King Abdullah University
of Science and Technology

GMRFLib 2.0 (?)

Henrique Ap. Laureano
henrique.laureano@kaust.edu.sa ∧ http://mynameislaure.github.io/

/KAUST/CEMSE/STAT

Summer Semester
2018

Contents

Markov Random Fields

"Statistical modeling of a finite collection of spatial random variables is often done through a Markov random field (MRF). A MRF is specified through the set of conditional distributions of one component given all the others. This enables one to focus on a single random variable at a time and leads to simple computational procedures for simulating MRFs, in particular for Bayesian inference via Markov chain Monte Carlo (MCMC)." In the Gaussian (*aka* Normal) case, we have the so called Gaussian MRFs (GMRFs).

Gaussian Markov Random Fields

"A GMRF is simply a Gaussian distributed random vector $\boldsymbol{x}$, which obeys some conditional independence properties. That is, for some $i \neq j$, then

$$x_i \perp x_j \mid \boldsymbol{x}_{-\{i,j\}}, \tag{1}$$

meaning that conditioned on $\boldsymbol{x}_{-\{i,j\}}$, x_i and x_j are independent. This conditional independence is represented using an (undirected) labeled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, ..., n\}$ is the set of vertices, an $\mathcal{E} = \{\{i, j\} : i, j \in \mathcal{V}\}$ is the set of edges in the graph. For all $i, j \in \mathcal{V}$, the edge $\{i, j\}$ is not included in $\mathcal{E}$ if (1) holds, and included otherwise. Figure 1 displays such a graph, where $n = 4$ and $\mathcal{E} = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 1\}\}$. From this graph we deduce that $x_2 \perp x_4 \mid \boldsymbol{x}_{\{1,3\}}$ and $x_1 \perp x_3 \mid \boldsymbol{x}_{\{2,4\}}$. A central goal is now to specify a GMRF $\boldsymbol{x}$ with conditional independence properties in agreement with some given graph $\mathcal{G}$."

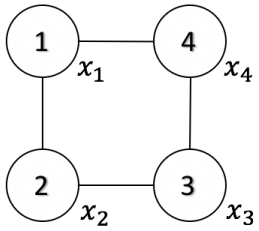


Figure 1: A conditional independence graph.

Theorem 1.
Let $\boldsymbol{x}$ be Gaussian distributed with a symmetric and positive definite (SPD) precision matrix $\boldsymbol{Q}$, then for $i \neq j$

$$x_i \perp x_j \mid \boldsymbol{x}_{-\{i,j\}} \quad \Longleftrightarrow \quad Q_{i,j} = 0.$$

"So any SPD precision matrix $\boldsymbol{Q}$ with $Q_{2,4} = Q_{4,2} = Q_{1,3} = Q_{3,1} = 0$ has conditional independence properties as displayed in Figure 1." A precision matrix $\boldsymbol{Q}$ that may correspond to this is presented in (2). "We then say that $\boldsymbol{x}$ is a GMRF with respect to $\mathcal{G}$. A formal definition follows."

$$\boldsymbol{Q} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} Q_{1,1} & Q_{1,2} & 0 & Q_{1,4} \\ Q_{2,1} & Q_{2,2} & Q_{2,3} & 0 \\ 0 & Q_{3,2} & Q_{3,3} & Q_{3,4} \\ Q_{4,1} & 0 & Q_{4,3} & Q_{4,4} \end{pmatrix} \end{array}. \tag{2}$$

Definition 1 (GMRF).
A random vector $\boldsymbol{x} = (x_1, \ldots, x_n)^\top \in \mathbb{R}^n$ is called a GMRF wrt the labeled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with mean $\boldsymbol{\mu}$ and SPD precision matrix $\boldsymbol{Q}$, iff its density has the form

$$\pi(\boldsymbol{x}) = (2\pi)^{-n/2} \, |\boldsymbol{Q}|^{1/2} \exp\Big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{Q}(\boldsymbol{x}-\boldsymbol{\mu})\Big) \tag{3}$$

and

$$Q_{i,j} \neq 0 \quad \Longleftrightarrow \quad \{i,j\} \in \mathcal{E} \quad \forall \quad i \neq j.$$

"The case where $\boldsymbol{Q}$ is singular still provides a GMRF with an explicit form for its joint density, but the joint density is improper. Such specifications cannot be used as data models, but can be used as priors as long as they yield proper posteriors. Here is a simple example of a (proper) GMRF."

Example 1.
"Let $\{x_t\}$ be a stationary autoregressive process of order one, *i.e.*, $x_t \mid x_{t-1} = \phi x_{t-1} + \epsilon_t$, for $t = 2, \ldots, n$, where $|\phi| < 1$ and ϵ_t are independent normally distributed zero mean innovations with unit variance. Further assume that x_1 is normal with mean zero and variance $1/(1\phi^2)$, which is simply the stationary distribution of this process. Then $\boldsymbol{x}$ is a GMRF *wrt* to $\mathcal{G}$ where $\mathcal{E} = \{\{1,2\},\{2,3\},\ldots,\{n-1,n\}\}$. The precision matrix has nonzero elements $Q_{i,j} = -\phi$ for $|i-j| = 1, Q_{1,1} = Q_{n,n} = 1$ and $Q_{i,i} = 1+\phi^2$ for $i = 2, \ldots, n-1$." Considering, *e.g.*, $n = 5$, the precision matrix $\boldsymbol{Q}$ is given by

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 1 & -\phi & 0 & 0 & 0 \\ -\phi & 1+\phi^2 & -\phi & 0 & 0 \\ 0 & -\phi & 1+\phi^2 & -\phi & 0 \\ 0 & 0 & -\phi & 1+\phi^2 & -\phi \\ 0 & 0 & 0 & -\phi & 1 \end{pmatrix} \end{array}.$$

"This example nicely illustrates why GMRFs are so useful." "Only $n+2(n-1) = 3n-2$ of the n^2 terms in $\boldsymbol{Q}$ are nonzero. The sparse precision matrix makes fast $\mathcal{O}(n)$ algorithms for the simulation of autoregressive processes possible."

Basic Properties

Conditional Properties

"Although a GMRF can be seen as a general multivariate Gaussian random variable, some properties simplify and some characteristics are easier to compute. For example, conditional distributions are easier to compute due to the sparse precision matrix. To see this, we split $\mathcal{V}$ into the nonempty sets A and $B = -A$. Partition $\boldsymbol{x}, \boldsymbol{\mu}$ and $\boldsymbol{Q}$ accordingly, *i.e.*,

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_A \\ \boldsymbol{x}_B \end{pmatrix}, \quad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{pmatrix} \quad \text{and} \quad \boldsymbol{Q} = \begin{pmatrix} \boldsymbol{Q}_{AA} & \boldsymbol{Q}_{AB} \\ \boldsymbol{Q}_{BA} & \boldsymbol{Q}_{BB} \end{pmatrix}.$$

We also need the notion of a subgraph $\mathcal{G}^A$, which is the graph restricted to A: the graph we obtain after removing all nodes not belonging to A and all edges where at least one node does not belong to A. Then the following theorem holds."

Theorem 2.
Let $\boldsymbol{x}$ be a GMRF wrt $\mathcal{G}$ with mean $\boldsymbol{\mu}$ and SPD precision matrix $\boldsymbol{Q}$. Let $A \subset \mathcal{V}$ and $B = \mathcal{V} \setminus A$ where $A, B \neq \emptyset$. The conditional distribution of $\boldsymbol{x}_A \mid \boldsymbol{x}_B$ is then a GMRF wrt the subgraph $\mathcal{G}^A$ with mean $\boldsymbol{\mu}_{A|B}$ and SPD precision matrix $\boldsymbol{Q}_{A|B}$, where

$$\boldsymbol{\mu}_{A|B} = \boldsymbol{\mu}_A - \boldsymbol{Q}_{AA}^{-1}\boldsymbol{Q}_{AB}(\boldsymbol{x}_B - \boldsymbol{\mu}_B) \quad \textit{and} \quad \boldsymbol{Q}_{A|B} = \boldsymbol{Q}_{AA}.$$

"The expression for the conditional mean $\boldsymbol{\mu}_{A|B}$ involves the inverse $\boldsymbol{Q}_{AA}^{-1}$, but only in a way such that we can write $\boldsymbol{\mu}_{A|B} = \boldsymbol{\mu}_A - \boldsymbol{b}$, where $\boldsymbol{b}$ is the solution of a sparse linear system $\boldsymbol{Q}_{AA}\boldsymbol{b} = \boldsymbol{Q}_{AB}(\boldsymbol{x}_B - \boldsymbol{\mu}_B)$. Note that the term $\boldsymbol{Q}_{AB}$ is nonzero only for those vertices in A that have an edge to a vertex in B, so usually only a few terms will enter in this matrix-vector product. In the special case $A = \{i\}$, the expressions simplify to

$$\mu_{i|-i} = \mu_i - \sum_{j:j\sim i} \frac{Q_{i,j}}{Q_{i,i}}(x_j - \mu_j) \quad \text{and} \quad Q_{i|-i} = Q_{i,i}. \tag{4}$$

Here we used the notation $j : j \sim i$ to indicate a sum over all vertices j that are neighbors to vertex i, *i.e.*, $\{i, j\} \in \mathcal{E}$. So $Q_{i,i}$ is the conditional precision of x_i and the conditional expectation of x_i is a weighted mean of neighboring x_js with weights $-Q_{i,j}/Q_{i,i}$."

