ISSN 1172-496X (Print)
ISSN 1172-4234 (Online)

TECHNICAL REPORTS FROM THE ELECTRONICS GROUP
AT THE UNIVERSITY OF OTAGO

A Conjugate Direction Sampler for Normal Distributions, with a Few Computed Examples

by

Colin Fox

ELECTRONICS TECHNICAL
REPORT No. 2008-1

UNIVERSITY OF OTAGO
DUNEDIN, NEW ZEALAND

Online version has URL: http://www.physics.otago.ac.nz/reports/electronics/ETR2008-1.pdf
The author has homepage: http://www.physics.otago.ac.nz/people/fox
E-mail: fox@physics.otago.ac.nz
Address: Physics Department, University of Otago, P.O. Box 56, Dunedin, New Zealand

Electronics Group at Otago

In 1987 Millman and Grabel discarded the historical definition of 'electronics' as the science and technology of the motion of charges, preferring instead the operational definition that the primary concern of people doing electronics is *information processing*. This makes a distinction from *energy processing* practiced in the rest of electrical engineering. The act of information processing is what gets electronics practicioners invloved in the fours 'C's: communication, computation, control, and components. This practical definition seems to describe well the activities within the Electronics Group in the Physics Department at the University of Otago, and the range of topics covered in this technical report series.

In September 2008, research within the Electronics Group include projects on lightweight GPS tags for birds, modelling and control of a robotic elbow, design and deployment of an under-sea glider, analysis of networks of random resistors, electrical impedance imaging, calibration of numerical models for geothermal fields using Bayesian inference, modelling and sampling of Gaussian processes, and efficient algorithms for Markov chain Monte Carlo applied to inverse problems.

A Conjugate Direction Sampler for Normal Distributions, with a Few Computed Examples

Colin Fox

Abstract

Gaussian models and processes are common in statistics, particularly spatial statistics, being convenient from both computational and theoretical viewpoints. Efficient algorithms for sampling from Gaussian Markov random fields exploit sparseness of the precision matrix within a Cholesky factorization, or use circulant structure to allow diagonalization by Fourier transforms.

I present an alternative, sequential, algorithm derived from the conjugate gradient (CG) optimization algorithm. CG has the remarkable property of minimizing a quadratic form exactly in a finite number of steps while requiring storage of only two state vectors. The conjugate direction (CD) sampler has the analogous property generating independent (or exact) samples after a fixed number of steps (equal to the dimension of the state space) while requiring storage of only two state vectors, and a third auxiliary vector. Within the sampler one needs to operate by the precision matrix but there is no need to store the matrix or factorize it. Hence the conjugate direction sampler is useful for drawing samples in high dimensional problems where forming the precision matrix is impractical or inconvenient.

When implemented using finite precision arithmetic the CD sampling algorithm terminates incorrectly due to loss of conjugacy arrising from ill-conditioning of the precision matrix, or when eigenvalues are repeated. To some degree these issues may be treated by use of a preconditioning matrix. In a computed example the maximum sample dimension for which the CD sampling algorithm correctly terminates is 10^5.

Contents

Chapter 1

Introduction

Normal (or Gaussian) distributions are common throughout statistical modelling, being convenient from both computational and theoretical viewpoints. The seminal works by Hammersley and Clifford (1971), Clifford (1993), Besag (1974), and Geman and Geman (1984) have lead to the special case of Gaussian Markov random fields (GMRF) becoming a basic component of models in spatial statistics, including inverse problems (see e.g. Kaipio and Somersalo, 2005). Not uncommonly the GMRF is defined on a state space with dimension 10^6 to 10^9, or more, in which case general sampling algorithms can be very slow. A typical example of a high-dimensional GMRF occurs in exploratory analyses in inverse problems, for example when the state corresponds to parameter values in a finite element discretization of a 3-dimensional region. Efficient GMRF samplers exploit sparseness of the precision matrix within a Cholesky factorization to allow efficient sampling from the full GMRF, as well as marginal and conditional distributions (Rue, 2001), or use circulant matrix structure that allows Fourier methods to be used (Gneiting et al., 2005). Predictive distributions resulting from Gaussian process regression, including machine learning, are a further source of high-dimensional Gaussian distributions.

This article introduces a new, sequential, 'conjugate direction' (CD) sampling algorithm derived from the conjugate gradient (CG) optimization algorithm. CG has the remarkable property of minimizing a quadratic form in a finite number of steps (when exact arithmetic is used) while requiring storage of only two vectors. The CD sampler has the analogous property of independence between output samples with more than a finite spacing (equal to the dimension of the state space) while requiring storage of only two state vectors, and a third auxiliary vector. Within the sampler one needs to operate by the precision matrix but there is no need to store the matrix or factorize it. Hence the CD sampler is useful in high dimensional problems where forming the precision matrix is impractical or inconvenient.

The CG algorithm was introduced by Hestenes and Stiefel (1952) as a means of solving the matrix equation $A\mathbf{x} = \mathbf{b}$, or equivalently, minimizing the associated quadratic form

$$\phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}} A\mathbf{x} - \mathbf{b}^{\mathrm{T}}\mathbf{x}$$

where A is a symmetric positive definite (SPD) matrix. CG uses a sequence of search directions equal to the gradient of ϕ projected onto the subspace conjugate to all previous directions. That choice implies that the projection step requires storage of just the previous search direction and not the full sequence of previous directions (cf. prop. 2.3.1), as would be required in general. Hence the CG algorithm requires storage of just two vectors, and the ability to operate by the matrix A. The former feature allows application to high-dimensional problems, while the latter gives computational efficiency in those problems where operation by A can be performed much more cheaply than by direct matrix multiplication.

The CD sampler works similarly, generating a sequence of mutually conjugate directions

along which the conditional distribution is sampled. Since mutually conjugate vectors define mutually independent conditional distributions of a multivariate Gaussian, a finite sequence of conditional samples generates a sample from the target multivariate Gaussian. Hence, the CD sampler may be viewed as a Gibbs sampler in which the sequence of sampling directions is chosen in an optimal way to produce an exact sample, rather than the usual sequence of coordinate directions that results in geometric convergence.

Connections between the CG algorithm and a problem in curve fitting were shown by Jennings (1977), giving insight into the way that the eigenvalue structure of the matrix A affects the convergence of the basic CG algorithm (see also Axelsson, 1994; Nocedal and Wright, 1999). In particular, the CG algorithm terminates in the number of steps equal to the number of distinct eigenvalues of A. Termination in as few steps as possible is desirable in optimization, but is not desirable when implementing a sampling algorithm that is required to continue exploring all of state space. Modifications to the basic CD sampler are given to circumvent this property.

Despite the similar background and purpose, the CD sampler is not a close relative of the conjugate gradient Markov chain Monte Carlo (CGMCMC) introduced by Liu (2001); Liu et al. (2000). CGMCMC is a variant of the multi-point adaptive direction sampler (ADS) (Gilks et al., 1994) that utilizes the deterministic CG optimization algorithm to find an 'anchor point' for 'snooker' type moves, i.e., CGMCMC uses CG within the proposal step of a Metropolis-Hastings algorithm. In contrast, the CD sampler is a stochastic adaptation of the CG algorithm, producing a chain that is neither reversible nor (first-order) Markov. The CD sampler adapts the finite termination property of CG for quadratic functions to give finite-time independence property for Gaussian distributions, and utilizes the structure of search directions and conjugacy so that storage of just two state vectors and an auxiliary vector is required. Indeed, a motivation for developing CD was the conviction that careful use of just two state vectors could give dramatically more efficient sampling algorithms than the random selection from multiple state vectors as in ADS. That lesson had been learned in the field of optimization of high-dimensional functions and it seemed reasonable that the same principle would hold for sampling high-dimensional probability distributions. The same motivation also lead to development of the t-walk version of the Metropolis-Hastings MCMC algorithm, where again just two state vectors are stored.

A possible meeting of the Metropolis-Hastings algorithm and the CD sampler is when Metropolis-Hastings dynamics is used to perform conditional sampling within the CD algorithm applied to non-Gaussian target distributions. Also, it seems likely that CD and the t-walk can be combined, perhaps with CD forming a proposal distribution for the t-walk.

