Maximum likelihood estimation of Gaussian graphical models: Numerical implementation and topology selection

Joachim Dahl* Vwani Roychowdhury† Lieven Vandenberghe†

Abstract

We describe algorithms for maximum likelihood estimation of Gaussian graphical models with conditional independence constraints. It is well-known that this problem can be formulated as an unconstrained convex optimization problem, and that it has a closed-form solution if the underlying graph is chordal. The focus of this paper is on numerical algorithms for large problems with non-chordal graphs. We compare different gradient-based methods (coordinate descent, conjugate gradient, and limited-memory BFGS) and show how problem structure can be exploited in each of them. A key element contributing to the efficiency of the algorithms is the use of chordal embeddings for the fast computation of gradients of the log-likelihood function. We also present a dual method suited for graphs that are nearly chordal. In this method, results from matrix completion theory are applied to reduce the number of optimization variables to the number of edges added in the chordal embedding. The paper also makes several connections between sparse matrix algorithms and the theory of normal graphical models with chordal graphs. As an extension we discuss numerical methods for topology selection in Gaussian graphical models.

1 Introduction

We consider the problem of computing maximum likelihood (ML) estimates of the mean μ and covariance Σ of a multivariate normal variable $X \sim \mathcal{N}(\mu, \Sigma)$, subject to the constraint that certain given pairs of variables are conditionally independent. As we will explain in §2, the conditional independence constraints prescribe the sparsity pattern of the inverse of Σ and, as a consequence, the maximum likelihood estimation problem can be formulated as a convex optimization problem with Σ^{-1} as variable. The problem is also known as the *covariance selection* problem and was first studied in detail by Dempster [13]. A closely related problem is the maximum-determinant positive definite *matrix completion* problem [19].

In a graph representation of the random variable X, the nodes represent the comoponents X_i; two nodes are connected by an undirected edge if the corresponding variables are conditionally dependent. This is called a *normal* (or *Gaussian*) *graphical model* of the random variable [22, chapter 7]. For the special case of a chordal graph (*i.e.*, a graph in which every cycle of length greater than three has an edge connecting nonconsecutive nodes) the solution of the problem can be expressed in closed form in terms of the principal minors of the sample covariance matrix (see, for

*Corresponding author. Department of Electrical Engineering, University of California, Los Angeles. (joachim@ee.ucla.edu)

†Department of Electrical Engineering, University of California, Los Angeles. (vwani@ee.ucla.edu, vandenbe@ee.ucla.edu)

example, [32], [22, §5.3]). For non-chordal graphs the ML estimate has to be computed iteratively. Common algorithms that have been proposed for this purpose include Newton's method and the coordinate steepest descent algorithm [13, 30].

In this paper we present several large-scale algorithms that exploit convexity and sparsity in covariance selection problems with large non-chordal graphs. The main innovation that contributes to the efficiency of the algorithms is a fast technique for computing the gradient of the cost function via a chordal embedding of the graph. This is particularly effective in algorithms that require only first derivatives, such as steepest descent, conjugate gradient, and quasi-Newton methods.

We also present a dual method that exploits results from matrix completion theory. In this method the number of optimization variables in the dual problem is reduced to the number of added edges in a chordal embedding of the graph. The algorithm is therefore well suited for graphs that are nearly chordal, *i.e.*, graphs that can be embedded in a chordal graph by adding relatively few edges.

Large-scale algorithms for covariance selection have several important applications; see, for example, [4, 14]. One of these applications, which we investigate, is the topology or model selection problem, in wich we wish to identify the topology of the graph based on measured samples of the distribution.

The paper is organized as follows. In §2 we introduce the basic covariance selection problem, formulate it as a convex optimization problem, and derive the optimality conditions and the dual problem. In §3 we discuss the graph interpretation and describe solutions to different linear algebra problems related to chordal graphs. Section §3.2 discusses the Cholesky factorization of a positive definite matrix with chordal sparsity pattern. In §3.3 we present an efficient method for computing a partial inverse of a positive definite matrix with chordal sparsity pattern. In §3.4 we describe the well-known characterization of maximum-determinant positive definite matrix completions with chordal graphs.

In §4 we give expressions for the gradient and Hessian of the log-likelihood function, and we show that the gradient can be computed efficiently via a chordal embedding. Section §5 compares three gradient methods for the covariance selection problem: the coordinate steepest descent, conjugate gradient, and (limited memory) quasi-Newton methods. Section §6 contains another contribution of the paper, a dual algorithm suited for graphs that are almost chordal. In §7 we discuss the model selection problem. We present some conclusions in §8.

Notation

Let A be an $n \times n$ matrix and let $I = (i_1, i_2, \ldots, i_q) \in \{1, 2, \ldots, n\}^q$ and $J = (j_1, j_2, \ldots, j_q) \in \{1, 2, \ldots, n\}^q$ be two index lists of length q. We define A_{IJ} as the $q \times q$ matrix with entries $(A_{IJ})_{kl} = A_{i_k j_l}$. If I and J are unordered sets of indices, then A_{IJ} is the submatrix indexed by the elements of I and J taken in the natural order. The notation A_{IJ}^{-1} denotes the matrix $(A_{IJ})^{-1}$; it is important to distinguish this from $(A^{-1})_{IJ}$.

We use e_i to denote the ith unit vector, with dimension to be determined from the context. $A \circ B$ denotes the Hadamard (componentwise) product of the matrices A and B: $(A \circ B)_{ij} = A_{ij} B_{ij}$. For a symmetric matrix A, $A \succ 0$ means A is positive definite and $A \succeq 0$ means A is positive semidefinite. We use $\mathbf{S}^n$ to denote the set of symmetric $n \times n$ matrices, and $\mathbf{S}^n_+ = \{X \in \mathbf{S}^n \mid X \succeq 0\}$ and $\mathbf{S}^n_{++} = \{X \in \mathbf{S}^n \mid X \succ 0\}$ for the positive (semi)definite matrices.

2 Covariance selection

In this section we give a formal definition of the covariance selection problem and present two convex optimization formulations. In the first formulation (see §2.2), the log-likelihood function is maximized subject to sparsity constraints on the inverse of the covariance matrix. This problem is convex in the inverse covariance matrix. In the second formulation (§2.3), which is related to the first by duality, the covariance matrix is expressed as a maximum-determinant positive definite completion of the sample covariance.

