NTNU
Norwegian University of
Science and Technology

How important are Cholesky factorisations for performing computations with large Gaussians?

Daniel Simpson

Erlend Aune, Jo Eidsvik, Håvard Rue (NTNU)
Ian Turner, Chris Strickland, Tony Pettitt (QUT)

NTNU
Norwegian University of
Science and Technology

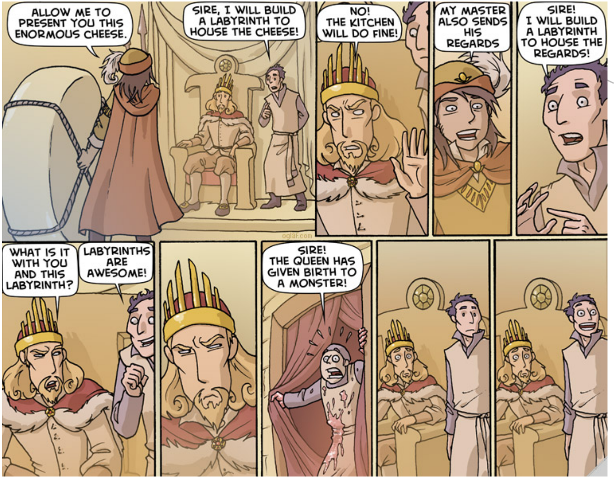
Another $%#&ing Sampling Talk

Daniel Simpson

Erlend Aune, Jo Eidsvik, Håvard Rue (NTNU)
Ian Turner, Chris Strickland, Tony Pettitt (QUT)

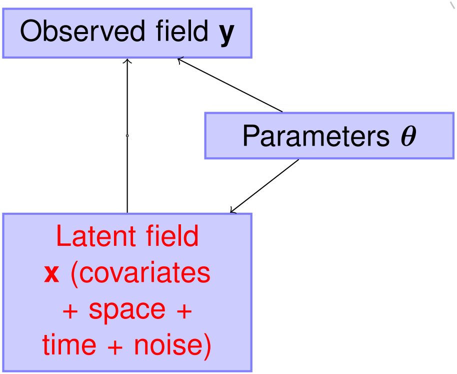
Outline

“The minotaur justifies the labyrinth”—Jorge Luis Borges


ALLOW ME TO PRESENT YOU THIS ENORMOUS CHEESE.
SIRE, I WILL BUILD A LABYRINTH TO HOUSE THE CHEESE!
NO! THE KITCHEN WILL DO FINE!
MY MASTER ALSO SENDS HIS REGARDS
SIRE! I WILL BUILD A LABYRINTH TO HOUSE THE REGARDS!
oglaf.com
WHAT IS IT WITH YOU AND THIS LABYRINTH?
LABYRINTHS ARE AWESOME!
SIRE! THE QUEEN HAS GIVEN BIRTH TO A MONSTER!

`http://oglaf.com/labyrinth/ (NSFW)`

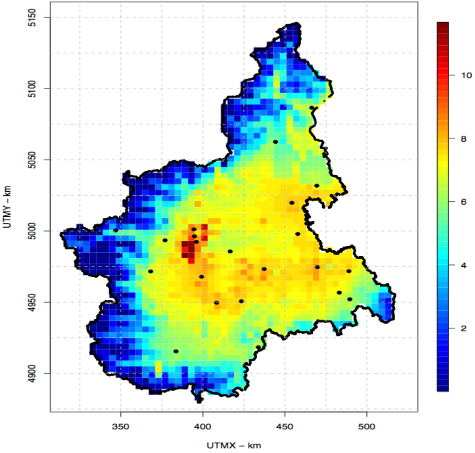
Scream and run away


Observed field $\mathbf{y}$
Parameters θ
Latent field $\mathbf{x}$ (covariates + space + time + noise)

Question: Can we infer $\mathbf{x}$ and θ from $\mathbf{y}$ when the sizes of $\mathbf{x}$ and $\mathbf{y}$ are large?

Against pollution

PM-10 concentration in Piemonte, Italy



The point of it all

A very ugly likelihood

The likelihood *in the most boring case* is

$$\log(\pi(Y|\eta)) = |\Omega| - \int_{\Omega} \Lambda(s)\, ds + \sum_{s_i \in Y} \Lambda(s_i),$$

where Y is the set of observed locations and $\Lambda(s) = \exp(Z(s))$, and $Z(s)$ is a Gaussian random field.

Add it up

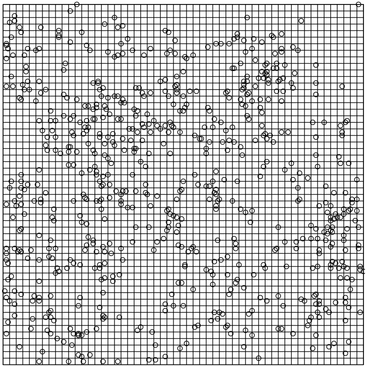
NB: *The number of points in a region R is Poisson distributed with mean* $\int_R \Lambda(s)\, ds$.

- Divide the 'observation window' into rectangles.
- Let y_i be the number of points in rectangle i.