1.1 Efficient sampling from a Gaussian

Consider drawing a sample $\mathbf{x} \in \mathbb{R}^n$ from the zero-mean multi-variate normal distribution $N(\mathbf{0}, A^{-1})$ having density function

$$\pi(\mathbf{x}) = \sqrt{\frac{\det(A)}{2\pi^n}} \exp\left\{-\frac{1}{2}\mathbf{x}^{\mathrm{T}} A\mathbf{x}\right\}.$$

The precision matrix $A \in R^{n\times n}$ is a symmetric positive definite (SPD) matrix, as is the covariance matrix A^{-1} (see e.g. Feller, 1966, Sec. III.6 Thm. 3).

Standard efficient samplers use the Cholesky factorization, or eigen structure, as follows:

- The classical algorithm calculates the Cholesky factorization of the covariance matrix A^{-1} to give lower triangular matrix R satisfying

$$RR^{\mathrm{T}} = A^{-1}.$$

  Then if $\mathbf{z} \sim N(\mathbf{0}, I_n)$, $\mathbf{x} = R\mathbf{z} \sim N\left(\mathbf{0}, A^{-1}\right)$ as desired. Here I_n is the $n \times n$ identity matrix. Note that components of $\mathbf{z}$ may be drawn from standard normals since $\mathbf{z}_i \overset{\text{i.i.d.}}{\sim} N(0,1)$.

- Rue (2001) suggested Cholesky factorization of the precision matrix A to give lower triangular matrix L satisfying

$$LL^{\mathrm{T}} = A.$$

  Then if $\mathbf{z} \sim N(\mathbf{0}, I_n)$, solving $\mathbf{z} = L^{\mathrm{T}}\mathbf{x}$ (by back-substitution) gives $\mathbf{x} \sim N\left(\mathbf{0}, A^{-1}\right)$. When π is a GMRF, the Markov property implies that A is a sparse matrix, for which computationally efficient algorithms are available.

- The eigenvectors of the covariance matrix A^{-1} (or precision matrix A) give coordinates that are uncorrelated and *independent*. If the normalized eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ satisfy $A^{-1}\mathbf{u}_i = \lambda_i \mathbf{u}_i$ and $z_i \overset{\text{i.i.d.}}{\sim} N(0,1)$, then $\mathbf{x} = \sum_{i=1}^{n} z_i \sqrt{\lambda_i}\mathbf{u}_i \sim N\left(\mathbf{0}, A^{-1}\right)$ as desired. When the eigen decomposition is cheap to compute, e.g. via the Fourier transform for circulant matrices, this expansion gives a feasible efficient method.

Remark 1.1.1 The validity of these algorithms follow from the classical result for linear transformation of a multi-variate normal (see e.g. Feller, 1966). If $\mathbf{z} \sim N(\mu, \Sigma)$ then $\mathbf{x} = T\mathbf{z} \sim N\left(T\mu, T\Sigma T^{\mathrm{T}}\right)$ for any matrix T of full column rank. For example, correctness of the algorithm suggested by Rue follows since $A^{-1} = \left(L^{\mathrm{T}}\right)^{-1} L^{-1}$, and solving $\mathbf{z} = L^{\mathrm{T}}\mathbf{x}$ is the same as setting $\mathbf{x} = T\mathbf{z}$ where $T = \left(L^{\mathrm{T}}\right)^{-1}$.

Chapter 2

Conjugate Direction Sampling Algorithm

Properties of the (linear) CG algorithm and variants are derived in many excellent sources (see e.g. Axelsson, 1994; Gill et al., 1981; Greenbaum, 1997; Nocedal and Wright, 1999). The following development of the CD sampling algorithm is adapted from the constructive development of the linear CG algorithm given by Tan (1986).

2.1 Mutually conjugate vectors (and factorizations)

Definition 2.1.1 A set of k non-zero vectors $\mathbf{p}^{(i)}$, $i = 0, 1, \ldots, k-1$, is *mutually conjugate* w.r.t.[1] the SPD matrix $A \in \mathbb{R}^{n \times n}$ (or *mutually A-conjugate*[2]) if

$$\mathbf{p}^{(i)\mathrm{T}} A \mathbf{p}^{(j)} = 0 \quad \forall i \neq j.$$

Remark 2.1.1 Denote $\mathbf{p}^{(i)\mathrm{T}} A \mathbf{p}^{(i)} = d_i$. Note that $d_i > 0 \ \forall i$, as A is strictly positive definite.

Remark 2.1.2 Let $P_k \in R^{n \times k}$ be the matrix $\left[\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}\right]$, i.e. with columns being the mutually conjugate vectors $\mathbf{p}^{(i)}$, $i = 0, 1, \ldots, k-1$, and $D_k = \mathrm{diag}\left(d_0, d_1, \ldots, d_{k-1}\right) \in R^{k \times k}$ with d_i as defined in Remark 2.1.1. Then

$$P_k^{\mathrm{T}} A P_k = D_k. \tag{2.1}$$

The condition that the right-hand side be diagonal in equation 2.1 gives an equivalent characterization of non-zero vectors being mutually conjugate w.r.t. A.

Lemma 2.1.1 The elements of any set of mutually conjugate vectors are linearly independent.

Proof. Suppose that $\sum_i \alpha_i \mathbf{p}^{(i)} = 0$. Then $0 = \left(\sum_i \alpha_i \mathbf{p}^{(i)}\right)^{\mathrm{T}} A \mathbf{p}^{(j)} = \alpha_j d_j \Longrightarrow \alpha_j = 0.$ ∎

Corollary 2.1.1 A set of n mutually conjugate vectors is complete.

Remark 2.1.3 Having a (complete) set on n mutually conjugate vectors allows sampling from π in a straightforward way. If $\mathbf{z} \sim N\left(\mathbf{0}, I_n\right)$, then $\mathbf{y} = \sqrt{D_n^{-1}}\mathbf{z} \sim N\left(\mathbf{0}, D_n^{-1}\right)$, and $\mathbf{x} = P_n \mathbf{y} \sim N\left(\mathbf{0}, A^{-1}\right)$. This follows since $P_n^{\mathrm{T}} A P_n = D_n$ so $A^{-1} = P_n D_n^{-1} P_n^{\mathrm{T}}$.

[1] w.r.t. is an abreviation for 'with respect to'.

[2] Axelsson (1994) uses the term *A-orthogonal* which has merit since conjugacy is the property of orthogonality w.r.t. the inner product $(x, y) = x^{\mathrm{T}} A y$. We use the term 'conjugate' since its usage is well established.

The expression for $\mathbf{x}$ can be written $\mathbf{x} = \sum_{i=0}^{n-1} \left(\mathbf{z}_i/\sqrt{d_i}\right) \mathbf{p}^{(i)}$. Since the $\mathbf{z}_i$ are i.i.d. $\sim N(0,1)$, this shows how a sample from π may be generated using a sequence of standard normal random numbers. It also shows that the multivariate Gaussian is transformed to independent random variables by using a mutually conjugate basis.

The following 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.

Example 2.1.1 (Cholesky factorization of A^{-1}) Since $RR^{\mathrm{T}} = A^{-1}$, $R^{\mathrm{T}}AR = I$, so the columns of R are mutually conjugate w.r.t. A.

Example 2.1.2 (Cholesky factorization of A) Since $LL^{\mathrm{T}} = A$, $\left(L^{\mathrm{T}}\right)^{-1} L^{-1} = A^{-1}$, so $\left(L^{\mathrm{T}}\right)^{-1} = R$ and so the columns of $\left(L^{\mathrm{T}}\right)^{-1}$ are mutually conjugate w.r.t. A.

Example 2.1.3 (Eigen decomposition of A or A^{-1}) Eigen decomposition of A gives

$$A = UDU^{\mathrm{T}}$$

where U is unitary (columns are normalized eigenvectors) and D is diagonal (eigenvalues on diagonal). Then $U^{\mathrm{T}}AU = D$, so the eigenvectors are mutually conjugate w.r.t. A.

We now consider the sequential generation of samples from π in more detail.

Proposition 2.1.1 Let $\mathbf{p}^{(i)}$, $i = 0, 1, \ldots, k-1$, be a set of k mutually conjugate vectors w.r.t. A. A sample from the conditional distribution

$$\pi\left(\mathbf{x} | \mathbf{x} \in \mathrm{span}\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}\right\}\right)$$

may be generated by a sequence of samples from the one-dimensional conditional distributions in directions $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}$.