2.1 Conditional independence in normal distributions

Let X, Y and Z be random variables with continuous distributions. We say that X and Y are *conditionally independent given* Z if

$$f(x|y,z) = f(x|z),$$

where $f(x|z)$ is the conditional density of X given Z, and $f(x|y,z)$ is the conditional density of X given Y and Z. Informally, this means that once we know Z, knowledge of Y gives no further information about X. Conditional independence is a fundamental property in expert systems and graphical models [22, 11, 26] and provides a simple factorization of the joint distribution $f(x,y,z)$: if X and Y are conditionally independent given Z then

$$f(x,y,z) = f(x|y,z)f(y|z)f(z) = f(x|z)f(y|z)f(z).$$

In this paper we are interested in the special case of conditional independence of two coefficients X_i, X_j of a vector random variable $X = (X_1, X_2, \ldots, X_n)$, given the other coefficients, *i.e.*, the condition that

$$f(x_i|x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = f(x_i|x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n).$$

If this holds, we simply say that X_i and X_j are conditionally independent.

There is a simple characterization of conditional independence for variables with a joint *normal* distribution. Suppose $I = (1, \ldots, k)$, $J = (k+1, \ldots, n)$. It is well-known that the conditional distribution of X_I given X_J is Gaussian, with covariance matrix

$$\Sigma_{II} - \Sigma_{IJ}\Sigma_{JJ}^{-1}\Sigma_{JI} = (\Sigma^{-1})_{II}^{-1} \tag{1}$$

(see, for example, [3, §2.5.1]). Applying this result to an index set $I = (i,j)$ with $i < j$, and $J = (1, 2, \ldots, i-1, i+1, \ldots, j-1, j+1, \ldots, n)$, we can say that X_i and X_j are conditionally independent if and only if the Schur complement (1), or, equivalently, its inverse, are diagonal. In other words, X_i and X_j are conditionally independent if and only if

$$(\Sigma^{-1})_{ij} = 0.$$

This classical result can be found in [13].

2.2 Maximum likelihood estimation

We now show that the ML estimation of the parameters μ and Σ of $\mathcal{N}(\mu, \Sigma)$, with constraints that given pairs of variables are conditionally independent, can be formulated as a convex optimization problem.

As we have seen, the constraints are equivalent to specifying the sparsity pattern of Σ^{-1}. Let S be the set of lower triangular positions of Σ^{-1} that are allowed to be nonzero, so the constraints are

$$(\Sigma^{-1})_{ij} = 0, \quad (i,j) \notin S. \tag{2}$$

Throughout the paper we assume that S contains all the diagonal entries. Let y_i, $i = 1, \ldots, N$, be independent samples of $\mathcal{N}(\mu, \Sigma)$. The log-likelihood function $L(\mu, \Sigma) = \log \prod_i f(y_i)$ of the observations is, up to a constant,

$$\begin{aligned} L(\mu, \Sigma) &= -\frac{N}{2} \log\det \Sigma - \frac{1}{2} \sum_{i=1}^{N} (y_i - \mu)^T \Sigma^{-1} (y_i - \mu) \\ &= \frac{N}{2} (-\log\det \Sigma - \mathbf{tr}(\Sigma^{-1}\bar{\Sigma}) - (\mu - \bar{\mu})^T \Sigma^{-1} (\mu - \bar{\mu})) \end{aligned} \tag{3}$$

where $\bar{\mu}$ and $\bar{\Sigma}$ are the sample mean and covariance

$$\bar{\mu} = \frac{1}{N} \sum_{i=1}^{N} y_i, \qquad \bar{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \bar{\mu})(y_i - \bar{\mu})^T.$$

The ML estimation problem with constraints (2) can therefore be expressed as

$$\begin{array}{ll} \text{maximize} & -\log\det \Sigma - \mathbf{tr}(\Sigma^{-1}\bar{\Sigma}) - (\mu - \bar{\mu})^T \Sigma^{-1} (\mu - \bar{\mu}) \\ \text{subject to} & (\Sigma^{-1})_{ij} = 0, \quad (i,j) \notin S, \end{array}$$

with domain $\{(\Sigma, \mu) \in \mathbf{S}^n \times \mathbf{R}^n \mid \Sigma \succ 0\}$. Clearly, the optimal value of μ is the sample mean $\bar{\mu}$, and if we eliminate the variable μ and make a change of variables $K = \Sigma^{-1}$, the problem reduces to

$$\begin{array}{ll} \text{maximize} & \log\det K - \mathbf{tr}(K\bar{\Sigma}) \\ \text{subject to} & K_{ij} = 0, \quad (i,j) \notin S. \end{array} \tag{4}$$

This is a convex optimization problem, since the objective function is concave on the set of positive definite matrices.

For sparse models, with few elements in S, it is often useful to express (4) as an unconstrained problem with the nonzero elements of K as variables. We therefore introduce the following notation. Suppose S has q elements $(i_1, j_1), \ldots, (i_q, j_q)$, and define two $n \times q$ matrices

$$E_1 = \begin{bmatrix} e_{i_1} & e_{i_2} & \cdots & e_{i_q} \end{bmatrix}, \qquad E_2 = \begin{bmatrix} e_{j_1} & e_{j_2} & \cdots & e_{j_q} \end{bmatrix}. \tag{5}$$

The elements of E_1 and E_2 are zero, except $(E_1)_{i_k,k} = 1$, $(E_2)_{j_k,k} = 1$, $k = 1, \ldots, q$. We can then parametrize K as

$$K(x) = E_1\, \mathbf{diag}(x) E_2^T + E_2\, \mathbf{diag}(x) E_1^T \tag{6}$$

where $x \in \mathbf{R}^q$ contains the nonzero elements in the strict lower triangular part of K, and the nonzero elements on the diagonal scaled by 1/2:

$$x_k = \begin{cases} K_{i_k,j_k} & i_k \neq j_k \\ (1/2) K_{i_k,i_k} & i_k = j_k, \end{cases} \qquad k = 1, \ldots, q.$$

With this notation, (4) is equivalent to the unconstrained convex optimization problem

$$\text{minimize} \quad -\log\det K(x) + \mathbf{tr}(K(x)\bar{\Sigma}) \tag{7}$$

with variable $x \in \mathbf{R}^q$.