Example 2.
"We continue with Example 1. From (4) we obtain the conditional mean and precision of $x_i \mid \boldsymbol{x}_{-i}$,"

$$\begin{aligned}
\mu_{i|-i} &= 0 - \left[\frac{-\phi}{1+\phi^2}(x_{i-1} - 0) + \frac{-\phi}{1+\phi^2}(x_{i+1} - 0)\right] \\
&= 0 - \frac{-\phi}{1+\phi^2}(x_{i-1} + x_{i+1}) \\
&= \frac{\phi}{1+\phi^2}(x_{i-1} + x_{i+1}), \quad \text{and} \quad Q_{i|-i} = 1 + \phi^2, \quad 1 < i < n.
\end{aligned}$$

Markov Properties

"The graph $\mathcal{G}$ of a GMRF is defined through looking at which x_i and x_j are conditionally independent, the so-called *pairwise* Markov property. However, more general Markov properties can be derived from $\mathcal{G}$.

A *path* from vertex i_1 to vertex i_m is a sequence of distinct nodes in $\mathcal{V}, i_1, i_2, \ldots, i_m$, for which $(i_j, i_{j+1}) \in \mathcal{E}$ for $j = 1, \ldots, m-1$. A subset $C \subset \mathcal{V}$ *separates* two nodes $i \notin C$ and $j \notin C$, if every path from i to j contains at least one node from C. Two disjoint sets $A \subset \mathcal{V} \setminus C$ and $B \subset \mathcal{V} \setminus C$ are separated by C , if all $i \in A$ and $j \in B$ are separated by C. In other words, we cannot walk on the graph starting somewhere in A ending somewhere in B without passing through C. The global Markov property, is that

$$\boldsymbol{x}_A \perp \boldsymbol{x}_B \mid \boldsymbol{x}_C$$

for all mutually disjoint sets A, B and C where C separates A and B, and A and B are nonempty."

Theorem 3.
Let $\boldsymbol{x}$ be a GMRF wrt $\mathcal{G}$, then $\boldsymbol{x}$ obeys the global Markov property.

Conditional Specification

"It is common to specify a GMRF implicitly through the so-called full conditionals $\{\pi(x_i \mid \boldsymbol{x}_{-i})\}$." "However, the full conditionals cannot be specified completely arbitrarily, as we must ensure that they correspond to a proper joint density.

A conditional specification defines the full conditional $\{\pi(x_i \mid \boldsymbol{x}_{-i})\}$ as normal with moments

$$\mathbb{E}(x_i \mid \boldsymbol{x}_{-i}) = \mu_i + \sum_{j \neq i} \beta_{i,j}(x_j - \mu_j) \quad \text{and} \quad \text{Precision}(x_i \mid \boldsymbol{x}_{-i}) = \kappa_i > 0. \tag{5}$$

The rationale for such an approach, is that it is easier to specify the full conditionals than the joint distribution. Comparing (5) with (3), we can choose $\boldsymbol{\mu}$ as the mean, $Q_{i,i} = \kappa_i$, $\beta_{i,j} = -Q_{i,j}/Q_{i,i}$ to obtain the same full conditionals. However, since $\boldsymbol{Q}$ is symmetric, we must require that

$$\kappa_i \beta_{i,j} = \kappa_j \beta_{j,i} \tag{6}$$

for all $i \neq j$." "In addition to the symmetry constraint (6), there is a joint requirement that $\boldsymbol{Q}$ is SPD. Unfortunately, this is a *joint* property, which is hard to validate locally. One convenient approach that avoids this problem is to choose a diagonally dominant parametrization that ensures $\boldsymbol{Q}$ to be SPD: $Q_{i,i} > \sum_j |Q_{i,j}|$ for all i. This implies that"

$$\sum_j |\beta_{i,j}| < 1, \quad \forall i.$$

Fixing $\boldsymbol{\mu} = \mathbf{0}$, using the full conditionals in Equation (5), and considering the symmetry constraint (6), the density of $\boldsymbol{x}$ can then be expressed as

$$\begin{aligned}
\log \pi(\boldsymbol{x}) &= \text{const} + \frac{1}{2} \log |\boldsymbol{Q}| - \frac{1}{2} \boldsymbol{x}^\top \boldsymbol{Q} \boldsymbol{x} \quad \text{(starting from (3))} \\
&= \text{const} + \frac{1}{2} \sum_{i \neq j} Q_{i,j} x_i x_j - \frac{1}{2} \sum_{i=1}^{n} Q_{i,i} x_i^2 \\
&= \text{const} - \frac{1}{2} \sum_{i \neq j} \kappa_i \beta_{i,j} x_i x_j - \frac{1}{2} \sum_{i=1}^{n} \kappa_i x_i^2;
\end{aligned}$$

"hence, $\boldsymbol{x}$ is zero mean GMRF provided $\boldsymbol{Q}$ is SPD."

Example 3.
"The image in Figure 2(a) is a 256×256 gamma camera image of a phantom designed

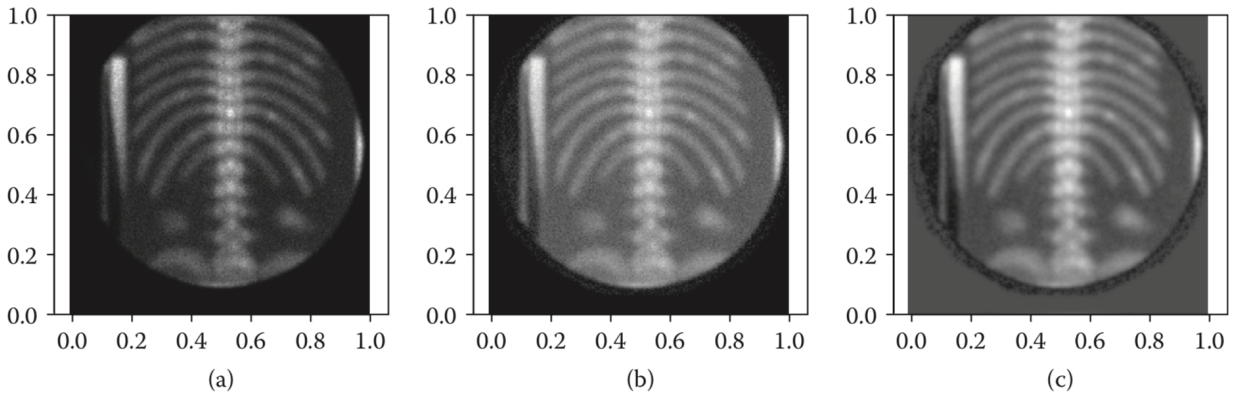


Figure 2: Panel (a) shows the raw x-ray image, (b) shows the square-root transformed image, and (c) shows the restored image (the posterior mean).

to reflect structure expected from cancerous bones. Each pixel in the circular part of the image, $\mathcal{I}$, represent the gamma radiation count, where a black pixel represents (essentially) zero counts and a white pixel the maximum count. The image is quite noisy and the task in this example is to (try to) remove the noise. The noise process is quite accurately described by a Poisson distribution, so that for each pixel i, the recorded count y_i relates to the true signal η_i, as $y_i \sim \text{Poisson}(\eta_i)$. For simplicity, we will use the approximation that"

$$\sqrt{y_i} \mid \eta_i \sim \mathcal{N}\left(\sqrt{\eta_i}, \frac{1}{4}\right), \quad i \in \mathcal{I}$$

"and the square-root transformed image is displayed in Figure 2(b). Taking a Bayesian approach, we need to specify a prior distribution for the (square-root-transformed) image $\boldsymbol{x} = (x_1, \ldots, x_n)^\top$, where $x_i = \sqrt{\eta_i}$. (We need η_i to be (somewhat) larger than zero for this approximation to be adequate.) Although this is a daunting problem in general, for such noise-removal tasks it is usually sufficient to specify the prior to be informative for how the true image behaves locally. Since the image itself is locally smooth, we might specify the prior through the full conditionals (5). Using the full conditionals we only need to answer questions like: *What if we do not know the true signal in pixel i, but all others; what is then our belief in x_i?* One choice, is to set $\beta_{i,j}$ to zero unless j is one of the four nearest neighbors of i; $N_4(i)$, say. As we have no particular preference for direction, we might take for each i,

$$\beta_{i,j} = \frac{\delta}{4}, \quad j \in N_4(i)$$

where δ is considered as fixed. Further, we take κ_i to be common (and unknown) for all i, and restrict δ to $|\delta| < 1$ so that the (prior) precision matrix is diagonally dominant. (We ignore here some corrections at the boundary where a boundary pixels may have less than four neighbors.) We take further $\boldsymbol{\mu} = \mathbf{0}$ and a (conjugate) $\Gamma(a, b)$ prior for κ (with density $\propto \kappa^{a-1} \exp(-b\kappa)$), and then the posterior for $(\boldsymbol{x}, \kappa)$ reads