Proof. First consider a sample from the whole space, i.e. $\mathrm{span}\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(n-1)}\right\}$ and write

$$x = \sum_{i=0}^{n-1} \alpha_i \mathbf{p}^{(i)}.$$

The result in Remark 1.1.1 shows that $\mathbf{x} \sim N\left(\mathbf{0}, A^{-1}\right)$ when $\alpha \sim N\left(\mathbf{0}, D_n^{-1}\right)$ hence $\alpha_i \sim N\left(0, 1/d_i\right)$ are independent normal. Now consider the conditional distribution along line $\mathbf{p}^{(k-1)}$ when $\alpha_0, \alpha_1, \ldots, \alpha_{k-2}$ are fixed, i.e. for

$$\pi\left(\mathbf{x} | \mathbf{x} \in \alpha_0 \mathbf{p}^{(0)} + \alpha_1 \mathbf{p}^{(1)} + \cdots \alpha_{k-2} \mathbf{p}^{(k-2)} + \mathrm{span}\left\{\mathbf{p}^{(k-1)}\right\}\right).$$

Write $\mathbf{x} = \alpha_0 \mathbf{p}^{(0)} + \alpha_1 \mathbf{p}^{(1)} + \cdots \alpha_{k-2} \mathbf{p}^{(k-2)} + \lambda \mathbf{p}^{(k-1)}$. The density over λ is

$$\begin{aligned}
\pi(\lambda) &\propto \exp\left\{-\frac{1}{2}\left(\sum_{i=0}^{k-2} \alpha_i \mathbf{p}^{(i)} + \lambda \mathbf{p}^{(k-1)}\right)^{\mathrm{T}} A \left(\sum_{i=0}^{k-2} \alpha_i \mathbf{p}^{(i)} + \lambda \mathbf{p}^{(k-1)}\right)\right\} \\
&= \exp\left\{-\frac{1}{2}\left(\sum_{i=0}^{k-2} \alpha_i^2 d_i + \lambda^2 d_{k-1}\right)\right\},
\end{aligned}$$

i.e. $\lambda \sim N\left(0, 1/d_{k-1}\right)$ which is the same as the distribution of α_{k-1} in a sample from $\pi(\mathbf{x})$. ∎

Thus, a sample from $\pi(\mathbf{x})$ may be generated by a sequence of exactly n independent univariable samples taken from the conditional distributions in directions $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(n-1)}$ with components in previous directions fixed.

2.2 The vectors $\mathbf{q}^{(i)} = A\mathbf{p}^{(i)}$ and A-projections

Lemma 2.2.1 If $\mathbf{p}^{(i)}$, $i = 0, 1, \ldots, k-1$, is a set of mutually A-conjugate vectors, then $\mathbf{q}^{(i)} = A\mathbf{p}^{(i)}$, $i = 0, 1, \ldots, k-1$ is a set of vectors mutually conjugate w.r.t. A^{-1}.

Proof. Let $P_k = \left[\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}\right]$ and $Q_k = \left[\mathbf{q}^{(0)}, \mathbf{q}^{(1)}, \ldots, \mathbf{q}^{(k-1)}\right] = AP_k$. Then $Q_k^{\mathrm{T}} A^{-1} Q_k = P_k^{\mathrm{T}} A^{\mathrm{T}} A^{-1} A P_k = D_k$ and the result follows from Remark 2.1.2. ∎

Corollary 2.2.1 For P_k and Q_k as defined in the proof to Lemma 2.2.1

$$P_k^{\mathrm{T}} Q_k = Q_k^{\mathrm{T}} P_k = D_k.$$

Definition 2.2.1 If $\mathbf{p}^{(i)}$, $i = 0, 1, \ldots, k-1$, is a set of mutually A-conjugate vectors and P_k and Q_k are the matrices defined in the proof to Lemma 2.2.1, the *A-projection* onto $\operatorname{span}\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}\right\}$ is defined as

$$R_k = P_k D_k^{-1} P_k^{\mathrm{T}} A = P_k D_k^{-1} Q_k^{\mathrm{T}}.$$

The name A-projection is justified by the following two Lemmas and Remark.

Lemma 2.2.2 If $\mathbf{x} \in \operatorname{span}\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}\right\}$ then $R_k \mathbf{x} = \mathbf{x}$.

Proof. Write $\mathbf{x} = P_k \mathbf{a}$ for some $\mathbf{a} \in \mathbb{R}^{k \times 1}$. Then $R_k \mathbf{x} = P_k D_k^{-1} Q_k^{\mathrm{T}} P_k \mathbf{a} = P_k \mathbf{a} = \mathbf{x}$. ∎

Lemma 2.2.3 If $\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}, \mathbf{v}\right\}$ form a mutually conjugate set then $R_k \mathbf{v} = \mathbf{0}$.

Proof. Since $\mathbf{v}$ is A-conjugate to $\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}\right\}$, $P_k^{\mathrm{T}} A \mathbf{v} = 0$. The result follows from the definition of R_k. ∎

Remark 2.2.1 It is easy to check that $R_k^2 = R_k$ and that $R_k^{\mathrm{T}} A = A R_k$, i.e. is symmetric in the inner product $(\mathbf{x}, \mathbf{y}) = \mathbf{x}^{\mathrm{T}} A \mathbf{y}$. Hence R_k is a projection in the inner product $(\cdot, \cdot)$.

Lemma 2.2.4 Let $\mathbf{v} = (I_n - R_k)\mathbf{x}$ for any $\mathbf{x}$. Then $\mathbf{v}$ is either zero or A-conjugate to $\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}\right\}$.

Proof. $P_k^{\mathrm{T}} A R_k = P_k^{\mathrm{T}} A P_k D_k^{-1} P_k^{\mathrm{T}} A = P_k^{\mathrm{T}} A$ hence $P_k^{\mathrm{T}} A \mathbf{v} = P_k^{\mathrm{T}} A (I_n - R_k)\mathbf{x} = 0$. ∎

Lemma 2.2.5 Given a set of mutually conjugate vectors $\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}\right\}$, each vector $\mathbf{x}$ has the unique decomposition

$$\mathbf{x} = \alpha_0 \mathbf{p}^{(0)} + \alpha_1 \mathbf{p}^{(1)} + \cdots \alpha_{k-1} \mathbf{p}^{(k-1)} + \mathbf{v}$$

where $\alpha_0, \alpha_1, \ldots, \alpha_{k-1}$ are scalars and $\mathbf{v}$ is either zero or A-conjugate to $\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}\right\}$.

Proof. Since $R_k\mathbf{x} \in \text{span}\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}\right\}$ we can write $R_k\mathbf{x} = \alpha_0\mathbf{p}^{(0)} + \alpha_1\mathbf{p}^{(1)} + \cdots \alpha_{k-1}\mathbf{p}^{(k-1)}$. Define $\mathbf{v} = (I_n - R_k)\,\mathbf{x}$ which, by Lemma 2.2.4, is either zero or conjugate to $\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}\right\}$. This gives a factorization of the desired type. Uniqueness of the factorization follows from Lemma 2.1.1. ∎

Remark 2.2.2 We will use the notation in Lemma 2.2.5 for factorizations in the state variable, $\mathbf{x}$. It will be convenient to factorize the auxiliary vectors using the mutually A^{-1}-conjugate set of vectors $\left\{\mathbf{q}^{(0)}, \mathbf{q}^{(1)}, \ldots, \mathbf{q}^{(k-1)}\right\}$ defined in Lemma 2.2.1, and write that factorization as

$$\mathbf{b} = \alpha_0\mathbf{q}^{(0)} + \alpha_1\mathbf{q}^{(1)} + \cdots \alpha_{k-1}\mathbf{q}^{(k-1)} + \mathbf{w}.$$

Lemma 2.2.6 $\mathbf{v}^{\mathrm{T}}\mathbf{q}^{(i)} = 0$, $i = 0, 1, \ldots, k-1$, and $\mathbf{w}^{\mathrm{T}}\mathbf{p}^{(i)} = 0$, $i = 0, 1, \ldots, k-1$, where $\mathbf{v}$ and $\mathbf{w}$ are defined in the decompositions in Lemma 2.2.5 and Remark 2.2.2, respectively.