2.3 Duality and optimality conditions

The Lagrange dual function of the problem (4) is

$$\begin{aligned} g(\nu) &= \inf_{K\succ 0}(\log\det K - \mathbf{tr}(K\bar{\Sigma}) - 2\sum_{(i,j)\notin S}\nu_{ij}K_{ij}) \\ &= -\log\det(\bar{\Sigma} + \sum_{(i,j)\notin S}\nu_{ij}(e_ie_j^T + e_je_i^T)) - n \end{aligned}$$

(see [8, chapter 5]). The variables ν_{ij} are the Lagrange multipliers for the equality constraints in (4). The dual problem is to minimize the dual function, *i.e.*,

$$\text{minimize} \quad -\log\det\left(\bar{\Sigma} + \textstyle\sum_{(i,j)\notin S}\nu_{ij}(e_ie_j^T + e_je_i^T)\right). \tag{8}$$

Equivalently, if we introduce a new variable Z for the argument of the objective, we obtain

$$\begin{array}{ll} \text{minimize} & -\log\det Z \\ \text{subject to} & Z_{ij} = \bar{\Sigma}_{ij}, \quad (i,j) \in S. \end{array} \tag{9}$$

In this problem we maximize the determinant of a positive definite matrix Z, subject to the constraint that Z agrees with the sample covariance matrix in the positions S. This is known as the *maximum determinant positive definite matrix completion* problem [19, 21].

It follows from convex duality theory that the optimal solutions K and Z in problems (4) and (9) are inverses, so the optimal Z is equal to the ML estimate of Σ. We conclude that the ML estimate of Σ is the maximum determinant positive definite completion of the sample covariance matrix $\bar{\Sigma}$, with free entries in the positions where Σ^{-1} is constrained to be zero. We can summarize this observation by stating the optimality conditions for the ML estimate Σ:

$$\Sigma \succ 0, \qquad \Sigma_{ij} = \bar{\Sigma}_{ij}, \quad (i,j) \in S, \qquad (\Sigma^{-1})_{ij} = 0, \quad (i,j) \notin S. \tag{10}$$

3 Graph interpretation and solution for chordal graphs

The sparsity pattern S defines an undirected graph $\mathcal{G} = (V, S_o)$ with vertices $V = \{1, 2, \ldots, n\}$ and edges $S_o = \{(i,j) \in S \mid i \neq j\}$. The edges define the free (nonzero) entries of Σ^{-1} in (10) and the fixed entries in Σ. We will assume without loss of generality that the graph $\mathcal{G}$ is connected; if it is not, the ML estimation problem can be decomposed into a number of independent problems on connected graphs.

In this section we discuss the covariance selection problem under the assumption that the graph $\mathcal{G}$ is *chordal* (as defined in §3). We present three related algorithms that exploit chordality.

- Zero fill-in Cholesky factorization of a sparse positive definite matrix with chordal sparsity pattern (§3.2).

- Computing the partial inverse of a sparse positive definite matrix with chordal sparsity pattern (§3.3). In the partial inverse, only the elements of the inverse in the positions of the nonzeros of the matrix are computed, but not the other elements in the inverse.

- Covariance selection with a chordal sparsity pattern and computation of the inverse covariance matrix (§3.4).

The first and third algorithms represent known results in linear algebra [7], the theory of graphical models [32, 22], and the literature on positive definite matrix completions [19]. We have not found a reference for the partial inverse algorithm, although the technique is related to the method of Erisman and Tinney [15].

3.1 Chordal graphs

An undirected graph $\mathcal{G}$ is called *chordal* if every cycle of length greater than three has a chord, *i.e.*, an edge joining nonconsecutive nodes of the cycle. In the graphical models literature the terms *triangulated graph* or *decomposable graph* are also used as synonyms for a chordal graph. Simple analytic formulas exist for the solution of the ML estimation problem (7) and its dual (9) in the special case when the graph $\mathcal{G} = (V, S_{\mathrm{o}})$ defined by S is chordal.

The easiest way to derive these formulas is in terms of *clique trees* (also called *junction trees*) associated with the graph $\mathcal{G}$. A clique is a maximal subset of $V = \{1, \ldots, n\}$ that defines a complete subgraph, *i.e.*, all pairs of nodes in the clique are connected by an edge. The cliques can be represented by an undirected graph that has the cliques as its nodes, and edges between any two cliques with a nonempty intersection. We call this graph the *clique graph* associated with $\mathcal{G}$. We can also assign to every edge (V_i, V_j) in the clique graph a weight equal to the number of nodes in the intersection $V_i \cap V_j$. A clique tree of a graph is a maximum weight spanning tree of its clique graph. Clique trees of chordal graphs can be efficiently computed by the *maximum cardinality search* algorithm [27, 28, 31].

For the rest of the section we assume that there are l cliques V_1, V_2, ..., V_l in $\mathcal{G}$, so that the set of nonzero entries is given by

$$\{(i,j) \mid (i,j) \in S \text{ or } (j,i) \in S\} = (V_1 \times V_1) \cup (V_2 \times V_2) \cup \cdots \cup (V_l \times V_l).$$

We assume a clique tree has been computed, and we number the cliques so that V_1 is the root of the tree and every parent in the tree has a lower number than its children. We define $S_1 = V_1$, $U_1 = \emptyset$ and, for $i = 2, \ldots, l$,

$$S_i = V_i \setminus (V_1 \cup V_2 \cup \cdots \cup V_{i-1}), \qquad U_i = V_i \cap (V_1 \cup V_2 \cup \cdots \cup V_{i-1}). \tag{11}$$

It can be shown that for a chordal graph

$$S_i = V_i \setminus V_k, \qquad U_i = V_i \cap V_k \tag{12}$$

where V_k is the parent of V_i in the clique tree. This important property is known as the *running intersection property* [7].

3.2 Cholesky factorization with chordal sparsity pattern

If $\mathcal{G}$ is chordal, then a clique tree of $\mathcal{G}$ defines a *perfect elimination order* for sparse positive definite matrices with sparsity pattern S, *i.e.*, an elimination order that produces triangular factors with zero fill-in. In this section we explain this for a factorization of the form $PXP^T = RR^T$ with P a permutation matrix and R upper triangular. This is equivalent to a standard Cholesky factorization $\tilde{P}X\tilde{P}^T = LL^T$ where L is lower triangular, and $\tilde{P}$ is the permutation matrix P with the order of its rows reversed.