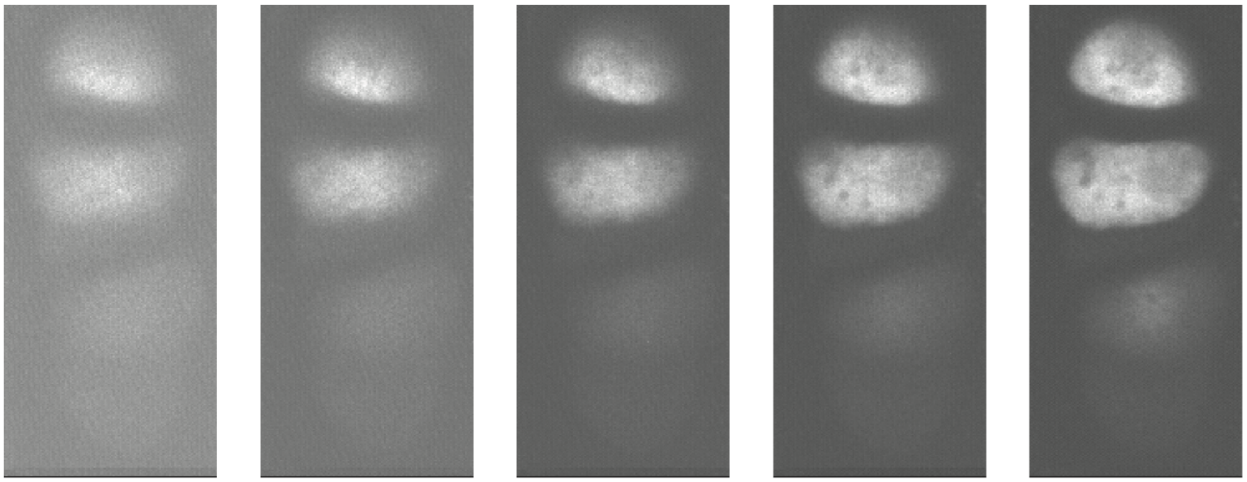


Figure 3: Panels (a) to (e) show five consecutive frames of a three-dimensional confocal microscopy image.

$$\begin{aligned}\pi(\boldsymbol{x}, \kappa \mid \boldsymbol{y}) &\propto \pi(\boldsymbol{x}, \kappa)\pi(\kappa) \prod_{i \in \mathcal{I}} \pi(y_i \mid x_i) \\ &\propto \kappa^{a-1} \exp(-b\kappa) \, |\boldsymbol{Q}_{prior}(\kappa)|^{1/2} \exp\Big(-\frac{1}{2} \boldsymbol{x}^\top \boldsymbol{Q}_{post}(\kappa) \boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{x} \Big).\end{aligned}$$

Here, $b_i = 4\sqrt{y_i}$ for $i \in \mathcal{I}$ and zero otherwise, $\boldsymbol{Q}_{post}(\kappa) = \boldsymbol{Q}_{prior}(\kappa) + \boldsymbol{D}$ where $\boldsymbol{D}$ is a diagonal matrix where $D_{i,i} = 4$ if $i \in \mathcal{I}$ and zero otherwise, and

$$\boldsymbol{Q}_{prior}(\kappa)_{i,j} = \kappa \begin{cases} 1, & i = j \\ \delta/4, & j \in N_4(i) \\ 0, & \text{otherwise.} \end{cases}$$

Conditioned on κ and the observations, then $\boldsymbol{x}$ is a GMRF with precision matrix $\boldsymbol{Q}_{post}$ and where the mean $\boldsymbol{\mu}_{post}$ is given by the solution of"

$$\boldsymbol{Q}_{post} \boldsymbol{\mu}_{post} = \boldsymbol{b}.$$

Multivariate GMRFs

"To fix ideas, we will consider a generalization of Example 3 where the observations are now sequences of images. The sequence can either be a movie where each frame in the sequence is indexed by time, or the height where recorded a three-dimensional object as a set of two-dimensional images. Other examples include a temporal version of spatial models of disease counts in each administrative region of a country. Figure 3 shows five consecutive frames of three-dimensional cells taken by confocal microscopy. The first

frame has a lot of noise, but the signal gets stronger farther up in the image stack. We consider the same problem as for Example 3; we want to estimate the true signal in the presence of the noise. The five frames represent the same three-dimensional object, but at different height.

We can use this information when we specify the full conditionals. It is then both easier and more natural to specify a multivariate version of the full conditionals (5), which we now will describe. Let $\boldsymbol{x}_i$ represent all the $p = 5$ observations at pixel i

$$\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4}, x_{i,5})^\top.$$

Here, $x_{i,2}$ is the pixel at location i in frame 2 and so on. The conditional specification (5) extends naturally to

$$\mathbb{E}(\boldsymbol{x}_i \mid \boldsymbol{x}_{-i}) = \boldsymbol{\mu}_i - \sum_{j:j\sim i} \boldsymbol{\beta}_{i,j}(\boldsymbol{x}_j - \boldsymbol{\mu}_j) \quad \text{and} \quad \text{Precision}(\boldsymbol{x}_i \mid \boldsymbol{x}_{-i}) = \boldsymbol{\kappa}_i > 0, \tag{7}$$

for some $p \times p$ matrices $\{\boldsymbol{\beta}_{i,j}\}$ and $\{\boldsymbol{\kappa}_i\}$. In this formulation, we can now specify that our knowledge of $x_{i,3}$ might benefit of knowing $x_{i,2}$ and $x_{i,4}$. These pixels are in the same $\boldsymbol{x}_i$ vector, although they represent the ith pixel at the previous and next frame. Additionally, we can have dependency from neighbors within the same frame, such as $\{x_{j,3}, j \in N_4(i)\}$. In short, we can specify how $\boldsymbol{x}_i$ depends on $\{\boldsymbol{x}_j\}, j \neq i$, and thinking about neighbors that are (small p-) vectors.

The conditional specification in this example motivates the introduction of a multivariate GMRF, which we denote as MGMRF$_p$. Its definition is a direct extension of (1). Let $\boldsymbol{x} = (\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_n^\top)^\top$ be Gaussian distributed, where each $\boldsymbol{x}_1$ is a p-vector. Similarly, let $\boldsymbol{\mu} = (\boldsymbol{\mu}_1^\top, \ldots, \boldsymbol{\mu}_n^\top)^\top$ denote the mean and $\widetilde{\boldsymbol{Q}} = (\widetilde{\boldsymbol{Q}}_{i,j})$ the precision matrix with $p \times p$ elements $\widetilde{\boldsymbol{Q}}_{i,j}$."

Definition 2 (MGMRF$_p$).
A random vector $\boldsymbol{x} = (\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_n^\top)^\top$ where dim$(\boldsymbol{x}_i) = p$, is called a MGMRF$_p$ wrt $\mathcal{G} = (\mathcal{V} = \{1, \ldots, n\}, \mathcal{E})$ with mean $\boldsymbol{\mu}$ and SPD precision matrix $\widetilde{\boldsymbol{Q}}$, iff its density has the form

$$\begin{aligned}\pi(\boldsymbol{x}) &= (2\pi)^{-np/2} \left|\widetilde{\boldsymbol{Q}}\right|^{1/2} \exp\Big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \widetilde{\boldsymbol{Q}}(\boldsymbol{x}-\boldsymbol{\mu})\Big) \\ &= (2\pi)^{-np/2} \left|\widetilde{\boldsymbol{Q}}\right|^{1/2} \exp\Big(-\frac{1}{2}\sum_{ij}(\boldsymbol{x}_i-\boldsymbol{\mu}_i)^\top \widetilde{\boldsymbol{Q}}_{i,j}(\boldsymbol{x}_j-\boldsymbol{\mu}_j)\Big)\end{aligned}$$

and

$$\widetilde{\boldsymbol{Q}}_{i,j} \neq \boldsymbol{0} \quad \Longleftrightarrow \quad \{i,j\} \in \mathcal{E} \quad \forall \quad i \neq j.$$