Proof. Since $\mathbf{w}$ is A^{-1}-conjugate to $\left\{\mathbf{q}^{(0)}, \mathbf{q}^{(1)}, \ldots, \mathbf{q}^{(k-1)}\right\}$, $0 = Q_k^{\mathrm{T}}A^{-1}\mathbf{w} = P_k^{\mathrm{T}}AA^{-1}\mathbf{w} = P_k^{\mathrm{T}}\mathbf{w}$ establishes the result for $\mathbf{w}$. The result for $\mathbf{v}$ holds similarly. ∎

2.3 Auxiliary vector and quadratic form

Definition 2.3.1 Given a partial sample with decomposition $\mathbf{x}^{(k)} = \alpha_0\mathbf{p}^{(0)} + \alpha_1\mathbf{p}^{(1)} + \cdots \alpha_{k-1}\mathbf{p}^{(k-1)} + \mathbf{v}$, an *auxiliary vector* is *any* vector with decomposition $\mathbf{b}^{(k)} = \alpha_0\mathbf{q}^{(0)} + \alpha_1\mathbf{q}^{(1)} + \cdots \alpha_{k-1}\mathbf{q}^{(k-1)} + \mathbf{w}$, i.e. satisfying

$$S_k\mathbf{b}^{(k)} = AR_k\mathbf{x}^{(k)},$$

where S_k is the A^{-1}-projection onto $\text{span}\left\{\mathbf{q}^{(0)}, \mathbf{q}^{(1)}, \ldots, \mathbf{q}^{(k-1)}\right\}$ and R_k is the A-projection in Definition 2.2.1.

Remark 2.3.1 $S_k = R_k^{\mathrm{T}}$. This follows directly from the observation that $S_k = Q_kD_k^{-1}Q_k^{\mathrm{T}}A = AP_kD_k^{-1}P_k^{\mathrm{T}}$ and the symmetry of both D_k and A.

Proposition 2.3.1 Given $\mathbf{x}^{(k)}$ and auxiliary vector $\mathbf{b}^{(k)}$ in Definition 2.3.1, $\mathbf{x}^{(k)}$ minimizes the quadratic form

$$\phi_k(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}A\mathbf{x} - \mathbf{b}^{(k)\mathrm{T}}\mathbf{x}$$

in the affine subspace $\mathbf{v} + \text{span}\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}\right\}$.

Proof. The minimum occurs when for $j = 0, 1, \ldots, k-1$

$$\begin{aligned}
0 &= \frac{\partial\phi_k\left(\mathbf{v} + \sum_{i=0}^{k-1} a_i\mathbf{p}^{(i)}\right)}{\partial a_j} \\
&= \frac{\partial}{\partial a_j}\left(\frac{1}{2}\left(\mathbf{v} + \sum_{i=0}^{k-1} a_i\mathbf{p}^{(i)}\right)^{\mathrm{T}} A\left(\mathbf{v} + \sum_{i=0}^{k-1} a_i\mathbf{p}^{(i)}\right) - \left(\mathbf{w} + \sum_{i=0}^{k-1}\alpha_i\mathbf{q}^{(i)}\right)^{\mathrm{T}}\left(\mathbf{v} + \sum_{i=0}^{k-1} a_i\mathbf{p}^{(i)}\right)\right) \\
&= \frac{\partial}{\partial a_j}\left(\frac{1}{2}\sum_{i=0}^{k-1} a_i^2 d_i - \sum_{i=0}^{k-1}\alpha_i a_i d_i - \mathbf{w}^{\mathrm{T}}\mathbf{v}\right) \\
&= a_j d_j - \alpha_j d_j.
\end{aligned}$$

Hence the minimum occurs for $a_j = \alpha_j$ as desired. ∎

In the case when $\mathbf{x}^{(0)}$ and $\mathbf{b}^{(0)}$ are zero, sequential calculation of vectors $\mathbf{x}^{(k)}$ and $\mathbf{b}^{(k)}$ as in Definition 2.3.1 is straightforward. However, setting $\mathbf{x}^{(0)} = \mathbf{b}^{(0)} = \mathbf{0}$ is not a feasible initialization of the CD sampler. Hence we must consider more general initial values. The following Lemma shows how vectors $\mathbf{x}^{(k)}$ and $\mathbf{b}^{(k)}$ may be sequentially generated from arbitrary initial values.

Lemma 2.3.1 Given the set of k mutually conjugate vectors $\mathbf{p}^{(i)}$, $i = 0, 1, \ldots, n-1$, constants α_i, $i = 0, 1, \ldots, n-1$, and initial vectors $\mathbf{x}^{(0)}$ and $\mathbf{b}^{(0)}$, a sequence of vectors with the decompositions in Lemma 2.2.5 and Remark 2.2.2 may be generated sequentially by

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \left(\alpha_{k-1} - \frac{\mathbf{p}^{(k-1)\mathrm{T}} A \mathbf{x}^{(k-1)}}{d_{k-1}} \right) \mathbf{p}^{(k-1)} \tag{2.2}$$

and

$$\mathbf{b}^{(k)} = \mathbf{b}^{(k-1)} + \left(\alpha_{k-1} - \frac{\mathbf{p}^{(k-1)\mathrm{T}} \mathbf{b}^{(k-1)}}{d_{k-1}} \right) A \mathbf{p}^{(k-1)} \tag{2.3}$$

for $k = 1, 2, \ldots, n$.

Proof. This result may be proved by induction on k. Suppose $\mathbf{x}^{(k-1)}$ and $\mathbf{b}^{(k-1)}$ are written according to the decompositions, i.e.,

$$\mathbf{x}^{(k-1)} = \sum_{i=0}^{k-2} \alpha_i \mathbf{p}^{(i)} + \mathbf{v}^{(k-1)}$$

and

$$\mathbf{b}^{(k-1)} = \sum_{i=0}^{k-2} \alpha_i \mathbf{q}^{(i)} + \mathbf{w}^{(k-1)}$$

where $P_{k-1}^{\mathrm{T}} A \mathbf{v}^{(k-1)} = 0$ and $Q_{k-1}^{\mathrm{T}} A^{-1} \mathbf{w}^{(k-1)} = 0$.

We may write $\mathbf{x}^{(k)}$ given by Equation 2.2 in this form as $\mathbf{x}^{(k)} = \sum_{i=0}^{k-1} \alpha_i \mathbf{p}^{(i)} + \mathbf{v}^{(k)}$, where

$$\mathbf{v}^{(k)} = \mathbf{v}^{(k-1)} - \frac{\mathbf{p}^{(k-1)\mathrm{T}} A \mathbf{x}^{(k-1)}}{d_{k-1}} \mathbf{p}^{(k-1)}.$$

Hence $P_{k-1}^{\mathrm{T}} A \mathbf{v}^{(k)} = 0$ since $P_{k-1}^{\mathrm{T}} A \mathbf{v}^{(k-1)} = 0$ and $P_{k-1}^{\mathrm{T}} A \mathbf{p}^{(k-1)} = 0$. Further, $\mathbf{p}^{(k-1)\mathrm{T}} A \mathbf{v}^{(k)} = \mathbf{p}^{(k-1)\mathrm{T}} A \left(\mathbf{v}^{(k-1)} - \frac{\mathbf{p}^{(k-1)\mathrm{T}} A \mathbf{x}^{(k-1)}}{d_{k-1}} \mathbf{p}^{(k-1)} \right) = 0$ by construction. Composing these two results gives $P_k^{\mathrm{T}} A \mathbf{v}^{(k)} = 0$, and hence $\mathbf{x}^{(k)}$ is written as the factorization in Lemma 2.2.5 as desired. The result for $\mathbf{b}^{(k)}$ holds similarly. ∎

Remark 2.3.2 Since the $\mathbf{x}^{(k)}$ and $\mathbf{b}^{(k)}$ generated in Lemma 2.3.1 share a common set of α_i in their respective factorizations, $\mathbf{b}^{(k)}$ is an auxiliary vector for $\mathbf{x}^{(k)}$ as in Definition 2.3.1.