Let $X \in \mathbf{S}^n_{++}$ have sparsity pattern S: $X_{ij} = X_{ji} = 0$ if $(i,j) \notin S$. Assume that the nodes in $\mathcal{G}$ are numbered so that

$$S_1 = \{1, \ldots, |S_1|\}, \qquad S_k = \left\{ \sum_{j=1}^{k-1} |S_j| + 1, \quad \sum_{j=1}^{k-1} |S_j| + 2, \quad \ldots, \quad \sum_{j=1}^{k} |S_j| \right\} \text{ for } k > 1. \tag{13}$$

(In general, this assumption requires a symmetric permutation of the rows and columns of X.) We will show that X can be factored as $X = RR^T$ where R is an upper triangular matrix with the same sparsity pattern as X, *i.e.*,

$$R_{ij} = 0, \qquad (j,i) \notin S. \tag{14}$$

The proof is by induction on the number of cliques. The result is obviously true if $l = 1$: If there is only one clique, then $\mathcal{G}$ is a complete graph and S contains all lower-triangular entries, so if we factor X as $X = RR^T$ then R satisfies (14). Next, suppose the result holds for all chordal sparsity patterns with $l - 1$ cliques. We partition X as

$$X = \begin{bmatrix} X_{WW} & X_{WS_l} \\ X_{S_lW} & X_{S_lS_l} \end{bmatrix},$$

with $W = \{1, \ldots, n\} \setminus S_l$, and examine the sparsity patterns of the different blocks in the factorization

$$X = \begin{bmatrix} I & X_{WS_l}X_{S_lS_l}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} X_{WW} - X_{WS_l}X_{S_lS_l}^{-1}X_{S_lW} & 0 \\ 0 & X_{S_lS_l} \end{bmatrix} \begin{bmatrix} I & 0 \\ X_{S_lS_l}^{-1}X_{S_lW} & I \end{bmatrix}.$$

The submatrix $X_{U_lS_l}$ of X_{WS_l} is dense, since $V_l = U_l \cup S_l$ is a clique. The submatrix $X_{W\setminus U_l, S_l}$ is zero: a nonzero entry (i,j) with $i \in W \setminus U_l$, $j \in S_l$ would mean that V_l is not the only clique that contains node j, which contradicts the definition of S_l in (11). The Schur complement

$$\tilde{X}_{WW} = X_{WW} - X_{WS_l}X_{S_lS_l}^{-1}X_{S_lW}$$

is therefore identical to X_{WW} except for the submatrix

$$\tilde{X}_{U_lU_l} = X_{U_lU_l} - X_{U_lS_l}X_{S_lS_l}^{-1}X_{S_lU_l}.$$

The first term $X_{U_lU_l}$ is dense, since U_l is a subset of the clique V_l, so $\tilde{X}_{WW}$ has the same sparsity pattern as X_{WW}.

The sparsity pattern of X_{WW} and $\tilde{X}_{WW}$ is represented by the graph $\mathcal{G}$ with the nodes in S_l removed. Now we use the running intersection property of chordal graphs (12): the fact that $U_l \subseteq V_k$, where the clique V_k is the parent of V_l in the clique tree, means that removing the nodes S_l reduces the number of cliques by one. The reduced graph is also chordal, and a clique tree of it

is obtained from the clique tree of $\mathcal{G}$ by deleting the clique V_l. By the induction assumption $\tilde{X}_{WW}$ can therefore be factored as

$$\tilde{X}_{WW} = R_{WW} R_{WW}^T$$

where R_{WW} is upper triangular with the same sparsity pattern as X_{WW}. The result is a factorization of X with zero fill-in:

$$X = \begin{bmatrix} R_{WW} & R_{WS_l} \\ 0 & R_{S_lS_l} \end{bmatrix} \begin{bmatrix} R_{WW}^T & 0 \\ R_{WS_l}^T & R_{S_lS_l}^T \end{bmatrix},$$

where $X_{S_lS_l} = R_{S_lS_l} R_{S_lS_l}^T$ is the Cholesky factorization of the (dense) matrix $X_{S_lS_l}$,

$$R_{U_lS_l} = X_{U_lS_l} X_{S_lS_l}^{-1} R_{S_lS_l} = X_{U_lS_l} R_{S_lS_l}^{-T}, \qquad R_{W\setminus U_l, S_l} = 0. \tag{15}$$

We summarize the ideas in the proof by outlining an algorithm for factoring X as

$$X = RDR^T, \tag{16}$$

where the matrix D is block-diagonal with l diagonal blocks $D_{S_kS_k}$, and the matrix R is unit upper triangular with zero off-diagonal elements, except for $R_{U_kS_k}$, $k = 1, \ldots, l$. The following algorithm overwrites X with the factorization data.

Cholesky factorization with chordal sparsity pattern

given a positive definite matrix X with chordal sparsity pattern.

1. Compute a clique tree with cliques $V_1, \ldots, V_l$ numbered so that V_k has a higher index than its parents. Compute the sets S_k, U_k defined in (12).
2. For $k = l, l-1, \ldots, 2$, compute

$$X_{U_kS_k} := X_{U_kS_k} X_{S_kS_k}^{-1}, \qquad X_{U_kU_k} := X_{U_kU_k} - X_{U_kS_k} X_{S_kS_k}^{-1} X_{U_kS_k}^T.$$

These steps do not alter the sparsity pattern of X but overwrite its nonzero elements with the elements of D and R_k. After completion of the algorithm, the nonzero elements of D are $D_{S_kS_k} = X_{S_kS_k}$, $k = 1, \ldots, l$. The nonzero elements of R are its diagonal and $R_{U_kS_k} = X_{U_kS_k}$ for $k = 1, \ldots, l$.

Example Figure 1 shows a clique tree for a chordal graph with 17 nodes, defined by the sparsity pattern in the lefthand plot of figure 2. From figure 1 one can verify the running intersection property. For example, for clique 6, we have

$$S_6 = V_6 \setminus \{V_1, V_2, V_3, V_4, V_5\} = \{v_7\}, \qquad U_6 = V_6 \cap \{V_1, V_2, V_3, V_4, V_5\} = \{v_8, v_{11}\}.$$

The running intersection property states that $U_6 \subseteq V_5$.

To obtain a perfect elimination order from the clique tree, we reorder the nodes according to (13), for example, as

$$v_9,\ v_8,\ v_4,\ v_1,\ v_{13},\ v_{15},\ v_5,\ v_{12},\ v_{11},\ v_7,\ v_{10},\ v_{14},\ v_2,\ v_{17},\ v_6,\ v_{16},\ v_3.$$

Applying this same permutation to the rows and columns of the sparsity pattern on the left in figure 2 results in the sparsity pattern on the right. It can be verified that any positive definite matrix X with this sparsity pattern can be factored as RR^T, where R is upper triangular and $R + R^T$ has the same sparsity pattern as X.