"It is important to note that a size n MGMRF$_p$ is just another GMRF of dimension np; so all our previous results and forthcoming sparse matrix algorithms for GMRFs also apply for a MGMRF$_p$. However, some results have easier interpretation using the block formulation, such as

$$\boldsymbol{x}_i \perp \boldsymbol{x}_j \mid \boldsymbol{x}_{-\{i,j\}} \quad \Longleftrightarrow \quad \widetilde{\boldsymbol{Q}}_{i,j} = \boldsymbol{0}$$

and

$$\mathbb{E}(\boldsymbol{x}_i \mid \boldsymbol{x}_{-i}) = \boldsymbol{\mu}_i - \widetilde{\boldsymbol{Q}}_{i,j}^{-1} \sum_{j:j\sim i} \widetilde{\boldsymbol{Q}}_{i,j}(\boldsymbol{x}_j - \boldsymbol{\mu}_j) \quad \text{and} \quad \text{Precision}(\boldsymbol{x}_i \mid \boldsymbol{x}_{-i}) = \widetilde{\boldsymbol{Q}}_{i,i}. \tag{8}$$

From Equation (8), we can obtain the consistency requirements for the conditional specification Equation (7) by choosing

$$\widetilde{\boldsymbol{Q}}_{i,j} = \begin{cases} \boldsymbol{\kappa}_i \boldsymbol{\beta}_{i,j}, & i \neq j \\ \boldsymbol{\kappa}_i, & i = j. \end{cases}$$

Since $\widetilde{\boldsymbol{Q}}_{i,j} = \widetilde{\boldsymbol{Q}}_{j,i}^\top$, then we have the requirement that $\boldsymbol{\kappa}_i \boldsymbol{\beta}_{i,j} = \boldsymbol{\beta}_{j,i}^\top \boldsymbol{\kappa}_j$ for $i \neq j$, additionally to $\boldsymbol{\kappa}_i > 0 \quad \forall i$. Finally, there is also the "global" requirement that $\widetilde{\boldsymbol{Q}}$ must be SPD, which is equivalent to $(\boldsymbol{I} + (\boldsymbol{\beta}_{i,j}))$ being SPD."

Exact Algorithms for GMRFs

GMRFs have "a nice connection with very efficient numerical algorithms for sparse matrices. This connection allows for exact algorithms for GMRFs. We will now discuss this connection, starting with various exact algorithms to efficiently sample from GMRFs. This includes solving tasks like unconditional and conditional sampling, sampling under linear hard and soft constraints, evaluating the log-density of a (possibly constrained) GMRF at a particular value, and computing marginal variances for (possibly constrained) GMRFs. Although all these tasks are formally "just matrix algebra," we need to ensure that we take advantage of the sparse precision matrix $\boldsymbol{Q}$ in all steps so computations can make use of the efficient numerical algorithms for sparse matrices developed in the computational sciences literature. Further, we can derive all the algorithms for sparse matrices by considering conditional independence properties of GMRFs. The core of all algorithms is the Cholesky factorization $\boldsymbol{Q} = \boldsymbol{L}\boldsymbol{L}^\top$ of the precision matrix $\boldsymbol{Q}$, where $\boldsymbol{L}$ is a lower-triangular matrix."

Why are Exact Algorithms Important?

"Exact efficient algorithms are generally preferable when they exist, even though they apparently require algorithms that are more involved than simple iterative ones. Computational feasibility is important even for statistical modeling, as a statistical model is not of much use if we cannot do inference efficiently enough to satisfy the end-user.

Sampling from a GMRF can be done exactly using the Cholesky factorization of the precision matrix". "In the spatial case, it turns out that we can (typically) sample a GMRF exactly at the cost of $\mathcal{O}(n^{3/2})$ operations". "The exact algorithm can further produce independent samples at the cost of $\mathcal{O}(n \log n)$ each."

Some Basic Linear Algebra

"Let A be SPD, then there exists a unique (Cholesky) factorization $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^\top$, where L is lower triangular and called the Cholesky triangle. This factorization is the starting point for solving $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}$ by first solving $\boldsymbol{L}\boldsymbol{v} = \boldsymbol{b}$ and then $\boldsymbol{L}^\top\boldsymbol{y} = \boldsymbol{v}$. The first linear system $\boldsymbol{L}\boldsymbol{v} = \boldsymbol{b}$ is solved directly using forward substitution

$$v_i = \frac{1}{L_{i,i}}\Big(b_i - \sum_{j=1}^{i-1} L_{i,j}v_j\Big), \quad i = 1, \ldots, n,$$

whereas $\boldsymbol{L}^\top\boldsymbol{y} = \boldsymbol{v}$ is solved using backward substitution

$$y_i = \frac{1}{L_{i,i}}\Big(v_i - \sum_{j=1+1}^{n} L_{j,i}y_j\Big), \quad i = n, \ldots, 1.$$

Computing $\boldsymbol{A}^{-1}\boldsymbol{Y}$, where $\boldsymbol{Y}$ is a $n \times k$ matrix, is done by computing $\boldsymbol{A}^{-1}\boldsymbol{Y}_j$ for each of the k columns $\boldsymbol{Y}_j$ using the algorithm above. Note that $\boldsymbol{A}$ needs to be factorized only once. Note that in the case $\boldsymbol{Y} = \boldsymbol{I}$ (and $k = n$), the inverse of $\boldsymbol{A}$ is computed".

Sampling from a GMRF

"Sampling from a GMRF can be done using the following steps: sample $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, *i.e.*, n standard normal variables, solve $\boldsymbol{L}^\top\boldsymbol{v} = \boldsymbol{z}$, and compute $\boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{v}$. The sample $\boldsymbol{x}$ has the correct distribution as $\mathbb{E}(\boldsymbol{v}) = \boldsymbol{0}$ and $\text{Cov}(\boldsymbol{v}) = \boldsymbol{L}^{-\top}\boldsymbol{I}\boldsymbol{L}^{-1} = (\boldsymbol{L}\boldsymbol{L}^\top)^{-1} = \boldsymbol{Q}^{-1}$. The log-density of $\boldsymbol{x}$,

$$\log \pi(\boldsymbol{x}) = -\frac{n}{2}\log 2\pi + \frac{1}{2}\log|\boldsymbol{Q}| - \frac{1}{2}\underbrace{(\boldsymbol{x}-\boldsymbol{\mu})^\top\boldsymbol{Q}(\boldsymbol{x}-\boldsymbol{\mu})}_{q}$$

is evaluated as follows. If $\boldsymbol{x}$ is sampled using the algorithm above, then $q = \boldsymbol{z}^\top\boldsymbol{z}$, otherwise, we compute $\boldsymbol{w} = \boldsymbol{x} - \boldsymbol{\mu}$, $\boldsymbol{u} = \boldsymbol{Q}\boldsymbol{w}$ and then $q = \boldsymbol{w}^\top\boldsymbol{u}$. Note that $\boldsymbol{u} = \boldsymbol{Q}\boldsymbol{w}$ is a sparse-matrix vector product, which can be computed efficiently:

$$u_i = Q_{i,i}w_i + \sum_{j:j\sim i} Q_{i,j}w_j,$$

where the diagonal term is added explicitly since i is not a neighbor of i. The determinant of $\boldsymbol{Q}$ can be found from the Cholesky factorization: $|\boldsymbol{Q}| = \left|\boldsymbol{L}\boldsymbol{L}^\top\right| = |\boldsymbol{L}|^2$. Since $\boldsymbol{L}$ is lower triangular, we obtain

$$\frac{1}{2}\log|\boldsymbol{Q}| = \sum_i \log L_{i,i}.$$

Conditional sampling of $\boldsymbol{x}_A$ conditioned on $\boldsymbol{x}_B$, as described in Theorem 2, is similar: factorize $\boldsymbol{Q}_{AA}$ and *solve*

$$\boldsymbol{Q}_{AA}(\boldsymbol{\mu}_{A|B} - \boldsymbol{\mu}_A) = -\boldsymbol{Q}_{AB}(\boldsymbol{x}_B - \boldsymbol{\mu}_B)$$

for $\boldsymbol{x}_{A|B}$, using forward and backward substitution. The (sparse) matrix vector product on the right-hand side is computed using only the nonzero terms in $\boldsymbol{Q}_{AB}$. The remaining steps are the same."