The sequential updating of vectors in Lemma 2.3.1 allows sequential sampling from the conditional distributions of the target Gaussian density along a sequence of mutually conjugate directions. In particular, the conditional density

$$\pi \left(\mathbf{x} | \mathbf{x} \in \mathbf{x}^{(k-1)} + \mathrm{span} \left\{ \mathbf{p}^{(k-1)} \right\} \right)$$

is a one dimensional Gaussian with mean $-\frac{\mathbf{p}^{(k-1)\mathrm{T}} A\mathbf{x}^{(k-1)}}{d_{k-1}}$ and variance $1/d_{k-1}$. Hence if $z \sim N(0,1)$ and $\alpha_{k-1} = z/\sqrt{d_{k-1}}$ then $\mathbf{x}^{(k)}$ is distributed as this conditional density. A sequence of conditional samples will give a sample from

$$\pi\left(\mathbf{x}|\mathbf{x} \in \mathrm{span}\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}\right\}\right)$$

as in Proposition 2.1.1.

2.4 Generating mutually conjugate vectors

Given any set of linearly independent vectors $\left\{\mathbf{r}^{(k)}\right\}_{k=0,1,\ldots,n-1}$, we can form a mutually conjugate set $\left\{\mathbf{p}^{(k)}\right\}_{k=0,1,\ldots,n-1}$ (spanning the same space) using the following modified Gram-Schmidt process. First set $\mathbf{p}^{(0)} = \mathbf{r}^{(0)}$ and then define

$$\mathbf{p}^{(k)} = \mathbf{r}^{(k)} - \sum_{i=0}^{k-1} \frac{\mathbf{r}^{(k)\mathrm{T}} A\mathbf{p}^{(i)}}{d_i}\mathbf{p}^{(i)} \quad \text{for } k = 1, 2, \ldots, n-1. \tag{2.4}$$

It is easy to check that the set $\left\{\mathbf{p}^{(k)}\right\}_{k=0,1,\ldots,n-1}$ has the right properties. This process for generating mutually conjugate vectors requires storage of all previous vectors to evaluate the summation. However, if the set $\left\{\mathbf{r}^{(k)}\right\}_{k=0,1,\ldots,n-1}$ is chosen carefully, most of the terms in the summation evaluate to zero, making it necessary to store fewer vectors. Moreover, the process may be performed sequentially with storage of just two state vectors at each stage, hence giving a sampling algorithm that produces a second-order Markov chain.

An appropriate choice is the negative gradient of the associated quadratic form at the current point $\mathbf{x}^{(k)}$, i.e.

$$\mathbf{r}^{(k)} = -\nabla\phi_k\left(\mathbf{x}^{(k)}\right) = \mathbf{b}^{(k)} - A\mathbf{x}^{(k)}. \tag{2.5}$$

The term on the right-hand side, and hence $\mathbf{r}^{(k)}$, is the *residual* of the linear equation at the point $\mathbf{x}^{(k)}$.

Lemma 2.4.1 The residual vector $\mathbf{r}^{(k)}$ is orthogonal to the space $\mathrm{span}\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}\right\}$.

Proof. By Proposition 2.3.1, $\mathbf{x}^{(k)}$ is the point in the space $\mathbf{v} + \mathrm{span}\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}\right\}$ at which ϕ_k is minimized. Hence the projection of the gradient $\mathbf{r}^{(k)} = -\nabla\phi_k\left(\mathbf{x}^{(k)}\right)$ onto $\mathrm{span}\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}\right\}$ is zero, i.e. $\mathbf{r}^{(k)}$ is orthogonal to the space. ∎

Lemma 2.4.2 $\mathbf{r}^{(i)\mathrm{T}}\mathbf{r}^{(j)} = 0$ for all $i \neq j$.

Proof. By construction, $\mathrm{span}\left\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(k-1)}\right\}$ is the same space as $\mathrm{span}\left\{\mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(k-1)}\right\}$, hence from Lemma 2.4.1, $\mathbf{r}^{(k)}$ is orthogonal to $\mathbf{r}^{(j)}$ for all $j < k$. The result follows from symmetry. ∎

Proposition 2.4.1 When $\mathbf{r}^{(k)}$ is calculated as Equation 2.5, and $\mathbf{x}^{(k)}$ and $\mathbf{b}^{(k)}$ are updated as in Lemma 2.3.1 then all the terms in the summation in Equation 2.4 are zero, except for the term $i = k - 1$.

Proof.

$$\begin{aligned}
\mathbf{r}^{(i+1)} &= \mathbf{b}^{(i+1)} - A\mathbf{x}^{(i+1)} \\
&= \mathbf{b}^{(i)} + \left(\alpha_i - \frac{\mathbf{p}^{(i)\mathrm{T}}\mathbf{b}^{(i)}}{d_i}\right) A\mathbf{p}^{(i)} - A\left(\mathbf{x}^{(i)} + \left(\alpha_i - \frac{\mathbf{p}^{(i)\mathrm{T}}A\mathbf{x}^{(i)}}{d_i}\right)\mathbf{p}^{(i)}\right) \\
&= \mathbf{r}^{(i)} - \frac{\mathbf{p}^{(i)\mathrm{T}}\mathbf{b}^{(i)}}{d_i} A\mathbf{p}^{(i)} + \frac{\mathbf{p}^{(i)\mathrm{T}}A\mathbf{x}^{(i)}}{d_i} A\mathbf{p}^{(i)} \\
&= \mathbf{r}^{(i)} - \frac{\mathbf{p}^{(i)\mathrm{T}}\mathbf{r}^{(i)}}{d_i} A\mathbf{p}^{(i)}.
\end{aligned}$$

In particular $A\mathbf{p}^{(i)} = \left(\mathbf{r}^{(i+1)} - \mathbf{r}^{(i)}\right)\frac{-d_i}{\mathbf{p}^{(i)\mathrm{T}}\mathbf{r}^{(i)}}$, i.e. consecutive residuals differ by a scalar multiple of $A\mathbf{p}^{(i)}$. Hence the summation is

$$\sum_{i=0}^{k-1} \frac{\mathbf{r}^{(k)\mathrm{T}}A\mathbf{p}^{(i)}}{d_i}\mathbf{p}^{(i)} = -\sum_{i=0}^{k-1} \frac{\mathbf{r}^{(k)\mathrm{T}}\left(\mathbf{r}^{(i+1)} - \mathbf{r}^{(i)}\right)}{\mathbf{p}^{(i)\mathrm{T}}\mathbf{r}^{(i)}}\mathbf{p}^{(i)} = -\frac{\mathbf{r}^{(k)\mathrm{T}}\mathbf{r}^{(k)}}{\mathbf{p}^{(k-1)\mathrm{T}}\mathbf{r}^{(k-1)}}\mathbf{p}^{(k-1)}.$$

■

2.5 Sequential algorithm

Proposition 2.4.1 shows that just the current state and the previous sampling direction is required to sequentially calculate the set of mutually conjugate vectors. The resulting sequential sampling algorithm is given in Algorithm 1.

Since a scaling of the previous sampling direction $\mathbf{p}^{(k-1)}$ does not alter the algorithm, storing the current state and the previous sampling direction is equivalent to storing the current and the previous state (with the sampling direction being the difference of these states). Hence we could rewrite the algorithm to require storage of two states, explicitly showing that the algorithm produces a chain that is second-order Markov.

2.6 Distribution of the auxiliary vector

Lemma 2.6.1 $\mathbf{b}^{(n)} \sim N(\mathbf{0}, A)$, i.e. with A as covariance matrix rather than precision matrix.

Proof. Since $S_n = R_n = I_n$, from Definition 2.3.1 we see that $\mathbf{b}^{(n)} = A\mathbf{x}^{(n)}$. The result follows from the transformation of multi-variate normals quoted in Remark 1.1.1. ■

The interesting consequence is that a sample from each of $N(\mathbf{0}, A^{-1})$ and $N(\mathbf{0}, A)$ is generated by the CD sampler. Thus, which ever of the covariance or precision matrix is computationally cheaper to operate by, may be used in the algorithm. A sample from the desired distribution is then produced as $\mathbf{x}^{(n)}$ or $\mathbf{b}^{(n)}$. Note, however, that $A\mathbf{x}^{(n)} = \mathbf{b}^{(n)}$ so these samples are not independent.