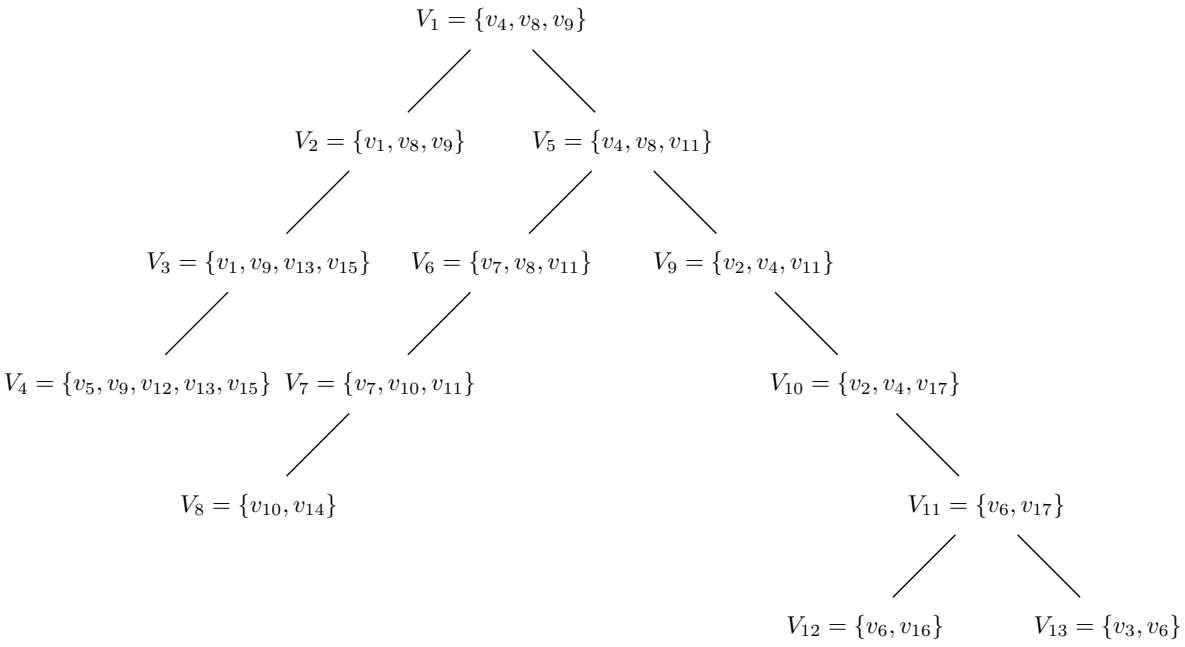

Figure 1: Clique tree of a chordal graph with 17 nodes, associated with the sparsity pattern of figure 2 (left).

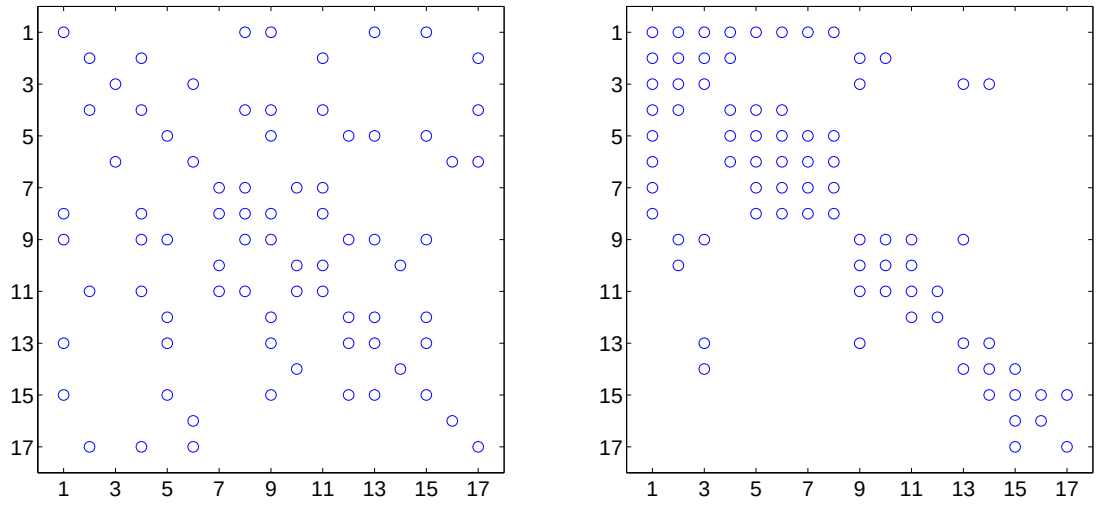

Figure 2: Left: sparsity pattern for a chordal graph. Right: sparsity pattern after a permutation using a perfect elimination ordering determined from the clique tree in figure 1.

3.3 Partial inverse of a positive definite matrix with chordal sparsity pattern

In this section we consider the problem of computing the elements $(X^{-1})_{ij}$, $(i,j) \in S$, where X is a positive definite matrix with sparsity pattern S. A straightforward solution to this problem consists in first computing the entire inverse X^{-1}, for example, from the Cholesky factorization $X = RR^T$, by solving the matrix equation

$$RR^T Y = I$$

in the unknown Y, and then selecting the specified entries of Y. This is inefficient for large sparse matrices because it computes all the entries of X^{-1}. In this section we will see that if the sparsity pattern S is chordal, then it is possible to efficiently compute the entries $(X^{-1})_{ij}$ for $(i,j) \in S$ directly, without calculating any other entries of X^{-1}.

The following algorithm returns a matrix $Y \in \mathbf{S}^n$ with $Y_{ij} = (X^{-1})_{ij}$ if $(i,j) \in S$ or $(j,i) \in S$, and $Y_{ij} = 0$ otherwise.

> Partial inverse of positive definite matrix with chordal sparsity pattern
>
> **given** a positive definite matrix X with chordal sparsity pattern.
>
> 1. Compute a clique tree with cliques $V_1, \ldots, V_l$ numbered so that V_k has a higher index than its parents. Compute the sets S_k, U_k defined in (12).
> 2. Compute the factorization $X = RDR^T$ by the algorithm in §3.2.
> 3. $Y := 0$. For $i = 1, \ldots, l$,
>
> $$Y_{U_i S_i} := -Y_{U_i U_i} R_{U_i S_i}, \qquad Y_{S_i U_i} := Y_{U_i S_i}^T, \qquad Y_{S_i S_i} := D_{S_i S_i}^{-1} - R_{U_i S_i}^T Y_{U_i S_i}.$$

To prove the correctness of the algorithm, let $Y^{(i)}$ be the value of Y after i cycles of the for-loop in step 3. We show that

$$Y^{(i)}_{V_k V_k} = (X^{-1})_{V_k V_k}, \quad k = 1, \ldots, i. \tag{17}$$

This implies that the final $Y = Y^{(l)}$ agrees with X^{-1} in the positions $(V_1 \times V_1) \cup (V_l \times V_l)$, *i.e.*, the nonzero positions of X.