Sampling from a GMRF Conditioned on Linear Constraints

"In practical applications, we often want to sample from a GMRF under a linear constraint,

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{e},$$

for a $k \times n$ matrix $\boldsymbol{A}$ of rank k. The common case is that $k \ll n$. A brute-force approach is to directly compute the conditional (Gaussian) density $\pi(\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{e})$, but this will reveal that the corresponding precision matrix is (usually) not sparse anymore. For example, if $x_i \perp x_j \mid \boldsymbol{x}_{-\{i,j\}}$ without the constraint, then a sum-to-zero constraint $\sum x_i = 0$ makes x_i and x_j negatively correlated. In order not to lose computational efficiency, we must approach this problem in a more subtle way by correcting an unconstrained sample $\boldsymbol{x}$ to obtain a constrained sample $\boldsymbol{x}^c$:

$$\boldsymbol{x}^c = \boldsymbol{x} - \boldsymbol{Q}^{-1}\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^\top)^{-1}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{e}). \tag{9}$$

A direct calculation shows that $\boldsymbol{x}^c$ has the correct mean and covariance. A closer look at (9) makes it clear that all the matrix terms are easy to compute: $\boldsymbol{Q}^{-1}\boldsymbol{A}^\top$ just solves k linear systems of type $\boldsymbol{Q}\boldsymbol{v}_j = (\boldsymbol{A}^\top)_j$, for $j = 1, \ldots, k$, whereas $\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^\top$ is a $k \times k$ matrix and its Cholesky factorization is fast to compute since k is small in typical applications. Note that the extra cost of having k constraints is $\mathcal{O}(nk^2)$, hence, negligible when k is small.

Evaluating the constrained log density perhaps needs more attention, since $\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x}$ is singular with rank $n - k$. However, the following identity can be used:

$$\pi(\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x}) = \frac{\pi(\boldsymbol{A}\boldsymbol{x} \mid \boldsymbol{x})\pi(\boldsymbol{x})}{\pi(\boldsymbol{A}\boldsymbol{x})}. \tag{10}$$

Note now that all terms on the right-hand side are easy to compute: $\pi(\boldsymbol{x})$ is a GMRF with mean $\boldsymbol{\mu}$ and precision matrix $\boldsymbol{Q}$, $\pi(\boldsymbol{A}\boldsymbol{x})$ is (k-dimensional) Gaussian with mean $\boldsymbol{A}\boldsymbol{\mu}$ and covariance $\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^\top$, while $\pi(\boldsymbol{A}\boldsymbol{x} \mid \boldsymbol{x})$ is either 0 (when the configuration $\boldsymbol{x}$ is inconsistent with the value of $\boldsymbol{A}\boldsymbol{x}$ or equal to $\left|\boldsymbol{A}\boldsymbol{A}^\top\right|^{-1/2}$."

Example 4.
"Let $\boldsymbol{x}$ be n independent, zero-mean, normal random variables with variance $\{\sigma_i^2\}$. A sum-to-zero constrained sample $\boldsymbol{x}^*$ can be generated from a sample of the unconstrained $\boldsymbol{x}$ using

$$x_i^* = x_i - c\sigma_i^2, \quad \text{where} \quad c = \sum x_j / \sum \sigma_j^2, \quad i = 1, \ldots, n.$$

The above construction can be generalized to condition on so-called *soft constraints*, which we condition on the k observations $\boldsymbol{y} = (y_1, \ldots, y_k)^\top$, where

$$\boldsymbol{y} \mid \boldsymbol{x} \sim \mathcal{N}(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{\Upsilon}).$$

Here, $\boldsymbol{A}$ is a $k \times n$ matrix with rank k and $\boldsymbol{\Upsilon} > 0$. The conditional density for $\boldsymbol{x} \mid \boldsymbol{y}$ has precision matrix $\boldsymbol{Q} + \boldsymbol{A}^\top \boldsymbol{\Upsilon}^{-1} \boldsymbol{A}$, which is often a dense matrix. We can use the same approach as in Equation (9), which now generalizes to

$$\boldsymbol{x}^c = \boldsymbol{x} - \boldsymbol{Q}^{-1}\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^\top + \boldsymbol{\Upsilon})^{-1}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{\epsilon}), \tag{11}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{y}, \boldsymbol{\Upsilon})$. Similarly, if $\boldsymbol{x}$ is a sample from $\pi(\boldsymbol{x})$ and then $\boldsymbol{x}^c$ computed from Equation (11) is distributed as $\pi(\boldsymbol{x} \mid \boldsymbol{y})$. To evaluate the conditional density, we use the same approach as in Equation (10), which now reads

$$\pi(\boldsymbol{x} \mid \boldsymbol{y}) = \frac{\pi(\boldsymbol{y} \mid \boldsymbol{x})\pi(\boldsymbol{x})}{\pi(\boldsymbol{y})}.$$

Here, $\pi(\boldsymbol{x})$ is the density for a GMRF, $\pi(\boldsymbol{y} \mid \boldsymbol{x})$ is the density for a k-dimensional Gaussian with mean $\boldsymbol{A}\boldsymbol{x}$ and covariance matrix $\boldsymbol{\Upsilon}$, whereas $\pi(\boldsymbol{y})$ is the density for a k-dimensional Gaussian with mean $\boldsymbol{A}\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^\top + \boldsymbol{\Upsilon}$."

The Cholesky Factorization of Q

"All the exact simulation algorithms are based on the Cholesky triangle $\boldsymbol{L}$, which is found by factorizing $\boldsymbol{Q}$ into $\boldsymbol{L}\boldsymbol{L}^\top$. We will now discuss this factorization in more detail, show why sparse matrices allow for faster factorization, and how reordering the indices can speed up the computations."

Interpretation of the Cholesky Triangle

"The Cholesky factorization of $\boldsymbol{Q}$ is explicitly available; it is just a matter of doing the computations in the correct order. By definition, we have

$$Q_{i,j} = \sum_{k=1}^{j} L_{i,k}L_{j,k}, \quad i \geq j,$$

where we have used that $\boldsymbol{L}$ is lower triangular meaning that $L_{i,k} = 0, \forall k > i$. To fix ideas, assume $n = 2$, so that $Q_{1,1} = L_{1,1}^2$, $Q_{2,1} = L_{2,1}L_{1,1}$ and $Q_{2,2} = L_{2,1}L_{2,1} + L_{2,2}^2$. Then we see immediately that we can compute $L_{1,1}$, $L_{2,1}$ and $L_{2,2}$ in this particular order. This generalizes for $n > 2$; we can compute $L_{i,1}$ for $i = 1, \ldots, n$ (in this order), then $L_{i,2}$ for $i = 2, \ldots, n$, and so on. Due to this simple explicit structure, the Cholesky factorization can be computed quickly". "However, the complexity of the computations is of order $\mathcal{O}(n^3)$.

The natural way to speed up the Cholesky factorization is to make use of a particular structure in the precision matrix or the Cholesky triangle. For GMRFs, the issue is that sparsity in $\boldsymbol{Q}$ implies (a related) sparsity in $\boldsymbol{L}$. The implication is that if we *know* that $L_{j,i} = 0$, then we do not need to compute it. And if the main bulk of $\boldsymbol{L}$ is zero, then we can achieve great computational savings. In order to understand these issues, we need

to understand what $\boldsymbol{L}$ really means in terms of statistical interpretation. The simulation algorithm for a zero mean GMRFs, which solves $\boldsymbol{L}^\top \boldsymbol{x} = \boldsymbol{z}$, where $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ gives the following result immediately."

Theorem 4.
Let $\boldsymbol{x}$ be a GMRF wrt the labeled graph $\mathcal{G}$, mean $\boldsymbol{\mu}$, and a SPD precision matrix $\boldsymbol{Q}$. Let $\boldsymbol{L}$ be the Cholesky triangle of $\boldsymbol{Q}$. Then for $i \in \mathcal{V}$,

$$\mathbb{E}(x_i \mid \boldsymbol{x}_{(i+1):n}) = \mu_i - \frac{1}{L_{i,i}} \sum_{j=i+1}^{n} L_{j,i}(x_j - \mu_j) \quad \textit{and}$$
$$\textit{Precision}(x_i \mid \boldsymbol{x}_{(i+1):n}) = L_{i,i}^2.$$

"Hence the elements of $\boldsymbol{L}$ have an interpretation as the contribution to the conditional mean and precision for x_i, given all those x_js where $j > i$. This is in contrast to the elements of $\boldsymbol{Q}$, which have a similar interpretation, but where we condition on all other x_js. A simple consequence of this interpretation is that $Q_{i,i} \geq L_{i,i}^2, \forall i$.

If we merge Equation (3) with Theorem 4, we obtain the following result."

Theorem 5.
Let $\boldsymbol{x}$ be a GMRF wrt to the labeled graph $\mathcal{G}$, with mean $\boldsymbol{\mu}$ and SPD precision matrix $\boldsymbol{Q}$. Let $\boldsymbol{L}$ be the Cholesky triangle of $\boldsymbol{Q}$ and define for $1 \leq i < j \leq n$ the future of i except j as

$$F(i,j) = \{i+1, \ldots, j-1, j+1, \ldots, n\}.$$

Then

$$x_i \perp x_j \mid \boldsymbol{x}_{F(i,j)} \quad \Longleftrightarrow \quad L_{j,i} = 0.$$

"So, if we consider the marginal distribution of $\boldsymbol{x}_{i:n}$, then $L_{j,i} = 0$ is equivalent to x_i and x_j being conditionally independent. This is a useful result, as it indicates that if we can determine zeros in $\boldsymbol{L}$ using conditional independence in the sequence of marginals $\{\boldsymbol{x}_{i:n}\}$, the marginals $\{\boldsymbol{x}_{i:n}\}$ are easier to compute through the Cholesky triangle. However, we can use a weaker criterion, which implies that $x_i \perp x_j \mid \boldsymbol{x}_{F(i,j)}$, the global Markov property in Theorem 3."