$\underline{\mathbf{x} = \text{cdsample}(A)}$
initialize $\mathbf{x}^{(0)}$ and $\mathbf{b}^{(0)}$ (with $A\mathbf{x}^{(0)} \neq \mathbf{b}^{(0)}$)
$\mathbf{r}^{(0)} \leftarrow \mathbf{b}^{(0)} - A\mathbf{x}^{(0)}$
$\mathbf{p}^{(0)} \leftarrow \mathbf{r}^{(0)}$
for $k = 1$ to n do
  $d_{k-1} \leftarrow \mathbf{p}^{(k-1)\mathrm{T}} A\mathbf{p}^{(k-1)}$
  draw $z_{k-1} \sim N(0,1)$
  $\alpha_{k-1} \leftarrow z_{k-1}/\sqrt{d_{k-1}}$
  $\mathbf{x}^{(k)} \leftarrow \mathbf{x}^{(k-1)} + \left(\alpha_{k-1} - \dfrac{\mathbf{p}^{(k-1)\mathrm{T}} A\mathbf{x}^{(k-1)}}{d_{k-1}}\right)\mathbf{p}^{(k-1)}$
  $\mathbf{b}^{(k)} \leftarrow \mathbf{b}^{(k-1)} + \left(\alpha_{k-1} - \dfrac{\mathbf{p}^{(k-1)\mathrm{T}} \mathbf{b}^{(k-1)}}{d_{k-1}}\right) A\mathbf{p}^{(k-1)}$
  $\mathbf{r}^{(k)} \leftarrow \mathbf{b}^{(k)} - A\mathbf{x}^{(k)}$
  $\mathbf{p}^{(k)} \leftarrow \mathbf{r}^{(k)} + \dfrac{\mathbf{r}^{(k)\mathrm{T}}\mathbf{r}^{(k)}}{\mathbf{p}^{(k-1)\mathrm{T}}\mathbf{r}^{(k-1)}}\mathbf{p}^{(k-1)}$
$\mathbf{x} \leftarrow \mathbf{x}^{(n)}$

Algorithm 1: Preliminary implementation of the conjugate direction sampling algorithm using direct implementation of sequential calculation formulas. Here n is the dimension of the state vector $\mathbf{x}$. If the algorithm completes the n iterations, i.e. $d_{k-1} \neq 0, \forall k = 1, 2, \ldots, n$, the algorithm returns $\mathbf{x} \sim N\left(\mathbf{0}, A^{-1}\right)$.

2.7 Practical implementation

When the CG algorithm is applied to quadratic objective functions, it is possible to use just one matrix operation and two vector products per iteration (see, e.g. Tan (1986)). Similarly, the calculations in the preliminary CD algorithm may be restructured to reduce the computation required per iteration.

Explicitly evaluating the vector $\mathbf{q}^{(k-1)} = A\mathbf{p}^{(k-1)}$ and some required vector products allows simplification of the sequential evaluation of $\mathbf{b}^{(k)}$ and $\mathbf{x}^{(k)}$. The result in Proposition 2.4.1 for the difference between consecutive residuals leads to a simplified calculation of $\mathbf{r}^{(k)}$. The same result along with Lemma 2.4.2 also leads to a simplified computation for computing the new sampling direction $\mathbf{p}^{(k)}$.

The resulting efficient CD sampler algorithm is presented in Algorithm 2. Cost per iteration is reduced to one matrix operation and four vector products per iteration.

The algorithm requires that $\mathbf{x}$ and $\mathbf{b}$ are initialized so that $A\mathbf{x} \neq \mathbf{b}$. (If $A\mathbf{x} = \mathbf{b}$ then $\mathbf{p} = \mathbf{q} = \mathbf{0}$ and so $d = 0$ and the algorithm cannot complete even a single iteration.) A simple way to do this is to draw $\mathbf{b}^{(0)} \sim N(0, I_n)$ and initialize $\mathbf{x} = \mathbf{0}$ and $\mathbf{b} = \mathbf{b}^{(0)}$. Then, with probability 1, $A\mathbf{x} \neq \mathbf{b}$ and the algorithm is correctly initialized. If more than one sample from the target Gaussian is required, a simple strategy for re-initializing after n iterations is to set $\mathbf{b} = \mathbf{b}^{(0)}$, with no change to $\mathbf{x}$.

$\underline{\mathbf{x}, \mathbf{b} = \text{cdsample}(A)}$

initialize $\mathbf{x}$ and $\mathbf{b}$ (with $A\mathbf{x} \neq \mathbf{b}$)

$\mathbf{r} \leftarrow \mathbf{b} - A\mathbf{x}$

$\mathbf{p} \leftarrow \mathbf{r}$

for $k = 1$ to n do

$\mathbf{q} \leftarrow A\mathbf{p}$

$d \leftarrow \mathbf{q}^{\mathrm{T}}\mathbf{p}$

$e \leftarrow \mathbf{q}^{\mathrm{T}}\mathbf{x}/d$

$f \leftarrow \mathbf{p}^{\mathrm{T}}\mathbf{b}/d$

draw $z \sim N(0, 1)$

$\alpha \leftarrow z/\sqrt{d}$

$\mathbf{x} \leftarrow \mathbf{x} + (\alpha - e)\,\mathbf{p}$

$\mathbf{b} \leftarrow \mathbf{b} + (\alpha - f)\,\mathbf{q}$

$\mathbf{r} \leftarrow \mathbf{r} - (f - e)\mathbf{q}$

$\mathbf{p} \leftarrow \mathbf{r} - \dfrac{\mathbf{r}^{\mathrm{T}}\mathbf{q}}{d}\mathbf{p}$

Algorithm 2: Implementation of the basic conjugate direction sampler with reduced computational cost. If the algorithm terminates without error, i.e. $d \neq 0$ in each iteration, it returns $\mathbf{x} \sim N\left(\mathbf{0}, A^{-1}\right)$ and $\mathbf{b} \sim N(\mathbf{0}, A)$. Indexes have been left off to also give an algorithm requiring minimal storage.

Chapter 3

Repeated eigenvalues and preconditioning

As with the CG algorithm, the state vector $\mathbf{x}^{(k)}$ in the CD algorithm is given by $\mathbf{x}^{(0)}$ plus an element of the Krylov subspace $\text{span}\{\mathbf{r}^{(0)}, A\mathbf{r}^{(0)}, A^2\mathbf{r}^{(0)}, \ldots, A^k\mathbf{r}^{(0)}\}$ (see e.g. Axelsson, 1994). When the matrix A has repeated eigenvalues, i.e. A is degenerate, the Krylov subspaces have dimension less than n, implying that the algorithm as presented will not terminate correctly.

The dimension of the Krylov subspaces may be bounded as follows. Label the eigenvalues of A by $\lambda_1, \lambda_2, \ldots, \lambda_m$ where $m \leq n$. Let $\mathbf{r}_i^{(0)}$ be the projection of $\mathbf{r}^{(0)}$ onto the subspace of eigenvectors associated with λ_i. Then $A^k\mathbf{r}^{(0)} = \sum_{i=1}^m \lambda_i^k \mathbf{r}_i^{(0)}$. Since the RHS is a linear combination of m vectors, the Krylov subspace is made up of linear combinations of m vectors and hence has dimension at most m.

Since the sampling vectors $\mathbf{p}^{(k)}$ are elements of the Krylov subspaces, Lemma 2.1.1 implies that $p^{(i)} = \mathbf{0}, \forall i > m$. Hence if eigenvalues of A are repeated, so $m < n$, we have $p^{(k-1)} = \mathbf{0}$ for some $k \leq n$, hence $d_{k-1} = 0$ for some $k \leq n$. With exact arithmetic the algorithm will then attempt a divide by zero, generating an error. In the presence of numerical roundoff, d_{k-1} may be small, but not exactly zero, causing a large erroneous calculation when dividing by d_{k-1}.

A simple example of this problem occurs when sampling from the simplest multivariate Gaussian $N(0, I_n)$. Then $A = I_n$, and inspection of Algorithm 1 shows that $\mathbf{r}^{(0)} = \mathbf{b}^{(0)} - \mathbf{x}^{(0)}$, $d_0 = \|\mathbf{p}^{(0)}\|^2$, giving $\mathbf{r}^{(1)} = \mathbf{0}$. Then $\mathbf{p}^{(1)} = \mathbf{0}$ and the algorithm fails as outlined above. Hence, a feature of the algorithm as presented is that the Gaussian which is the simplest to sample by existing methods presents a difficulty to the CD sampler. On the other hand, Gaussian densities that arise in 'natural' settings are unlikely to have any particular eigenvalue structure, and may be sampled using the CD sampler without modification.