Since $S_1 = V_1$ and $U_1 = \emptyset$, the matrix $Y^{(1)}$ is zero except for the submatrix

$$Y^{(1)}_{V_1 V_1} = Y^{(1)}_{S_1 S_1} = D^{-1}_{S_1 S_1} = (X^{-1})_{S_1 S_1} = (X^{-1})_{V_1 V_1}.$$

Therefore, (17) holds for $i = 1$. Next, assume that

$$Y^{(i-1)}_{V_k V_k} = (X^{-1})_{V_k V_k}, \quad k = 1, \ldots, i-1.$$

This immediately gives

$$Y^{(i)}_{U_i U_i} = Y^{(i-1)}_{U_i U_i} = (X^{-1})_{U_i U_i}, \tag{18}$$

because by the running intersection property $U_i \subseteq V_k$ for some $k < i$, and $Y_{U_i U_i}$ is not modified in interation i. To compute $(X^{-1})_{U_i S_i}$ and $(X^{-1})_{S_i S_i}$ we consider the matrix equation

$$R^T X^{-1} = D^{-1} R^{-1}. \tag{19}$$

We first examine the S_i, U_i block of this equation. The matrix R is unit upper triangular, with zero off-diagonal elements, except for the blocks $R_{U_k S_k}$, $k = 1, \ldots, l$. We have

$$(R^T X^{-1})_{S_i U_i} = (X^{-1})_{S_i U_i} + R^T_{U_i S_i} (X^{-1})_{U_i U_i}, \qquad (D^{-1} R^{-1})_{S_i U_i} = D^{-1}_{S_i S_i} (R^{-1})_{S_i U_i} = 0.$$

Solving for $(X^{-1})_{S_i U_i}$, we obtain

$$(X^{-1})_{S_i U_i} = -R^T_{U_i S_i} (X^{-1})_{U_i U_i}. \tag{20}$$

The two sides of the S_i, S_i block of the equation (19) are

$$(R^T X^{-1})_{S_i S_i} = R^T_{U_i S_i} (X^{-1})_{U_i S_i} + (X^{-1})_{S_i S_i}, \qquad (D^{-1} R^{-1})_{S_i S_i} = D^{-1}_{S_i S_i}.$$

Solving for $(X^{-1})_{S_i S_i}$ gives

$$(X^{-1})_{S_i S_i} = D^{-1}_{S_i S_i} - R^T_{U_i S_i} (X^{-1})_{U_i S_i}. \tag{21}$$

Combining (18), (20) and (21), we see that the ith cycle of the for-loop results in

$$Y^{(i)}_{U_i U_i} = (X^{-1})_{U_i U_i}, \qquad Y^{(i)}_{U_i S_i} = (X^{-1})_{U_i S_i}, \qquad Y^{(i)}_{S_i U_i} = (X^{-1})_{S_i U_i}, \qquad Y^{(i)}_{S_i S_i} = (X^{-1})_{S_i S_i},$$

and therefore $Y^{(i)}_{V_i V_i} = (X^{-1})_{V_i V_i}$. By induction this shows that (17) holds.

3.4 Maximum likelihood estimation in chordal graphical models

Recall from §2.3 that the ML estimate of the covariance matrix is given by the solution of the matrix completion problem (9). We now derive a solution for this problem assuming that the graph $\mathcal{G} = (V, S_o)$ is chordal and that the sample covariance $\bar{\Sigma}$ is positive definite. This result can be found, in different forms, in [19, 5], [22, page 146], [16, §2] [23], [32, §3.2] and [23]. We follow the derivation of [23].

We assume that the nodes of $\mathcal{G}$ are numbered as in §3.2 and show that the optimal solution can be expressed as

$$Z = L_l L_{l-1} \ldots L_2 D L_2^T \ldots L_{l-1}^T L_l^T \tag{22}$$

where D is block-diagonal with diagonal blocks

$$D_{S_k S_k} = \begin{cases} \bar{\Sigma}_{S_1 S_1} & k = 1 \\ \bar{\Sigma}_{S_k S_k} - \bar{\Sigma}_{S_k U_k} \bar{\Sigma}^{-1}_{U_k U_k} \bar{\Sigma}_{U_k S_k} & k = 2, \ldots, l. \end{cases} \tag{23}$$

The matrix L_k is unit lower triangular with zero off-diagonal elements except for the subblock

$$(L_k)_{S_k U_k} = \bar{\Sigma}_{S_k U_k} \bar{\Sigma}^{-1}_{U_k U_k}.$$

The proof of the result is by induction on the number of cliques.

The factorization is obviously correct if $l = 1$. In this case V is a clique, the ML estimate is simply $Z = \bar{\Sigma}$, and the expression (22) reduces to $Z = \bar{\Sigma}$.

Suppose the factorization (22) is correct for all sparsity patterns with $l - 1$ cliques. Partition Z as

$$Z = \begin{bmatrix} Z_{WW} & Z_{W S_l} \\ Z_{S_l W} & Z_{S_l S_l} \end{bmatrix}$$

where $W = V \setminus S_l$. The constraints in (9) fix certain entries in Z_{WW} and also imply that

$$Z_{S_l S_l} = \bar{\Sigma}_{S_l S_l}, \qquad Z_{U_l S_l} = \bar{\Sigma}_{U_l S_l}$$

because $V_l = S_l \cup U_l$ is a clique of $\mathcal{G}$. The entries in $Z_{W \setminus U_l, S_l}$ on the other hand are free, as a consequence of the definition (11). It can be verified that Z can be factored as

$$Z = L_l \tilde{Z} L_l^T = \begin{bmatrix} I & 0 \\ (L_l)_{S_l W} & I \end{bmatrix} \begin{bmatrix} Z_{WW} & \tilde{Z}_{W S_l} \\ \tilde{Z}_{S_l W} & \tilde{Z}_{S_l S_l} \end{bmatrix} \begin{bmatrix} I & (L_l^T)_{W S_l} \\ 0 & I \end{bmatrix}$$

where

$$(L_l)_{S_l U_l} = \bar{\Sigma}_{S_l U_l} \bar{\Sigma}_{U_l U_l}^{-1}, \qquad (L_l)_{S_l, W \setminus U_l} = 0, \tag{24}$$

and

$$\tilde{Z}_{U_l S_l} = 0, \qquad \tilde{Z}_{W \setminus U_l, S_l} = Z_{W \setminus U_l, S_l} - Z_{W \setminus U_l, U_l} \bar{\Sigma}_{U_l U_l}^{-1} \bar{\Sigma}_{U_l S_l}, \qquad \tilde{Z}_{S_l S_l} = \bar{\Sigma}_{S_l S_l} - \bar{\Sigma}_{S_l U_l} \bar{\Sigma}_{U_l U_l}^{-1} \bar{\Sigma}_{U_l S_l}.$$

This means that, for given Z_{WW}, the optimal (maximum-determinant) choice for $Z_{W \setminus U_l, S_l}$ is

$$Z_{W \setminus U_l, S_l} = Z_{W \setminus U_l, U_l} \bar{\Sigma}_{U_l U_l}^{-1} \bar{\Sigma}_{U_l S_l}.$$

If we make this choice, $\tilde{Z}$ reduces to

$$\tilde{Z} = \begin{bmatrix} Z_{WW} & 0 \\ 0 & \bar{\Sigma}_{S_l S_l} - \bar{\Sigma}_{S_l U_l} \bar{\Sigma}_{U_l U_l}^{-1} \bar{\Sigma}_{U_l S_l} \end{bmatrix}.$$

Other choices of $Z_{W \setminus U_l, S_l}$ change the 1,2 block in this matrix and therefore increase the determinant of $\tilde{Z}$, which is equal to the determinant of Z.