Corollary 1.
If $F(i,j)$ separates $i < j$ in $\mathcal{G}$, then $L_{j,i} = 0$.

"This is the main result. If we can verify that $i < j$ are separated by $F(i,j)$, a operation that only depends on the graph and not the numerical values in $\boldsymbol{Q}$, then we know that $L_{j,i} = 0$ no matter what the numerical values in $\boldsymbol{Q}$ are. If $i < j$ are not separated by $F_(i,j)$, then $L_{j,i}$ *can* be zero, but is in general nonzero. Since two neighbors $i \sim j$ are not separated by any set, then $L_{j,i}$ is in general nonzero for neighbors."

Example 5.
"Consider a GMRF with graph as in Figure 1. We then know that $L_{1,1}$, $L_{2,2}$, $L_{3,3}$, $L_{4,4}$ is nonzero, and $L_{2,1}$, $L_{3,2}$, $L_{4,3}$, and $L_{4,1}$ is in general nonzero. The two elements remaining are $L_{3,1}$ and $L_{4,2}$, which we check using Corollary 1; nodes 1 and 3 are separated by $F(1,3) = \{2,4\}$, so $L_{3,1}$ must be zero, whereas $F(4,2) = \{3\}$ does not separate 2 and 4 due to node 1, hence $L_{4,2}$ is in general nonzero." A representation of this $\boldsymbol{L}$ can be seen in (12), with the zero elements represented as empty spaces.

$$\boldsymbol{L} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} L_{1,1} & & & \\ L_{2,1} & L_{2,2} & & \\ & L_{3,2} & L_{3,3} & \\ L_{4,1} & L_{4,2} & L_{4,3} & L_{4,4} \end{pmatrix} \end{array}. \tag{12}$$

Cholesky Factorization of Band Matrices

"Although only one element of the Cholesky triangle in Example 5 was necessarily zero, we obtain a larger amount of zeros for autoregressive models. Let $\boldsymbol{x}$ be a pth order autoregressive process, AR(p),

$$x_t \mid x_{t-1}, \ldots, x_1 \sim \mathcal{N}(\phi_1 x_{t-1} + \cdots + \phi_p x_{t-p}, \sigma^2), \quad t = 1, \ldots, n.$$

where we set $x_0 = x_{-1} = \cdots = x_{-p+1} = 0$ for simplicity. The precision matrix for an AR(p) process will then be a band matrix with bandwidth $b_w = p$. Using Corollary 1, it follows immediately that $L_{j,i} = 0$, $j > i$, $\forall\ j - i > p$, hence $\boldsymbol{L}$ is a band matrix with the same bandwidth."

Theorem 6.
If $\boldsymbol{Q}$ is a SPD band matrix with bandwidth b_w, then its Cholesky triangle $\boldsymbol{L}$ is a lower triangular band matrix with the same bandwidth.

"In this example, only $\mathcal{O}(n(b_w + 1))$ of the $\mathcal{O}(n^2)$ terms in$\boldsymbol{L}$ are nonzero, which is a significant reduction. A direct consequence is that the algorithm for computing the Cholesky factorization can be simplified; two of the three loops only need to go within the bandwidth, so the complexity is reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(nb_w^2)$. For fixed b_w, this gives a computational cost, which is linear in n."

Reordering Techniques: Band Matrices

"The great computational savings we obtained for band matrices naturally raise the question of who then we can use this approach also for "nonband", but sparse, matrices. A rationale for such an approach is that the indices in the graph are arbitrary, hence, we can permute the indices to obtain a small bandwidth, do the computations and perform the inverse permutation on the answer. Formally, let $\boldsymbol{P}$ by one of the $n!$, $n \times n$ permutation matrices; each row and column of $\boldsymbol{P}$ has one and only one nonzero entry, which is 1. The transpose of a permutation matrix is the inverse permutation, $\boldsymbol{P}^\top \boldsymbol{P} = \boldsymbol{I}$. For example, $\boldsymbol{Q}\boldsymbol{\mu} = \boldsymbol{b}$ can be solved as follows; multiply both sides with $\boldsymbol{P}$

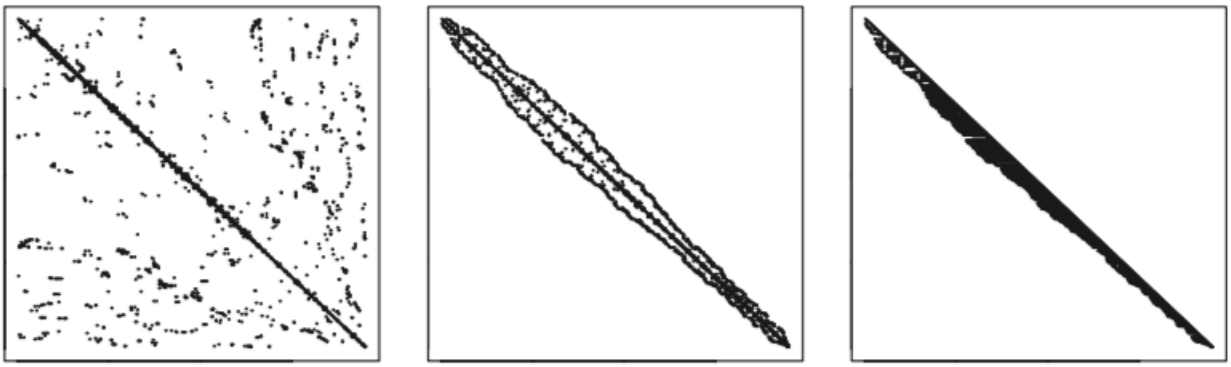


Figure 4: Panel (a) displays the nonzeros of the 380×380 precision matrix, panel (b) displays the reordered precision matrix with bandwidth 38, and panel (c) displays the Cholesky triangle of the band matrix (b).

$$\underbrace{(\boldsymbol{P}\boldsymbol{Q}\boldsymbol{P}^\top)}_{\widetilde{\boldsymbol{Q}}}\underbrace{\boldsymbol{P}\boldsymbol{\mu}}_{\widetilde{\boldsymbol{\mu}}} = \underbrace{\boldsymbol{P}\boldsymbol{b}}_{\widetilde{\boldsymbol{b}}}$$

solve $\widetilde{\boldsymbol{Q}}\widetilde{\boldsymbol{\mu}} = \widetilde{\boldsymbol{b}}$, and then apply the inverse permutation to obtain the solution $\boldsymbol{\mu} = \boldsymbol{P}^\top\widetilde{\boldsymbol{\mu}}$.

The next issue is how to permute the sparse matrix in order to obtain a (possible) small bandwidth. This issue is somewhat more technical, but there is a huge literature in computer science, and good working algorithms. So any algorithm that runs quick and gives reasonable results is fine. Figure 4 displays an example, where in panel (a) we display the precision matrix found from a spatial application

- (SECTION 4.2.2 OF GMRFs BOOK),

in (b) the reordered precision matrix using the Gibbs-Poole-Stockmeyer reordering algorithm

- (J.G. Lewis. Algorithm 582: The Gibbs-Poole-Stockmeyer and Gibbs-King algorithms for reordering sparse matrices. *ACM Transactions on Mathematical Software*, 8(2):190-194, June 1982.),

and in (c), the Cholesky triangle of the reordered precision matrix. The bandwidth after reordering is 38."

Reordering Techniques: General Sparse Matrices

"Although the band matrix approach gives very efficient algorithms for certain graphs, we often encounter situations where a more general approach is required. One such example is where the graph has some "global nodes"; nodes which are neighbors to (near) all other nodes. In statistical applications, such situations occur quite frequently, as shown in the following example."

Example 6.
"Let $\mu \sim \mathcal{N}(0,1)$ and $\{z_t\}$ be a AR(1) process of length T with mean μ; then $\boldsymbol{x} = (\boldsymbol{z}^\top, \mu)^\top$ is a GMRF *wrt* $\mathcal{G}$ where node μ is neighbor of all other nodes. The bandwidth is $n-1$ where $n = T+1$, for all $n!$ reorderings."

"The band matrix approach is not successful in this example, but we can derive efficient factorizations by making use of a general (and complex) factorization scheme. The general scheme computes only the nonzero terms in $\boldsymbol{L}$, which requires a substantial increase of complexity. The issue then is to reorder to minimize the number of terms in $\boldsymbol{L}$ not known to be zero. Define $M(\mathcal{G})$ as the number of not-zero terms in $\boldsymbol{L}$ found using Corollary 1.