The CG algorithm is also known to be optimal, in a certain sense, for an equivalent polynomial fitting problem (Jennings, 1977; Nocedal and Wright, 1999). This connection shows that convergence of the CG algorithm depends not only on the number of distinct eigenvalues, but also on the distribution of eigenvalues. In particular, CG converges more rapidly when eigenvalues are clustered, and can give good approximate solutions in a reduced number of iterations, thereby getting close to the performance for repeated eigenvalues. Conversely, convergence is slowest when eigenvalues are uniformly distributed. While convergence in fewer than n steps is desirable for an optimization algorithm, it is undesirable in a sampling algorithm. Therefore, performance of the sampling algorithm is best for matrices A with uniformly spaced, distinct eigenvalues.

For CG algorithms, a standard method for accelerating convergence is to 'precondition' the

linear system, or quadratic form, by transforming to a system with more clustered eigenvalues. Similarly, performance of the CD sampler may be improved by transforming the precision matrix A as

$$\tilde{A} = U^T A U$$

so that the eigenvalues of $\tilde{A}$ are uniformly distributed, and sample instead from $N(0, \tilde{A}^{-1})$. Then if $z \sim N(0, \tilde{A}^{-1})$, $x = Uz \sim N(0, A^{-1})$ as desired. The only other requirement for U is that it is cheap to operate by. Similarly, we also call transformation by U *preconditioning*, despite its effect on eigenvalues being almost the opposite of the preconditioning matrices used in optimization.

3.1 A preconditioning matrix U

In the examples that follow in sections 4.1 and 4.2 a preconditioning matrix was used that is one on the diagonal, i.i.d. uniform random variables in the entries just above the diagonal, and is otherwise zero. Admittedly this was the first preconditioning matrix I tried and since it worked no others were tested.

Clearly there are many more possible preconditioning matrices. Indeed the eigenvalue structure of $U^T U$ for this choice of U is close to the eigenvalue structure of the discrete Laplacian, which has a $\sin^2$ functional form. Since this has flat sections for small and large eigenvalues, this preconditioner is only useful for problems with thousands of variables, or fewer. The development of 'best' preconditioning matrices therefore remains quite open.

Chapter 4

Numerical Examples

4.1 Random tridiagonal SPD

In this example the precision matrix is derived from a $n \times n$ matrix that has ones on the diagonal, uniform random variables on the super-diagonal, and is otherwise zero. The case for $n = 10$ looks like (with zeros marked as a dot)

$$U = \begin{pmatrix} 1 & U_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & U_2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & U_3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & U_4 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & U_5 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 & U_6 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & U_7 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & U_8 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & U_9 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \end{pmatrix}$$

where $U_i \overset{\text{i.i.d.}}{\sim} U(0,1)$. The precision matrix is calculated as $A = U^{\mathrm{T}}U$. In this example we use the following instance of this process

$$A = \begin{pmatrix} 1 & 0.9501 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.9501 & 1.9027 & 0.2311 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2311 & 1.0534 & 0.6068 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.6068 & 1.3683 & 0.4860 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4860 & 1.2362 & 0.8913 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8913 & 1.7944 & 0.7621 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.7621 & 1.5808 & 0.4565 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.4565 & 1.2084 & 0.0185 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0185 & 1.0003 & 0.8214 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8214 & 1.6747 \end{pmatrix}$$

The eigenvalues of A are plotted in Figure 4.1. Note that all eigenvalues are distinct, hence the naive CD sampler will not have problems of premature termination.

The exact covariance matrix is (rounded to 4 decimal places)

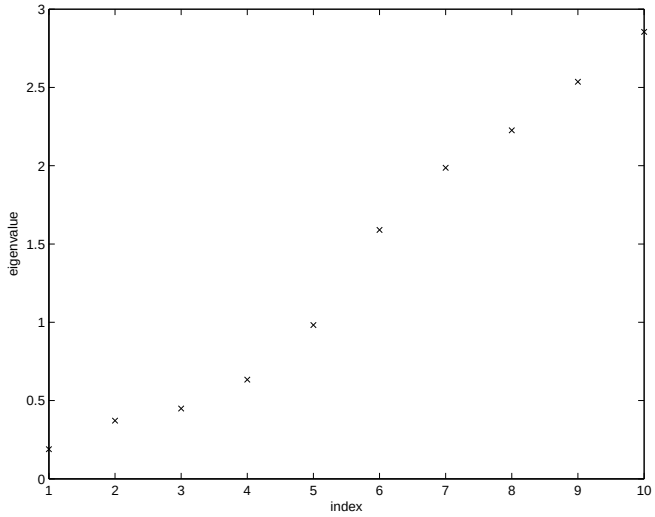


Figure 4.1. Eigenvalues of the precision matrix A plotted against index, in ascending order.

$A^{-1} =$

$$\begin{pmatrix}
1.9786 & -1.0300 & 0.3454 & -0.2073 & 0.1523 & -0.0982 & 0.0531 & -0.0201 & 0.0006 & -0.0003 \\
-1.0300 & 1.0840 & -0.3635 & 0.2181 & -0.1603 & 0.1034 & -0.0560 & 0.0211 & -0.0007 & 0.0003 \\
0.3454 & -0.3635 & 1.5728 & -0.9439 & 0.6936 & -0.4474 & 0.2421 & -0.0915 & 0.0028 & -0.0014 \\
-0.2073 & 0.2181 & -0.9439 & 1.5555 & -1.1430 & 0.7372 & -0.3989 & 0.1508 & -0.0047 & 0.0023 \\
0.1523 & -0.1603 & 0.6936 & -1.1430 & 2.3520 & -1.5169 & 0.8209 & -0.3102 & 0.0096 & -0.0047 \\
-0.0982 & 0.1034 & -0.4474 & 0.7372 & -1.5169 & 1.7019 & -0.9210 & 0.3481 & -0.0108 & 0.0053 \\
0.0531 & -0.0560 & 0.2421 & -0.3989 & 0.8209 & -0.9210 & 1.2084 & -0.4567 & 0.0141 & -0.0069 \\
-0.0201 & 0.0211 & -0.0915 & 0.1508 & -0.3102 & 0.3481 & -0.4567 & 1.0006 & -0.0310 & 0.0152 \\
0.0006 & -0.0007 & 0.0028 & -0.0047 & 0.0096 & -0.0108 & 0.0141 & -0.0310 & 1.6747 & -0.8214 \\
-0.0003 & 0.0003 & -0.0014 & 0.0023 & -0.0047 & 0.0053 & -0.0069 & 0.0152 & -0.8214 & 1.0000
\end{pmatrix}$$

The CD sampler with was used to generate 1×10^6 samples from $N(\mathbf{0}, A^{-1})$, and the following sample covariance was calculated
$\hat{A}^{-1} =$

$$\begin{pmatrix}
1.9767 & -1.0307 & 0.3472 & -0.2071 & 0.1495 & -0.0962 & 0.0527 & -0.0183 & 0.0026 & -0.0009 \\
-1.0300 & 1.0857 & -0.3640 & 0.2178 & -0.1593 & 0.1016 & -0.0544 & 0.0205 & -0.0008 & 0.0003 \\
0.3472 & -0.3640 & 1.5750 & -0.9455 & 0.6965 & -0.4484 & 0.2414 & -0.0900 & 0.0029 & -0.0010 \\
-0.2071 & 0.2178 & -0.9455 & 1.5550 & -1.1455 & 0.7386 & -0.3975 & 0.1488 & -0.0065 & 0.0038 \\
0.1495 & -0.1593 & 0.6965 & -1.1455 & 2.3564 & -1.5192 & 0.8203 & -0.3103 & 0.0112 & -0.0059 \\
-0.0962 & 0.1016 & -0.4484 & 0.7386 & -1.5192 & 1.7027 & -0.9221 & 0.3488 & -0.0117 & 0.0061 \\
0.0527 & -0.0544 & 0.2414 & -0.3975 & 0.8203 & -0.9221 & 1.2074 & -0.4557 & 0.0158 & -0.0087 \\
-0.0183 & 0.0205 & -0.0900 & 0.1488 & -0.3103 & 0.3488 & -0.4557 & 1.0005 & -0.0348 & 0.0167 \\
0.0026 & -0.0008 & 0.0029 & -0.0065 & 0.0112 & -0.0117 & 0.0158 & -0.0348 & 1.6744 & -0.8214 \\
-0.0009 & 0.0003 & -0.0010 & 0.0038 & -0.0059 & 0.0061 & -0.0087 & 0.0167 & -0.8214 & 1.0001
\end{pmatrix}.$$

The close agreement provides a 'sanity check' that the CD sampler is indeed producing samples from the correct distribution.