It remains to derive the optimal value of Z_{WW}. By the running intersection property, the subgraph of $\mathcal{G}$ corresponding to Z_{WW} has $l-1$ cliques, and a clique tree for it is the clique tree of $\mathcal{G}$ with the clique V_l removed. By the induction assumption the optimal Z_{WW} can therefore be expressed as

$$Z_{WW} = \tilde{L}_{l-1} \ldots \tilde{L}_2 \tilde{D} \tilde{L}_2^T \ldots \tilde{L}_{l-1}^T,$$

where $\tilde{L}_k$ is unit lower triangular of order $n - |S_l|$, with zero entries except for the subblock

$$(\tilde{L}_k)_{S_k U_k} = \bar{\Sigma}_{S_k U_k} \bar{\Sigma}_{U_k U_k}^{-1}.$$

The matrix $\tilde{D}$ is block diagonal with

$$\tilde{D}_{S_k S_k} = \begin{cases} \bar{\Sigma}_{S_1 S_1} & k = 1 \\ \bar{\Sigma}_{S_k S_k} - \bar{\Sigma}_{S_k U_k} \bar{\Sigma}_{U_k U_k}^{-1} \bar{\Sigma}_{U_k S_k} & k = 2, \ldots, l-1. \end{cases}$$

This means that if we define

$$D = \begin{bmatrix} \tilde{D} & 0 \\ 0 & \bar{\Sigma}_{S_l S_l} - \bar{\Sigma}_{S_l U_l} \bar{\Sigma}_{U_l U_l}^{-1} \bar{\Sigma}_{U_l S_l} \end{bmatrix}, \qquad L_k = \begin{bmatrix} \tilde{L}_k & 0 \\ 0 & I \end{bmatrix}, \quad k = 2, \ldots, l-1,$$

and L_l as in (24), we obtain the factorization (22).

As we have seen in §2.2 the inverse of the ML estimate Z is the solution of the primal problem (4). It follows that the optimal solution of (4) can be factored as

$$K = L_l^{-T} L_{l-1}^{-T} \ldots L_2^{-T} D^{-1} L_2^{-1} \ldots L_{l-1}^{-1} L_l^{-1}.$$

The following algorithm evaluates this product to compute K.

Inverse ML covariance matrix for a chordal sparsity pattern

given a sample covariance matrix $\bar{\Sigma}$ and a chordal sparsity pattern S.

1. Compute a clique tree with cliques $V_1, \ldots, V_l$ numbered so that V_k has a higher index than its parents. Compute the sets S_k, U_k defined in (12).
2. Compute the matrix D defined in (23). Set $K := D^{-1}$.
3. For $i = 2, \ldots, l$, compute

$$K_{S_iU_i} := -K_{S_iS_i}\bar{\Sigma}_{S_iU_i}\bar{\Sigma}_{U_i,U_i}^{-1}, \quad K_{U_iS_i} := K_{S_iU_i}^T, \quad K_{U_iU_i} := K_{U_iU_i} + K_{S_iU_i}^T K_{S_iS_i}^{-1} K_{S_iU_i}.$$

4 Gradient and Hessian of the log-likelihood function

In this section we derive expressions for the gradient and Hessian of the objective function of (7),

$$f(x) = -\log\det K(x) + \mathbf{tr}(K(x)\bar{\Sigma}),$$

with $K(x) = E_1\,\mathbf{diag}(x)E_2^T + E_2\,\mathbf{diag}(x)E_1^T$ as defined in (6). We also present an efficient method for evaluating the gradient via a chordal embedding of the sparsity pattern S of $K(x)$.

4.1 General expressions

The gradient and Hessian of f are easily derived from the second order approximation of the concave function $\log\det X$ at some $X \succ 0$:

$$\log\det(X + \Delta X) = \log\det(X) + \mathbf{tr}(X^{-1}\Delta X) - \frac{1}{2}\,\mathbf{tr}(X^{-1}\Delta X X^{-1}\Delta X) + \frac{1}{2}o(\|\Delta X\|^2).$$

Applying this with $X = K(x)$ and $\Delta X = K(\Delta x)$ gives the second order approximation of f:

$$f(x + \Delta x) = f(x) + \mathbf{tr}((\bar{\Sigma} - K(x)^{-1})K(\Delta x)) + \frac{1}{2}\,\mathbf{tr}(K(\Delta x)K(x)^{-1}K(\Delta x)K(x)^{-1}) + o(\|\Delta x\|^2). \quad (25)$$

To find $\nabla f(x)$ and $\nabla^2 f(x)$, we write the righthand side in the form

$$f(x) + \nabla f(x)^T\Delta x + \frac{1}{2}\Delta x^T\nabla^2 f(x)\Delta x + o(\|\Delta x\|^2)$$

using the definition $K(\Delta x) = E_1\,\mathbf{diag}(\Delta x)E_2^T + E_2\,\mathbf{diag}(\Delta x)E_1^T$. The second (linear) term is

$$\begin{aligned}\mathbf{tr}((\bar{\Sigma} - K(x)^{-1})K(\Delta x)) &= \mathbf{tr}\left((\bar{\Sigma} - K(x)^{-1})(E_1\,\mathbf{diag}(\Delta x)E_2^T + E_2\,\mathbf{diag}(\Delta x)E_1^T)\right)\\ &= 2\Delta x^T\,\mathbf{diag}\left(E_1^T(\bar{\Sigma} - K(x)^{-1})E_2\right),\end{aligned}$$

so the gradient of f is

$$\begin{aligned}\nabla f(x) &= 2\,\mathbf{diag}\left(E_1^T(\bar{\Sigma} - K(x)^{-1})E_2\right)\\ &= 2\,\mathbf{diag}(\bar{\Sigma}_{IJ} - (K(x)^{-1})_{IJ}).\end{aligned} \quad (26)$$