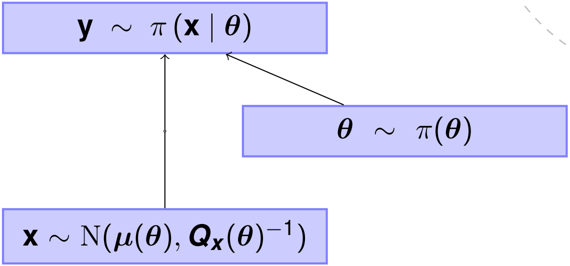
$$y_i | x_i, \boldsymbol{\theta} \sim Po(e^{x_i}),$$

- The log-risk surface is replaced with

$$\mathbf{x} | \boldsymbol{\theta} \sim N(\boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{Q}(\boldsymbol{\theta})^{-1}).$$



Makes an ass out of you and umption

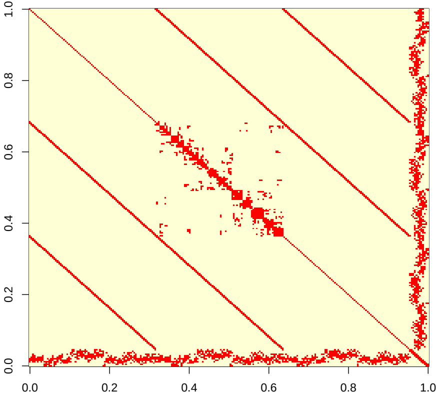

$\mathbf{y} \sim \pi(\mathbf{x} \mid \theta)$
$\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$
$\mathbf{x} \sim \mathrm{N}(\boldsymbol{\mu}(\theta), \boldsymbol{Q}_{\boldsymbol{x}}(\theta)^{-1})$

$$\pi(\boldsymbol{x}|\boldsymbol{y}) \propto \pi(\boldsymbol{y} \mid \boldsymbol{x}, \theta)\pi(\boldsymbol{x} \mid \theta)\pi(\theta)$$

HOW DO WE DO THIS WHEN THE DIMENSIONS OF **x** AND **y** ARE HUGE?

Knowing me, knowing you

What does the precision matrix (usually) look like?



NB: It's good to consider the whole (jointly) Gaussian part: fixed + random effects + noise.

Gimme! Gimme! Gimme! (A man after midnight)

So what do we want?

We are typically interested in MCMC proposals that require:

— A *single* sample from $\boldsymbol{x} \sim N(\boldsymbol{\mu}, \boldsymbol{Q}^{-1})$; or

— A *sequence* of samples from $\boldsymbol{x} \sim N(\boldsymbol{\mu}_i, \boldsymbol{Q}_i^{-1})$.

— Possible the log-density

It is often (usually) the case that the precision matrix of the Gaussian that we are sampling from changes at every MCMC sweep (due to parameter updating or local approximation to the posterior).

The village green preservation society

Direct methods

All methods from sampling from a Gaussian require a factorisation of the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{R}\boldsymbol{R}^T$ or the precision matrix $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1} = \boldsymbol{L}\boldsymbol{L}^T$. This is always† done with a Cholesky factorisation.

- Making these factorisations computationally feasible is the main aim of modern spatial statistics.
- Sparse matrices (aka models with the Markov property) are necessary‡ to make large problems feasible.
- This was the (computational) state of the art 10 years ago and, with some minor blips, it is still the state of the art.

Outline

Danicing on my own (2005–2008)

I am Loki of Asgard, and I am burdened with glorious porpoise.

“Oh those Russians!”—Indirect methods

For spatial problems in applied mathematics, physics, and engineering, direct methods are typically overlooked in favour of *iterative methods*.

- — The are a huge variety of *Krylov subspace* methods, the most famous being the Conjugate Gradient method (also GMRES, BiCG-Stab, LSQR,...)
- — These *do not* require the matrix, but rather access to matrix-vector products of the form $\boldsymbol{Qx}$.
- — Typically, these methods are exact if you run them for long enough, and they converge superlinearly in subspace size.

Ignition

Sampling from a Gaussian

If we have a factorisation of the precision matrix $\boldsymbol{Q} = \boldsymbol{L}\boldsymbol{L}^{\boldsymbol{T}}$, then it's easy to see that $\boldsymbol{x} = \boldsymbol{L}^{-T}\boldsymbol{z}$, z i.i.d. standard normal, is a sample from $N(\mathbf{0}, \boldsymbol{Q}^{-1})$.

- We need a version of $\boldsymbol{L}^{-T}\boldsymbol{z}$ that we can compute using only matrix-vector products from $\boldsymbol{Q}$.
- The standard is to take $\boldsymbol{L}$ to be the Cholesky triangle of $\boldsymbol{Q}$, but maybe this isn't a good idea....