4.2 Identity Covariance

As noted in section 3, the CD sampler fails to complete correctly when the precision matrix, or covariance matrix equals the $n \times n$ identity matrix. In that case preconditioning is necessary. When $n = 10$, the matrix U of the previous section is a suitable preconditioning matrix. Given samples from $N(0, (U^{\mathrm{T}}U)^{-1}))$ as in the previous section, applying the matrix U gives samples from $N(0, I_n)$.

4.3 An empirical upper bound on n

The results in section 2 show that the CD sampling algorithm 2 will generate samples from π when the (preconsditioned) covariance matrix has distinct eigenvalues, and computation is performed exactly.

However, it is well known that practical computation performed with finite precision causes Krylov-space methods such as CG or CD to eventually lose conjugacy of (search) sampling directions with consequent failure of the algorithm. The problem becomes worse with increasing n and increasing condition number of A since then the sequence of vectors $\left\{\mathbf{r}^{(0)}, A\mathbf{r}^{(0)}, A^2\mathbf{r}^{(0)}, \ldots, A^k\mathbf{r}^{(0)}\right\}$ become close to parallel for large k and hence are indistinguishable from linearly dependent vectors when computing to finite precision.

In that context, it is interesting to know what size of problem n one might realistically expect to successfully sample using the CD sampling algorithm. To find an effective upper bound for the sample dimension n, algorithm 2 was implemented[1] in MatLab (hence using IEEE double precision arithmetic) and run on a sequence of problems with increasing n. We chose a problem with a good, though not ideal, eigenvalue structure and so the bound we establish is probably optimistic in general.

The test problem uses a Gaussian process on n points equally spaced on the interval $[0, 1]$ with exponential covariance function. Rather than forming the dense covariance matrix, an equivalent formulation based on a finite element method discretization of a suitable variational form was used[2]. Specifically, we used linear elements on an equi-partition of $[0, 1]$ to form the discrete Hessian A of the quadratic form

$$\mathcal{I}(u) = \int_0^1 \left(\frac{r}{4c} \left(\frac{du}{dx} \right)^2 + \frac{1}{4rc} u^2 \right) dx + \frac{u(0)^2}{4c} + \frac{u(1)^2}{4c}$$

where $c = 1$ sets the covariance and $r = 0.1$ sets the length scale[3]. The covariance function is $\exp\{10x\}$.

The distribution of eigen-values of A for $n = 1000$ is shown in Figure 4.2. This $\sin^2$ shape is determined by the discrete Laplacian, for all n, while the maximum eigenvalue scales as n.

[1] The example presented here was coded by Al Parker, and included several sophisticated diagnostics for loss of conjugacy and other informative diagnostics. We will report on that work elsewhere.

[2] This formulation is based on Hilbert space methods for boundary value problems to show equivalence between locally-linear Gaussian processes and the definition using covariance functions. Finite element method discretizations provide usefully sparse representations of the precision matrix. The details of this formulation will be reported elsewhere.

[3] These can be functions of x.

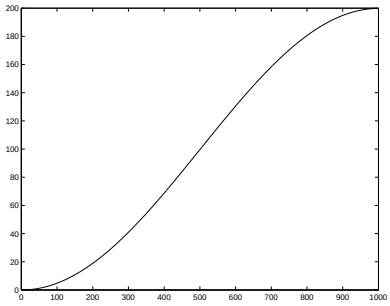


Figure 4.2. Eigenvalues of the precision matrix A plotted against index, in ascending order, for $n = 1000$.

For n much greater than 10^5 the algorithm fails to terminate properly, hence $n = 10^5$ sets an effective upper bound. A sample from $N(0, A^{-1}$ with $n = 10^5$ is shown in Figure 4.3 generated using each of the CD sampler and using Cholesky factoring of A.

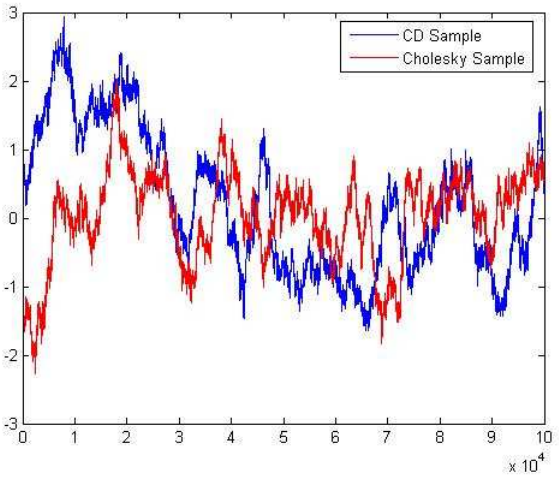


Figure 4.3. Two samples from $N(0, A)$ with $n = 10^5$.

In this case A is tridiagonal. Hence the Cholesky factor has $O(n)$ non-zero elements, can be computed in $O(n)$ operations, and allows samples to be generated in $O(n)$ operations. This is substantially cheaper than the $O(n^2)$ operations required for generating a sample using the CD algorithm in this example. Also $O(n)$ though faster than the method using Cholesky factoring is an algorithm based on a factorization of A in terms of local stiffness matrices.

4.4 A concluding note

We have successfully sampled problems with $n \geq 10^6$ using a block Gibbs sampler that uses the CD sampler to perform block-wise sampling. We expect that other modifications of the basic CD algorithm presented here will eventually lead to efficient algorithms that may be used for arbitrary problem sizes.

References

Owe Axelsson. *Iterative Solution Methods*. Cambridge University Press, 1994.

Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society Ser. B*, 36:192–236, 1974.

Peter Clifford. Discussion on the meeting on the Gibbs sampler and other statistical Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society*, 55:53–102, 1993.

William Feller. *An Introduction to Probability and its Applications*, volume 2. John Wiley & Sons, 1966.

Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

Walter R. Gilks, Gareth O. Roberts, and Edward I. George. Adaptive direction sampling. *The Statistician*, 43(1):179–189, 1994.

Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization*. Academic Press, London, 1981.

Tilmann Gneiting, Hana Ševčíková, Donald B. Percival, Martin Schlather, and Yindeng Jiang. Fast and exact simulation of large Gaussian lattice systems in $\mathbb{R}^2$: Exploring the limits. Technical Report 477, Department of Statistics, University of Washington, 2005.

Anne Greenbaum. *Iterative methods for solving linear systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997. ISBN 0-89871-396-X.

John M. Hammersley and Peter Clifford. Markov fields on finite graphs and lattices. *Unpublished*, 1971.

Magnus R. Hestenes and Eduard L. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Standards Sect. 5*, 49:409–436, 1952.

Alan Jennings. Influence of the eigenvalue spectrum on the convergence rate of the conjugate gradient method. *J. Inst. Maths Applics*, 20:61–72, 1977.

Jari Kaipio and Erkki Somersalo. *Statistical and Computational Inverse Problems*. Number 160 in Applied Mathematics. Springer, 2005.

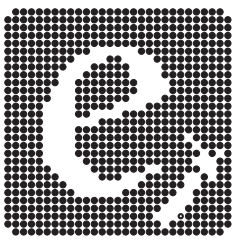
Jun S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2001. ISBN 0387952306.

Jun S. Liu, Faming Liang, and Wing Hung Wong. The use of multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association*, 95: 125–134, 2000.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 1999. ISBN 0387987932.

Håvard Rue. Fast sampling of Gaussian Markov random fields. *Journal of the Royal Statistical Society Ser. B*, 63:325–338, 2001.

Sze Meng Tan. *Aperture-synthesis mapping and parameter estimation.* PhD thesis, Cambridge University, 1986.



Electronics Group
Department of Physics
University of Otago
elec.otago.ac.nz