Then the efficiency of any reordering is usually compared using the number of *fill-ins*

$$\text{fill} - \text{ins}(\mathcal{G}) = M(\mathcal{G}) - (|\mathcal{V}| + |\mathcal{E}|/2).$$

Since $L_{i,i} > 0$, $\forall i$,and $L_{j,i}$ is in general nonzero for $i \sim j$ and $j > i$, then fill-ins$(\mathcal{G}) \geq 0$.

Autoregressive processes of order p are optimal in the sense that the precision matrix is a band matrix with bandwidth p (and dense within the band), and with identity ordering, the number of fill-ins is zero (see Theorem 6). For other GMRFs, the number of fill-ins is (in most cases) nonzero and different reordering schemes can be compared to find a reasonable reordering. Note that there is no need to find *the optimal* reordering, but any reasonable one will suffice.

Let us reconsider Example 6 where we compare two reorderings where the global node μ is ordered first and last, respectively; $\boldsymbol{x} = (\boldsymbol{z}^\top, \mu)^\top$ and $\boldsymbol{x}' = (\mu, \boldsymbol{z}^\top)^\top$. The precision matrices are

$$\boldsymbol{Q} = \begin{pmatrix} \times & \times & & & & & \times \\ \times & \times & \times & & & & \times \\ & \times & \times & \times & & & \times \\ & & \times & \times & \times & & \times \\ & & & \times & \times & \times & \times \\ & & & & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \end{pmatrix} \quad \text{and} \quad \boldsymbol{Q}' = \begin{pmatrix} \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & & & & \\ \times & \times & \times & \times & & & \\ \times & & \times & \times & \times & & \\ \times & & & \times & \times & \times & \\ \times & & & & \times & \times & \times \\ \times & & & & & \times & \times \end{pmatrix},$$

respectively. Here, $\times$ indicates a nonzero value. Using Corollary 1, we obtain the (general) nonzero structure for the Cholesky triangles

$$\boldsymbol{L} = \begin{pmatrix} \times & & & & & & \\ \times & \times & & & & & \\ & \times & \times & & & & \\ & & \times & \times & & & \\ & & & \times & \times & & \\ & & & & \times & \times & \\ \times & \times & \times & \times & \times & \times & \times \end{pmatrix} \quad \text{and} \quad \boldsymbol{L}' = \begin{pmatrix} \times & & & & & & \\ \times & \times & & & & & \\ \times & \times & \times & & & & \\ \times & \checkmark & \times & \times & & & \\ \times & \checkmark & \checkmark & \times & \times & & \\ \times & \checkmark & \checkmark & \checkmark & \times & \times & \\ \times & \checkmark & \checkmark & \checkmark & \checkmark & \times & \times \end{pmatrix},$$

where a ✓indicates the fill-ins. Placing the global node μ last does not give any fill-ins, whereas placing it first gives a maximum number of fill-ins. This insight can be used to derive the reordering scheme called nested dissection, which goes as follows.

- Select a (small) set of nodes whose removal divides the graph into two disconnected subgraphs of almost equal size
- Order the nodes chosen *after* ordering all the nodes in both subgraphs
- Apply this procedure recursively to the nodes in eachs ubgraph

Formally, this can be described as follows:

Lemma 1.
Let $\boldsymbol{x}$ be a GMRF wrt to $\mathcal{G}$ and SPD precision matrix $\boldsymbol{Q}$, and partition $\boldsymbol{x}$ as $(\boldsymbol{x}_A^\top, \boldsymbol{x}_B^\top, \boldsymbol{x}_C^\top)^\top$. Partition the Cholesky triangle of $\boldsymbol{Q}$ as

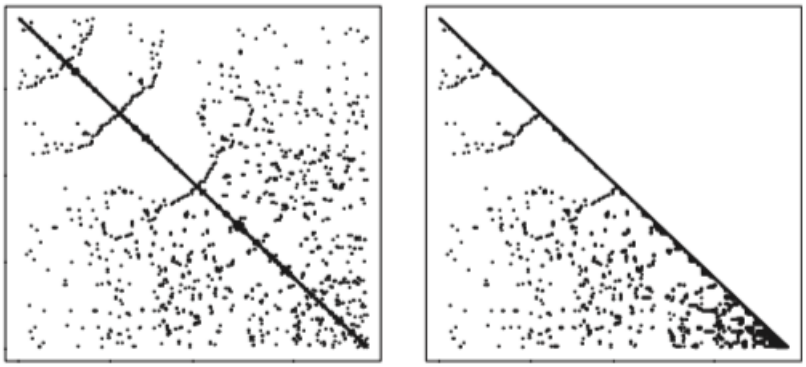
$$\boldsymbol{L} = \begin{pmatrix} \boldsymbol{L}_{AA} & & \\ \boldsymbol{L}_{BA} & \boldsymbol{L}_{BB} & \\ \boldsymbol{L}_{CA} & \boldsymbol{L}_{CB} & \boldsymbol{L}_{CC} \end{pmatrix}.$$

If C separates A and B in $\mathcal{G}$, then $\boldsymbol{L}_{BA} = \boldsymbol{0}$.

The recursive approach, proceeds by partitioning A and B similarly, and so on. It turns out that the nested dissection reordering gives optimal reorderings (in the order sense) for GMRFs found from discretizing the lattice or the cube: Consider a regular square lattice with n sites where each vertex is neighbor to the nearest four vertices. The nested dissection reordering will use C as the middle column, and A and B as the left and right part. Then this process is repeated recursively. It turns out that the cost of factorization of the reordered precision matrix will be $\mathcal{O}(n^{3/2})$, which is $\sqrt{n}$ times faster than using the band approach. The numbers of fill-ins will (only) be $\mathcal{O}(n \log n)$. It can be shown that factorization of the precision matrix applying any reordering is larger or equal to $\mathcal{O}(n^{3/2})$; hence optimal in the order sense. For a 3D box-lattice with n vertices, the computational cost is $\mathcal{O}(n^2)$ and the number of fill-ins is $\mathcal{O}(n^{4/3})$.

In between the band reordering for long and thin graphs and the nested dissection reordering for lattice-like graphs, there are several other reordering schemes than can provide, by a case to case basis, better reordering. Which one to try depends on which implementation one has available; however, any reasonable choice for the reordering will suffice. Note that the number of fill-ins for a specific graph can be computed (by sparse matrix libraries) without having to do the actual factorization, as it only depends on the graph; hence, if several factorizations should be performed on the same graph, it can be of benefit to compare (a few) reorderings and choose the one with fewest number of fill-ins.

We close this discussion by revisiting the 380×380 precision matrix displayed in Figure 4a. The band reordering gives 11112 fill-ins, the nested dissection gives 2460, while the "optimal" one produced 2182 fill-ins. The optimal reordered precision matrix and the corresponding Cholesky triangle are displayed in Figure 5."



(a) (b)

Figure 5: Panel (a) displays the "optimal" reordering with 2182 number of fill-ins, of the 380×380 precision matrix in Figure 4a. Panel (b) displays the corresponding Cholesky triangle.

Exact Calculations of Marginal Variances

"We will now turn to a more "statistical" issue; how to compute all (or nearly all) marginal variances of a GMRF. This is a different task than computing only one variance, say of x_i , which can simply be done by solving $\boldsymbol{Qv} = \mathbf{1}_i$, where $\mathbf{1}_i$ is one at position i and zero otherwise, and then $\text{Var}(x_i) = v_i$." "We will derive the recursions from a statistical point of view, starting with the case that $\boldsymbol{Q}$ is SPD with no additional linear constraints. The case with constraints will be discussed afterwards."

General Recursions

"The starting point is again that the solution of $\boldsymbol{L}^\top \boldsymbol{x} = \boldsymbol{z}$ provides a sample $\boldsymbol{x}$ with precision matrix $\boldsymbol{Q}$, which implies that

$$x_i = \frac{z_i}{L_{i,i}} - \frac{1}{L_{i,i}} \sum_{k=i+1}^{n} L_{k,i} x_k, \quad i = n, \ldots, 1.$$

Multiply both sides with x_j for $j \geq i$. Then the expected value reads

$$\Sigma_{i,j} = \frac{\delta_{i,j}}{L_{i,i}^2} - \frac{1}{L_{i,i}} \sum_{k=i+1}^{n} L_{k,i} \Sigma_{k,j}, \quad j \geq i, \quad i = n, \ldots, 1, \tag{13}$$

where $\delta_{i,j}$ is one if $i = j$ and zero otherwise. The sum in (13) only needs to be over all nonzero $L_{j,i}$s, or at least, all those ks so that i and $k > i$ are ***not*** separated by $F(i,k)$; see Corollary 1. To simplify notation, define this index set as

$$\mathcal{I}(i) = \{k > i : i \text{ and } k \text{ are not separated by } F(i,k)\}$$

and their "union",

$$\mathcal{I} = \{\{i,k\} : k > i, i \text{ and } k \text{ are not separated by } F(i,k)\}$$

for k, $i = 1, \ldots, n$. Note that the elements of $\mathcal{I}$ are sets; hence if $\{i, j\} \in \mathcal{I}$, then so does $\{j, i\}$. $\mathcal{I}$ represent all those indices in $\boldsymbol{L}$ that are not known upfront to be zero; hence, must be computed doing the Cholesky factorization. With this notation Equation (13) reads

$$\Sigma_{i,j} = \frac{\delta_{i,j}}{L_{i,i}^2} - \frac{1}{L_{i,i}} \sum_{k \in \mathcal{I}(i)} L_{k,i} \Sigma_{k,j}, \quad j \geq i, \quad i = n, \ldots, 1, \tag{14}$$

Looking more closely into these equations, it turns out that we compute all the $\boldsymbol{\Sigma}_{i,j}$s s explicitly if we apply Equation (14) in the correct order:

for $i = n, \ldots, 1$
  for $j = n, \ldots, i$
    Compute $\Sigma_{i,j}$ from Equation (14) (recalling that $\Sigma_{k,j} = \Sigma_{j,k}$).