Dumb things

If we can diagonalise $\boldsymbol{Q} = \boldsymbol{O}\boldsymbol{\Lambda}\boldsymbol{O}^T$, then we can take $\boldsymbol{Q} = \boldsymbol{L}\boldsymbol{L}^T$ where

$$\boldsymbol{L} = \boldsymbol{Q}^{1/2} \equiv \boldsymbol{O} \begin{pmatrix} \lambda_1^{1/2} & & \\ & \ddots & \\ & & \lambda_n^{1/2} \end{pmatrix} \boldsymbol{O}^T.$$

- It's easy to see that $\boldsymbol{x} = \boldsymbol{Q}^{-1/2}\boldsymbol{z}$ has the correct precision matrix
- Clearly this is a stupid thing to do!

Not that kind of girl

Idea

It's easy to show that if $\boldsymbol{Q} = \boldsymbol{VTV}^T$, $\boldsymbol{V}$ orthogonal, $\boldsymbol{T}$ tridiagonal, then

$$\boldsymbol{Q}^{-1/2} z = \boldsymbol{V}\boldsymbol{T}^{-1/2}\boldsymbol{V}^T\boldsymbol{z}.$$

What if we don't go all the way??

— Try to pick smart directions that know about $\boldsymbol{Q}$ and $\boldsymbol{z}$.

— Analogy with CG and other Krylov methods for linear systems

— Analogy with partial least squares...

Hallo Spaceboy

Definition (Krylov Subspace)

The *m*-dimensional Krylov Subspace generated by $\boldsymbol{Q}$ and $\boldsymbol{z}$ is defined by

$$\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{z}) = span\left\{\boldsymbol{z}, \boldsymbol{A}\boldsymbol{z}, \boldsymbol{A}^2\boldsymbol{z}, \ldots, \boldsymbol{A}^{m-1}\boldsymbol{z}\right\}.$$

Hallo Spaceboy

Definition (Krylov Subspace)

The *m*-dimensional Krylov Subspace generated by $\boldsymbol{Q}$ and $\boldsymbol{z}$ is defined by

$$\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{z}) = span\left\{\boldsymbol{z}, \boldsymbol{A}\boldsymbol{z}, \boldsymbol{A}^2\boldsymbol{z}, \ldots, \boldsymbol{A}^{m-1}\boldsymbol{z}\right\}.$$

— Defined as all vectors of the form $p_{m-1}(\boldsymbol{Q})\boldsymbol{z}$, where p_{m-1} is any polynomial of degree $m - 1$.

Hallo Spaceboy

Definition (Krylov Subspace)

The m-dimensional Krylov Subspace generated by $\boldsymbol{Q}$ and $\boldsymbol{z}$ is defined by

$$\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{z}) = span\left\{\boldsymbol{z}, \boldsymbol{A}\boldsymbol{z}, \boldsymbol{A}^2\boldsymbol{z}, \dots, \boldsymbol{A}^{m-1}\boldsymbol{z}\right\}.$$

- Defined as all vectors of the form $p_{m-1}(\boldsymbol{Q})\boldsymbol{z}$, where p_{m-1} is any polynomial of degree $m-1$.
- The basis given in the definition is useless for computation!

Hallo Spaceboy

Definition (Krylov Subspace)

The *m*-dimensional Krylov Subspace generated by $\boldsymbol{Q}$ and $\boldsymbol{z}$ is defined by

$$\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{z}) = span\left\{\boldsymbol{z}, \boldsymbol{A}\boldsymbol{z}, \boldsymbol{A}^2\boldsymbol{z}, \ldots, \boldsymbol{A}^{m-1}\boldsymbol{z}\right\}.$$

- Defined as all vectors of the form $p_{m-1}(\boldsymbol{Q})\boldsymbol{z}$, where p_{m-1} is any polynomial of degree $m-1$.
- The basis given in the definition is useless for computation!
- If we find the *best* approximation to $\boldsymbol{Q}^{-1/2}\boldsymbol{z}$ in $\mathcal{K}_m(\boldsymbol{Q}, \boldsymbol{z})$.... best polynomial approximation...

Ace of Bas(is)

Theorem (Lanczos Decomposition)

If $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m\}$ *forms an ONB for* $\mathcal{K}_m(\boldsymbol{Q}, \boldsymbol{z})$*, then the matrix* $\boldsymbol{V}_m = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m]$*, with* $v_1 = z/\left\|z\right\|$*, satisfies*

$$\boldsymbol{Q}\boldsymbol{V}_m = \boldsymbol{V}_m\boldsymbol{T}_m + \beta_m \boldsymbol{v}_{m+1}\boldsymbol{e}_m^T,$$

where $\boldsymbol{V}_m^T\boldsymbol{v}_{m+1} = \boldsymbol{0}$ *and* $\boldsymbol{T}_m = \boldsymbol{V}_m^T\boldsymbol{Q}\boldsymbol{V}_m$.

Ace of Bas(is)

Theorem (Lanczos Decomposition)

If $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m\}$ *forms an ONB for* $\mathcal{K}_m(\boldsymbol{Q}, \boldsymbol{z})$*, then the matrix* $\boldsymbol{V}_m = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m]$*, with* $v_1 = z/\left\|z\right\|$*, satisfies*

$$\boldsymbol{Q}\boldsymbol{V}_m = \boldsymbol{V}_m\boldsymbol{T}_m + \beta_m\boldsymbol{v}_{m+1}\boldsymbol{e}_m^T,$$

where $\boldsymbol{V}_m^T\boldsymbol{v}_{m+1} = \boldsymbol{0}$ *and* $\boldsymbol{T}_m = \boldsymbol{V}_m^T\boldsymbol{Q}\boldsymbol{V}_m$.

