<tr>
<th></th>
<th>burn in</th>
<th>total time</th>
<th>acceptance rate</th>
<th>CCES for (\lambda)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block Gibbs</td>
<td>60</td>
<td>83.1</td>
<td>1</td>
<td>0.17</td>
</tr>
<tr>
<td>One block</td>
<td>20</td>
<td>127.8</td>
<td>0.33</td>
<td>0.090</td>
</tr>
<tr>
<td>MTC Option 1</td>
<td>20</td>
<td>63.1</td>
<td>0.33</td>
<td>0.050</td>
</tr>
<tr>
<td>MTC Option 2</td>
<td>20</td>
<td>11.8</td>
<td>0.46</td>
<td>0.015</td>
</tr>
</tbody>
</table>
## Integrated autocorrelation times

Table 2: Integrated autocorrelation times (in iterations) of various statistics of interest

<table>
<thead>
<tr>
<th></th>
<th>$\gamma$</th>
<th>$\delta$</th>
<th>$\lambda = \delta / \gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block Gibbs</td>
<td>1.6</td>
<td>22.3</td>
<td>21.0</td>
</tr>
<tr>
<td>One block</td>
<td>7.8</td>
<td>6.7</td>
<td>7.1</td>
</tr>
<tr>
<td>MTC Option 1</td>
<td>7.6</td>
<td>7.8</td>
<td>7.9</td>
</tr>
<tr>
<td>MTC Option 2</td>
<td>2.1</td>
<td>5.0</td>
<td>5.7</td>
</tr>
</tbody>
</table>
### Compute time for images

Table 3: Compute time (seconds) for a regularized image or independent image sample

<table>
<thead>
<tr>
<th></th>
<th>time to $\lambda$</th>
<th>time to $x$</th>
<th>total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regularization</td>
<td>0.52</td>
<td>0.0024</td>
<td>0.52</td>
</tr>
<tr>
<td>Block Gibbs</td>
<td>0.85</td>
<td>-</td>
<td>0.85</td>
</tr>
<tr>
<td>One block</td>
<td>0.44</td>
<td>0.0020</td>
<td>0.47</td>
</tr>
<tr>
<td>MTC Option 1</td>
<td>0.23</td>
<td>0.0096</td>
<td>0.24</td>
</tr>
<tr>
<td>MTC Option 2</td>
<td>0.037</td>
<td>0.0082</td>
<td>0.045</td>
</tr>
</tbody>
</table>
Conclusions

- GS ≡ GS
- Numerical linear algebra can really speed up sampling
- Structuring the stochastic calculation can substantially reduce the computational task
- Marginal posterior over hyperparameters allows selection/sampling before deconvolution
- We factorized the problem into a large Gaussian/linear part and a small non-Gaussian/non-linear part
- Small part independent of discretization or data size
- ... hence can perform inference over function space without approximation
- Trace class prior covariance potentially gives analytic ratio of determinants
Thank You

 Thanks also to Dan Simpson, Harvard Rue