Although this direct procedure computes all the marginal variances $\Sigma_{n,n}, \ldots, \Sigma_{1,1}$,, it is natural to ask if it is necessary to compute all the $\Sigma_{i,j}$s in order to obtain the marginal variances. Let $\mathcal{J}$ be a set of pairs of indices $\{i, j\}$, and *assume* we can compute $\Sigma_{i,j}$ from Equation (14) only for all $\{i, j\} \in \mathcal{J}$, and still obtain all the marginal variances. Then the set $\mathcal{J}$ must satisfy two requirements:

Requirement 1. $\mathcal{J}$ must contain $\{1, 1\}, \ldots, \{n, n\}$

Requirement 2. While computing $\Sigma_{i,j}$ from Equation (14), we need to have already computed all those $\Sigma_{k,j}$s that we need, *i.e.*,

$$\{i, j\} \in \mathcal{J} \quad \text{and} \quad k \in \mathcal{I}(i) \quad \Longrightarrow \quad \{k, j\} \in \mathcal{J}. \tag{15}$$

The rather surprising result is that $\mathcal{J} = \mathcal{I}$ satisfy these requirements; a result that only depends on $\mathcal{G}$ and not the numerical values in $\boldsymbol{Q}$. This result implies that we can compute all the marginal variances as follows:

for $i = n, \ldots, 1$
  for decreasing $j \in \mathcal{I}(i)$
    Compute $\Sigma_{i,j}$ from Equation (14),

where the j-loop visits all entries in $\mathcal{I}(i)$ in decreasing order."

Recursions for Band Matrices

"The simplification is perhaps most transparent when $\boldsymbol{Q}$ is a band matrix with bandwidth b_w, where we have previously shown that $\mathcal{I}(i) = \{i + 1, \ldots, \min(n, i + b_w)\}$; see Theorem 6. In this case, Requirement 2 reads (for an interior vertex and $j \geq i$),

$$0 \leq j - i \leq b_w \quad \text{and} \quad 0 < k - i \leq b_w \quad \Longrightarrow \quad -b_w \leq k - j \leq b_w,$$

which is trivially true. The algorithm then becomes

for $i = n, \ldots, 1$
  for $j = \min(i + b_w, n), \ldots, i$
    Compute $\Sigma_{i,j}$ from Equation (14).

Note that this algorithm is formally equivalent to Kalman recursions for smoothing. The computational cost for autoregressive models is $\mathcal{O}(n)$."

Correcting for Linear Constraints

"With additional linear constraints, the constrained precision matrix will be less sparse, so we need an approach to correct marginal variances for additional linear constraints. This is similar to Equation (9). Let $\widetilde{\boldsymbol{\Sigma}}$ be the covariance matrix with the additional k linear constraints $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{e}$, and $\boldsymbol{\Sigma}$ the covariance matrix without constraints. The two covariance matrices then relate as follows:

$$\widetilde{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma} - \boldsymbol{Q}\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^\top)^{-1}\boldsymbol{A}\boldsymbol{Q}^{-1}. \tag{16}$$

Let $\boldsymbol{W}$ be the $n \times k$ matrix solving $\boldsymbol{Q}\boldsymbol{W} = \boldsymbol{A}^\top$, $\boldsymbol{V}$ the $k \times k$ Cholesky triangle of $\boldsymbol{A}\boldsymbol{W}$, and $\boldsymbol{Y}$ the $k \times n$ matrix solving $\boldsymbol{V}\boldsymbol{Y} = \boldsymbol{W}^\top$, then the i, jth element of Equation (16) can be written as

$$\widetilde{\Sigma}_{i,j} = \Sigma_{i,j} - \sum_{t=1}^{k} Y_{t,i} Y_{t,j}. \tag{17}$$

All terms of $\boldsymbol{\Sigma}$ that we compute solving Equation (14) can now be corrected using Equation (17). The computational cost of this correction is dominated by computing $\boldsymbol{Y}$, which costs $\mathcal{O}(nk^2)$. Again, with not too many constraints, this correction will not require any additional computational burden."

Markov Random Fields

"We" "now leave the Gaussian case and discuss Markov random fields (MRFs) more generally. We will first study the case where each x_i is one of K different "colors" or states; *i.e.*, $x_i \in \mathcal{S}_i = \{0, 1, \ldots, K-1\}$, and $x \in \mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_n$. The case $K = 2$ is particularly important and corresponds to a binary MRF. The main result, Theorem 7, can then be generalized to nonfinite $\mathcal{S}$."

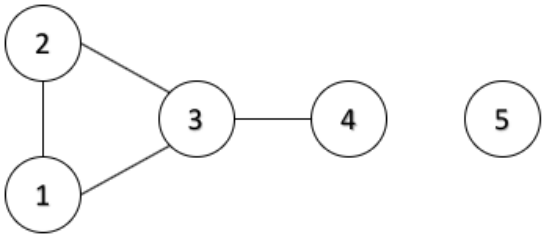


Figure 6: An example of a graph.

Background

"Recall the notion of the full conditionals for a joint distribution $\pi(\boldsymbol{x})$ that is the n conditional distributions $\pi(x_i \mid \boldsymbol{x}_{-i})$. From the full conditionals, we can define the notion of a neighbor."

Definition 3 (Neighbor).
Site $j \neq i$ is called a neighbor of site i if x_j contributes to the full conditional for x_i.

"Denote by ∂i, set all neighbors to site i, then

$$\pi(x_i \mid \boldsymbol{x}_{-i}) = \pi(x_i \mid \boldsymbol{x}_{\partial i}), \quad \forall i.$$

In a spatial context, it is easy to visualize this by, for example, considering ∂i as those sites that are (spatially) close to site i in some sense."

The Hammersley-Clifford Theorem

"Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the graph as defined through our specification of the neighbors to eachsite; $\mathcal{V} = \{1, \ldots, n\}$ and draw a directed edge from j to i if $j \in \partial i$ and $i \notin \partial j$. If i and j are mutually neighbors, draw an undirected edge. (In fact, it will turn out that if i is a neighbor of j, then also j must be a neighbor of i, although this is not known at the current stage.)"

Definition 4 (Markov random field).
If the full conditionals of $\pi(\boldsymbol{x})$, $\boldsymbol{x} \in \boldsymbol{S}$, honor a given graph $\mathcal{G}$, the distribution is called a Markov random field with respect to $\mathcal{G}$.

"For the main result, we also need the notion of a *clique*."

Definition 5 (Clique).
Any single site or any set of sites, all distinct pairs of which are mutual neighbors, is called a clique.

Example 7.
The cliques in the graph in Figure 6 are $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$, $\{5\}$, $\{1,2\}$, $\{1,3\}$, $\{2,3\}$, $\{1,2,3\}$, $\{3,4\}$.

"The main result is the Hammersley-Clifford theorem, which states what form the joint distribution must take to honor a given graph $\mathcal{G}$."

Theorem 7 (Hammersley-Clifford).
Let $\pi(\boldsymbol{x}) > 0$, $\boldsymbol{x} \in \boldsymbol{S}$ denote a Markov random field wrt a graph $\mathcal{G}$ with cliques $\mathcal{C}$, then

$$\pi(\boldsymbol{x}) \propto \prod_{C \in \mathcal{C}} \Psi_C(\boldsymbol{x}_C), \tag{18}$$

where the function Ψ_C can be chosen arbitrarily, subject to $0 < \Psi_C(\boldsymbol{x}_C) < \infty$.

"One *important* consequence of this result, is that for a given graph, the full conditionals should either be specified implicitly through the Ψ functions or verified that the chosen full conditionals can be derived from Equation (18) for some Ψ functions."

```
[1] "2018-08-09 17:01:27 +03"
```