— The columns of $\boldsymbol{V}_m \in \mathbb{R}^{n\times m}$ form an orthonormal basis for $\mathcal{K}_m(\boldsymbol{Q}, \boldsymbol{z})$.

Ace of Bas(is)

Theorem (Lanczos Decomposition)

If $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m\}$ *forms an ONB for* $\mathcal{K}_m(\boldsymbol{Q}, \boldsymbol{z})$*, then the matrix* $\boldsymbol{V}_m = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m]$*, with* $v_1 = z/\, \|z\|$*, satisfies*

$$\boldsymbol{Q}\boldsymbol{V}_m = \boldsymbol{V}_m\boldsymbol{T}_m + \beta_m\boldsymbol{v}_{m+1}\boldsymbol{e}_m^T,$$

where $\boldsymbol{V}_m^T\boldsymbol{v}_{m+1} = \boldsymbol{0}$ *and* $\boldsymbol{T}_m = \boldsymbol{V}_m^T\boldsymbol{Q}\boldsymbol{V}_m$*.*

— The columns of $\boldsymbol{V}_m \in \mathbb{R}^{n\times m}$ form an orthonormal basis for $\mathcal{K}_m(\boldsymbol{Q}, \boldsymbol{z})$.

— T_m is small ($m \times m$), symmetric and *tridiagonal*.

Little people

So, after computing the Lanczos decomposition, we have the estimate

$$\boldsymbol{x}_m = \boldsymbol{V}_m \boldsymbol{T}_m^{-1/2} \boldsymbol{V}_m^T \boldsymbol{z}.$$

How do we compute the inverse square root of $\boldsymbol{T}_m$??

— Eigendecomposion. $\mathcal{O}(m^3)$.

Little people

So, after computing the Lanczos decomposition, we have the estimate

$$\boldsymbol{x}_m = \boldsymbol{V}_m \boldsymbol{T}_m^{-1/2} \boldsymbol{V}_m^T \boldsymbol{z}.$$

How do we compute the inverse square root of $\boldsymbol{T}_m$??

— Eigendecomposion. $\mathcal{O}(m^3)$.

— Rational approximation

$$\boldsymbol{T}_m^{-1/2} \boldsymbol{e}_1 \approx \sum_{i=1}^{p} \alpha_i (w_i \boldsymbol{I} + \boldsymbol{T}_m)^{-1} \boldsymbol{e}_1$$

Little people

So, after computing the Lanczos decomposition, we have the estimate

$$\boldsymbol{x}_m = \boldsymbol{V}_m \boldsymbol{T}_m^{-1/2} \boldsymbol{V}_m^T \boldsymbol{z}.$$

How do we compute the inverse square root of $\boldsymbol{T}_m$??

— Eigendecomposion. $\mathcal{O}(m^3)$.

— Rational approximation

$$\boldsymbol{T}_m^{-1/2} \boldsymbol{e}_1 \approx \sum_{i=1}^{p} \alpha_i (w_i \boldsymbol{I} + \boldsymbol{T}_m)^{-1} \boldsymbol{e}_1$$

— Best rational approximation (Zolotarev): $\alpha_i, w_i > 0$ given explicitly in terms of Jacobi elliptic functions. (See Hale, Higham and Trefethen 2008)

Little people

So, after computing the Lanczos decomposition, we have the estimate

$$\boldsymbol{x}_m = \boldsymbol{V}_m \boldsymbol{T}_m^{-1/2} \boldsymbol{V}_m^T \boldsymbol{z}.$$

How do we compute the inverse square root of $\boldsymbol{T}_m$??

— Eigendecomposion. $\mathcal{O}(m^3)$.

— Rational approximation

$$\boldsymbol{T}_m^{-1/2} \boldsymbol{e}_1 \approx \sum_{i=1}^{p} \alpha_i (w_i \boldsymbol{I} + \boldsymbol{T}_m)^{-1} \boldsymbol{e}_1$$

— Best rational approximation (Zolotarev): $\alpha_i, w_i > 0$ given explicitly in terms of Jacobi elliptic functions. (See Hale, Higham and Trefethen 2008)

— Actually, can do this directly for $\boldsymbol{Q}$. Converges geometrically in p $(= \mathcal{O}(\log(\kappa_2(\boldsymbol{Q}))))$

The Lanczos Decomposition

Input: A, z, and m.
Output: V_m and T_m.

```
Set $v_1 = z / \|z\|$.
for $j = 1 : m$ do
    $q = Av_j$.
    if $j \neq 1$ then
        $q = q - \beta_{j-1} v_{j-1}$.
    end
    $\alpha_j = v_j^T q$
    $q = q - \alpha_j v_j$
    $\beta_j = \|q\|_2$
    $v_{j+1} = q / \beta_j$
end
Set $x_m = \boldsymbol{V}_m \boldsymbol{T}_m^{-1/2} \boldsymbol{V}_m^T \boldsymbol{z}$.
```

Just like you imagined

Theorem

Let $\boldsymbol{x}_m$ be the sample produced in the mth step of the Krylov sampler and let $\boldsymbol{x} = \boldsymbol{Q}^{-1/2}\boldsymbol{z}$ be the true sample from $\boldsymbol{x} \sim N(\boldsymbol{0}, \boldsymbol{Q}^{-1})$. If $\boldsymbol{r}_m$ is the residual at the mth iteration of the conjugate gradient method for solving $\boldsymbol{Qy} = \boldsymbol{z}$, then

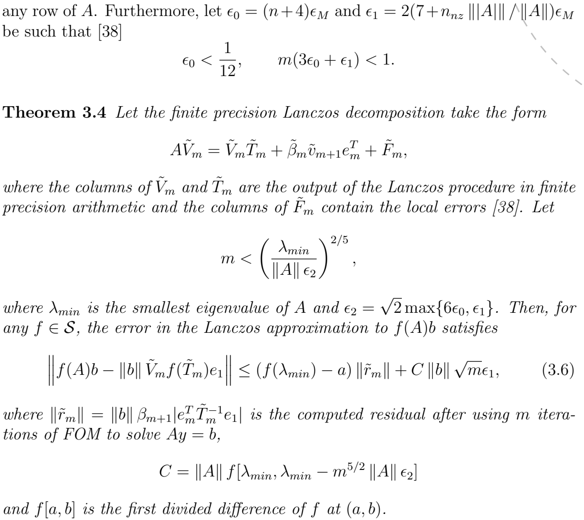
$$\|\boldsymbol{x} - \boldsymbol{x}_m\| \leq \lambda_{min}^{-1/2} \|\boldsymbol{r}_m\|, \tag{1}$$

where λ_{min} is the smallest eigenvalue of $\boldsymbol{Q}$. Furthermore, the following a priori bound holds:

$$\|\boldsymbol{x} - \boldsymbol{x}_m\| \leq 2\lambda_{min}^{-1/2}\sqrt{\kappa}\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^m \|z\|, \tag{2}$$

and $\kappa = \lambda_{max}/\lambda_{min}$ is the condition number of $\boldsymbol{Q}$.

Float on!



any row of A. Furthermore, let $\epsilon_0 = (n+4)\epsilon_M$ and $\epsilon_1 = 2(7+n_{nz}\,\| |A| \| / \|A\|)\epsilon_M$ be such that [38]

$$\epsilon_0 < \frac{1}{12}, \qquad m(3\epsilon_0 + \epsilon_1) < 1.$$

Theorem 3.4 *Let the finite precision Lanczos decomposition take the form*

$$A\tilde{V}_m = \tilde{V}_m \tilde{T}_m + \tilde{\beta}_m \tilde{v}_{m+1} e_m^T + \tilde{F}_m,$$

where the columns of $\tilde{V}_m$ and $\tilde{T}_m$ are the output of the Lanczos procedure in finite precision arithmetic and the columns of $\tilde{F}_m$ contain the local errors [38]. Let

$$m < \left(\frac{\lambda_{min}}{\|A\|\, \epsilon_2} \right)^{2/5},$$

where λ_{min} is the smallest eigenvalue of A and $\epsilon_2 = \sqrt{2}\max\{6\epsilon_0, \epsilon_1\}$. Then, for any $f \in \mathcal{S}$, the error in the Lanczos approximation to $f(A)b$ satisfies

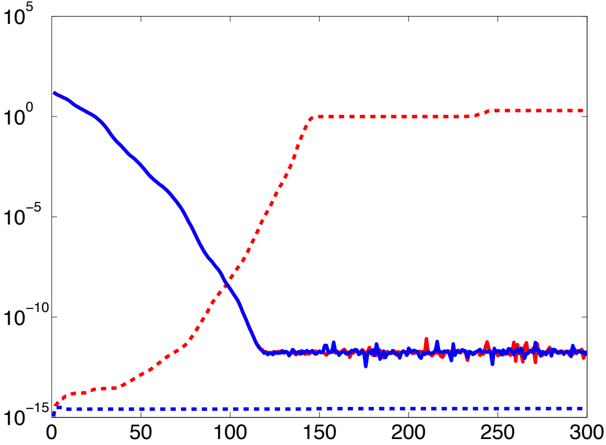
$$\left\| f(A)b - \|b\|\, \tilde{V}_m f(\tilde{T}_m) e_1 \right\| \leq (f(\lambda_{min}) - a)\, \|\tilde{r}_m\| + C\, \|b\|\, \sqrt{m}\epsilon_1, \qquad (3.6)$$

where $\|\tilde{r}_m\| = \|b\|\, \beta_{m+1} |e_m^T \tilde{T}_m^{-1} e_1|$ is the computed residual after using m iterations of FOM to solve $Ay = b$,

$$C = \|A\|\, f[\lambda_{min}, \lambda_{min} - m^{5/2}\, \|A\|\, \epsilon_2]$$

and $f[a, b]$ is the first divided difference of f at (a, b).

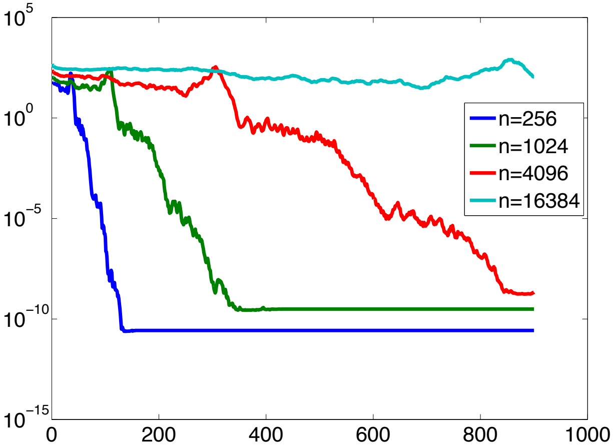
Running up that hill



I like big buts

But...

No easy way down



When all is said and done

What went wrong?

— The matrix in question was the precision for a CAR(2) process (essentially the discrete biharmonic operator)

When all is said and done

What went wrong?

- The matrix in question was the precision for a CAR(2) process (essentially the discrete biharmonic operator)
- The rate of convergence depends on the condition number of $\boldsymbol{Q}$, which is $\mathcal{O}(h^4)$.

When all is said and done

What went wrong?

- The matrix in question was the precision for a CAR(2) process (essentially the discrete biharmonic operator)
- The rate of convergence depends on the condition number of $\boldsymbol{Q}$, which is $\mathcal{O}(h^4)$.
- This is a standard problem with Krylov subspace methods.

Your other man

A variety of other things were tried:

- 2–pass Lanczos: Compute $\boldsymbol{T}_m^{-1/2}\boldsymbol{e}_1$ in one sweep, compute $\boldsymbol{x}_m$ in the second.
  - Very fast!
  - Low storage (3 vectors rather than m).
  - Twice the work.

Your other man

A variety of other things were tried:

- 2–pass Lanczos: Compute $\boldsymbol{T}_m^{-1/2}\boldsymbol{e}_1$ in one sweep, compute $\boldsymbol{x}_m$ in the second.
  - Very fast!
  - Low storage (3 vectors rather than m).
  - Twice the work.
- Using a different basis $\mathcal{K}_m\left((\xi \boldsymbol{I} - \boldsymbol{Q})^{-1}, \boldsymbol{z}\right)$, $\mathcal{K}_m(\boldsymbol{Q}, \boldsymbol{z}) \cup \mathcal{K}_m(\boldsymbol{Q}^{-1}, \boldsymbol{z})$

Your other man

A variety of other things were tried:

— 2–pass Lanczos: Compute $\boldsymbol{T}_m^{-1/2}\boldsymbol{e}_1$ in one sweep, compute $\boldsymbol{x}_m$ in the second.

- Very fast!
- Low storage (3 vectors rather than m).
- Twice the work.

— Using a different basis $\mathcal{K}_m\left((\xi \boldsymbol{I} - \boldsymbol{Q})^{-1}, \boldsymbol{z}\right)$, $\mathcal{K}_m(\boldsymbol{Q}, \boldsymbol{z}) \cup \mathcal{K}_m(\boldsymbol{Q}^{-1}, \boldsymbol{z})$

— Rational approximation.

Your other man

A variety of other things were tried:

- 2–pass Lanczos: Compute $\boldsymbol{T}_m^{-1/2}\boldsymbol{e}_1$ in one sweep, compute $\boldsymbol{x}_m$ in the second.
  - Very fast!
  - Low storage (3 vectors rather than m).
  - Twice the work.
- Using a different basis $\mathcal{K}_m\left((\xi\boldsymbol{I}-\boldsymbol{Q})^{-1},\boldsymbol{z}\right)$, $\mathcal{K}_m(\boldsymbol{Q},\boldsymbol{z})\cup\mathcal{K}_m(\boldsymbol{Q}^{-1},\boldsymbol{z})$
- Rational approximation.
- “Least-squares sampling” (i.e. the simplest of John’s methods)

Outline

I’m so glad

Stand by your manatee

Some obvious points...

- When solving linear systems, the solution is to *precondition* the system, i.e. find $\boldsymbol{F}\boldsymbol{F}^T \approx \boldsymbol{Q}$ and solve $\boldsymbol{F}^{-1}\boldsymbol{Q}\boldsymbol{F}^{-T}y = \boldsymbol{F}^{-1}\boldsymbol{z}$.
- A preconditioner is *optimal* if the condition number remains $\mathcal{O}(1)$ as $h \to 0$.

Can we precondition the sampling routine?

Ignition (Remix)

Rather than looking for a transformation that preserves the solution to the linear system, we look for one that gives the same *distribution*.

- Connection with re-parameterisations of models
- In particular with centred and non-centred parameterisations
- (Can help improve mixing between the latent field and the hyper-parameters)

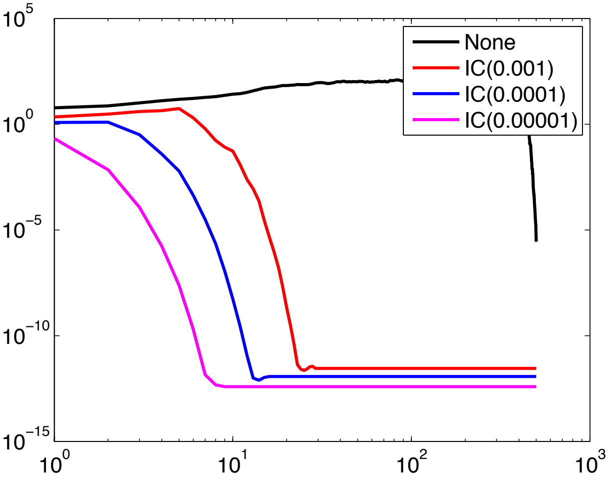
We both go down together

Preconditioned Sampling

Let $\boldsymbol{Q}$ and $\boldsymbol{M} = \boldsymbol{F}\boldsymbol{F}^T$ be symmetric positive definite matrices. If $\boldsymbol{y} \sim N\left(\mathbf{0}, \left(\boldsymbol{F}^{-1}\boldsymbol{Q}\boldsymbol{F}^{-T}\right)^{-1}\right)$, then the solution to $\boldsymbol{F}^T\boldsymbol{x} = \boldsymbol{y}$ is a zero-mean Gaussian random vector with precision matrix $\boldsymbol{Q}$.

- This replaces the problem of sampling from $N(\mathbf{0}, \boldsymbol{Q}^{-1})$ with sampling from $\sim N\left(\mathbf{0}, \left(\boldsymbol{F}^{-1}\boldsymbol{Q}\boldsymbol{F}^{-T}\right)^{-1}\right)$, which should be better behaved.
- Generic choice of $\boldsymbol{F}$: incomplete Cholesky factorisation of $\boldsymbol{Q}$.
- *Key point:* $\boldsymbol{F}^{-1}\boldsymbol{Q}\boldsymbol{F}^{-T}$ is almost certainly dense, but the product is cheap!

Speed Lab



10^{5}
10^{0}
10^{-5}
10^{-10}
10^{-15}
10^{0}
10^{1}
10^{2}
10^{3}
None
IC(0.001)
IC(0.0001)
IC(0.00001)

Breaking Glass

For stationary Gaussian random fields on a regular lattice (on a torus), the precision matrix (and the covariance matrix) is *circulant* and all of the calculations can be done in $\mathcal{O}(n \log n)$ operations using FFTs.

— Any operation involving *data* destroys the circulant structure, leading to precision matrices of the form $\boldsymbol{Q}_{post} = \boldsymbol{Q}_{prior} + \boldsymbol{\Lambda}$.

— This means that good MCMC methods that take into account the second order properties of the likelihood *cannot* be used!

All the best

Equivalent operator preconditioning

For Log-Gaussian Cox processes, if we precondition with the circulant prior precision, $\boldsymbol{\Lambda}$ is the Fisher information matrix, and we only observe a finite number of point patterns, then

$$\|\boldsymbol{x} - \boldsymbol{x}_m\| \leq C \left(\frac{\left(\int_W \exp(x(s))\, ds\right)}{m} \right)^{-m}.$$

- *Mesh independent superlinear convergence!*
  (NB: superlinear $\equiv$ super-geometric...)
- Therefore sampling from circulant + (nice) sparse matrices can be done in $\mathcal{O}(n \log n)$ operations!
- ("Nice" means $\operatorname{tr}(Q)^{-1}\boldsymbol{\Lambda}$) can be uniformly bounded)

Simon Smith and his amazing dancing bear

$\boldsymbol{Q}$ is generated using the exponential covariance function on a torus and the diagonals of Λ are $U[0, 10]$.

m ($m \times m$ grid)	16	32	64	128	256	512	1024	2048	4096
Preconditioned	6	6	6	6	6	6	6	6	6
Unpreconditioned	102	286	790	2166	-	-	-	-	-

Outline

Less talk, more rock

A classic latent Gaussian process in which the latent field is almost always modelled as block circulant (or block Toeplitz) is the log-Gaussian Cox process model for point pattern data.

$$\begin{aligned} y_i|\boldsymbol{\eta} &\sim Po(h^2 e^{\eta_i}) \\ \boldsymbol{\eta}|\boldsymbol{\theta} &\sim N(\mathbf{0}, \boldsymbol{Q}(\boldsymbol{\theta})^{-1}) \\ \boldsymbol{\theta} &\sim \pi(\boldsymbol{\theta}). \end{aligned}$$

— Here $\boldsymbol{Q}$ is a circulant matrix that has possibly been extended to include fixed effects.

— The preconditioner for in the case of fixed effects is the same! (block diagonal with the circulant preconditioner and maybe a scaling for the fixed effects components).

The problem with iterative methods

Iterative methods (LSQR for least squares sampling, and the matrix function methods) have one major drawback:

THEY DON'T COMPUTE THE LOG-DETERMINANT!

DETERMINANTS ARE VERY DIFFICULT TO COMPUTE!

Idea 1: Approximate factorisations

Concept: Even if we don't want to use the approximate factorisation to compute the sample, it will give a decent approximation to the determinant.

Problem: We have no control over the error. Furthermore, there is no way of checking how good your answer is.

Idea 2: Matrix functions (Bai et al ’96)

If the Cholesky decomposition is unavailable, a better way is to use the identity

$$\log(\det(A)) = \operatorname{tr}(\log(A)) = \sum_{i=1}^{n} e_i^T \log(A) e_i.$$

Is there a cheap way to approximate $\operatorname{tr}(\log(A))$*?*

A Stochastic Estimator of the Trace

Theorem (Hutchinson '90)

Let $B \in \mathbb{R}^{n\times n}$ be a symmetric matrix with non-zero trace. Let Z be the discrete random variable which takes the values $-1, 1$ each with probability $1/2$ and let z be a vector of n independent samples from Z. Then $z^T Bz$ is an unbiased estimator of $\mathrm{tr}(B)$ and Z is the unique random variable amongst zero mean random variables for which $z^T Bz$ is a minimum variance, unbiased estimator of $\mathrm{tr}(B)$.

Therefore

$$\log(\det(A)) = \mathbb{E}\left(z^T \log(A)z\right).$$

This can be estimated using a Monte Carlo method.

Nobody does it better?

The advantage of the MC scheme is that it is *unbiased* and, should you so desire, you can account for the extra randomness in an MCMC scheme to keep it asymptotically exact.

But it is slow!

As with all other things, it turns out that if you chose “better” than random vectors, you can get a method that is practically much better.

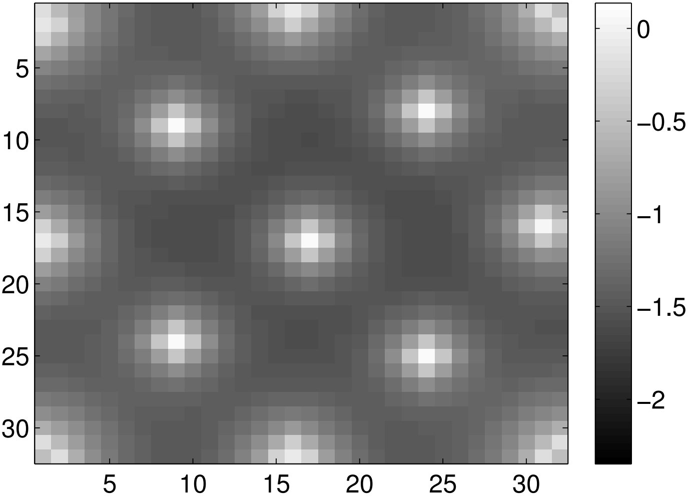
Putting it together

Here is the procedure that works best:

1. Pick a value p and produce a graph colouring of $\boldsymbol{Q}^p$.
2. For each colour c , construct a vector $\boldsymbol{z}_c$ that is randomly ± 1 (w.p. $1/2$) at the vertices of that colour and zero everywhere else
3. Use these vectors in Hutchinson's estimator of log(det($\boldsymbol{Q}$))

Sometimes it's worth doing a change of basis (wavelet transform).

A probing vector



I have no idea why this works!

What I know

- The elements of $\log(Q)$ decay exponentially away from the non-zero entries of $\boldsymbol{Q}$
- For each colour c,

$$\boldsymbol{z}_c^T \mathbf{log}(Q)\boldsymbol{z}_c = \sum_{i \in c} [\log(\boldsymbol{Q}]_{ii} + 2 \sum_{i,j \in c} (\pm 1) \log(\boldsymbol{Q})_{ij}$$

  and the first term will (maybe) dominate asymptotically.
- The “accidental” off diagonals cancel (?) and there are fewer of them than in the basic sampler and they are smaller (???)
- High p means more colours, but fewer vertices with each colour. If $p = n$ then you recover the trace formula.

Precondtioning?

If $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$ is a preconditioner, then

$$\log(\det(\boldsymbol{Q})) = 2\log(\det(\boldsymbol{L})) + \log(\det(\boldsymbol{L}^{-1}\boldsymbol{Q}\boldsymbol{L}^{-T})).$$

Typically, the first term is easy to compute, while the second is much better conditioned!

Outline

I could never take the place of your man

Hopefully, I have convinced you that there are a suite of iterative methods that can be used as efficient replacements for traditional methods.

That being said, these are still methods for *HARD* problems—if existing methods are satisfactory, there is no reason to change!

Au Suivant

Advertisement

Next September, the Third conference in Latent Gaussian Models and applications will be held in Reykjavik!

Come!

Working 9 to 5

We have a PhD position at NTNU working on the methods I have been talking about.

- Working with Elena Celledoni (numerics), Håvard Rue (statistics) and me.
- Topic: applying fast iterative methods to approximate inference for latent Gaussian models
- We need: Numerics person, parallel programming, C/C++, linear algebra/ Krylov methods.