The third (quadratic) term in (25) is

$$\begin{aligned}\mathbf{tr}\left(K(\Delta x)K(x)^{-1}K(\Delta x)K(x)^{-1}\right) &= 2\,\mathbf{tr}\left(E_1\,\mathbf{diag}(\Delta x)E_2^TK(x)^{-1}E_1\,\mathbf{diag}(\Delta x)E_2^TK(x)^{-1}\right)\\ &\quad + 2\,\mathbf{tr}\left(E_1\,\mathbf{diag}(\Delta x)E_2^TK(x)^{-1}E_2\,\mathbf{diag}(\Delta x)E_1^TK(x)^{-1}\right)\end{aligned}$$

and can be simplified using the identity

$$\mathbf{tr}(A\,\mathbf{diag}(v)B\,\mathbf{diag}(v)) = \sum_{i,j} v_iA_{ij}B_{ji}v_j = v^T(A\circ B^T)v. \tag{27}$$

We find

$$\begin{aligned}\nabla^2 f(x) &= 2\left(E_1^TK(x)^{-1}E_1\right)\circ\left(E_2^TK(x)^{-1}E_2\right) + 2\left(E_1^TK(x)^{-1}E_2\right)\circ\left(E_2^TK(x)^{-1}E_1\right)\\ &= 2\left(K(x)^{-1}\right)_{II}\circ\left(K(x)^{-1}\right)_{JJ} + 2\left(K(x)^{-1}\right)_{IJ}\circ\left(K(x)^{-1}\right)_{JI}.\end{aligned} \tag{28}$$

Although from this expression it is not immediately clear that $\nabla^2 f(x)$ is positive definite when $K(x)\succ 0$, this is easily shown as follows. Let $X = K(x)$. We first note that the matrix

$$\begin{aligned}\begin{bmatrix} X_{II}\circ X_{JJ} & X_{IJ}\circ X_{JI}\\ X_{JI}\circ X_{IJ} & X_{JJ}\circ X_{II}\end{bmatrix} &= \begin{bmatrix} X_{II} & X_{IJ}\\ X_{JI} & X_{JJ}\end{bmatrix}\circ\begin{bmatrix} X_{JJ} & X_{JI}\\ X_{IJ} & X_{II}\end{bmatrix}\\ &= \left(\begin{bmatrix} E_1^T\\ E_2^T\end{bmatrix}X\begin{bmatrix} E_1 & E_2\end{bmatrix}\right)\circ\left(\begin{bmatrix} E_2^T\\ E_1^T\end{bmatrix}X\begin{bmatrix} E_2 & E_1\end{bmatrix}\right)\end{aligned} \tag{29}$$

is positive definite. Indeed, using the idenitity (27) we can write

$$v^T\begin{bmatrix} X_{II}\circ X_{JJ} & X_{IJ}\circ X_{JI}\\ X_{JI}\circ X_{IJ} & X_{JJ}\circ X_{II}\end{bmatrix}v = \mathbf{tr}\left(XWXW^T\right)$$

where

$$W = \begin{bmatrix} E_1 & E_2\end{bmatrix}\mathbf{diag}(v)\begin{bmatrix} E_2^T\\ E_1^T\end{bmatrix}.$$

Now since $X\succ 0$ and $WXW^T\neq 0$ if $v\neq 0$, we have $\mathbf{tr}(XWXW^T)>0$ for all $v\neq 0$. Therefore the matrix (29) is positive definite. From this it immediately follows that the matrix

$$\nabla^2 f(x) = 2\left(X_{II}\circ X_{JJ} + X_{IJ}\circ X_{JI}\right) = \begin{bmatrix} I\\ I\end{bmatrix}^T\begin{bmatrix} X_{II}\circ X_{JJ} & X_{IJ}\circ X_{JI}\\ X_{JI}\circ X_{IJ} & X_{JJ}\circ X_{II}\end{bmatrix}\begin{bmatrix} I\\ I\end{bmatrix}$$

is also positive definite.

4.2 Gradient via chordal embedding

The expression (26) shows that evaluating the gradient requires the partial inverse of $K(x)$, *i.e.*, the elements of $K(x)^{-1}$ in the positions of the nonzeros of $K(x)$:

$$\frac{\partial f(x)}{\partial x_k} = 2\bar{\Sigma}_{i_kj_k} - 2(K(x)^{-1})_{i_kj_k},\quad k=1,\ldots,q,$$

where $S=\{(i_1,j_1),(i_2,j_2),\ldots,(i_q,j_q)\}$ are the positions of the lower-triangular nonzero entries of $K(x)$. If S is chordal, the algorithm of §3.3 therefore provides a very efficient method for evaluating

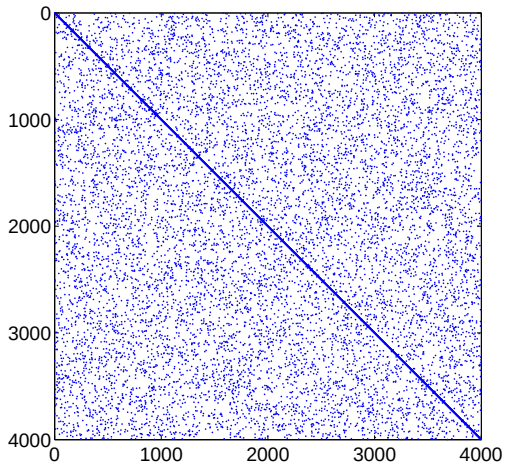

Figure 3: Sparsity pattern S of a sparse matrix with 14,938 non-zero elements.

the gradient. The algorithm is also useful when the sparsity pattern S is not chordal. In this case we first create a chordal sparsity pattern $\tilde{S}$ that contains S, and then apply the method of §3.3 to compute the elements of $(K(x)^{-1})_{ij}$ for all $(i,j) \in \tilde{S}$. If $\tilde{S}$ is not much larger than S, this method can be significantly faster than computing the entire inverse $K(x)^{-1}$.

A chordal sparsity pattern $\tilde{S}$ that contains S is known as a *chordal embedding* or *triangulation* of S. A good heuristic for computing a chordal embedding is to generate a fill-in reducing ordering of S (for example, an approximate minimum degree ordering [2]), followed by a symbolic Cholesky factorization. The sparsity pattern $\tilde{S}$ of the Cholesky factor defines a chordal embedding for S.

Example Figure 3 shows the sparsity pattern S of a matrix $X \in \mathbf{S}^{4000}_{++}$ with 14,938 non-zero elements. A symmetric minimum-degree reordering results in a a chordal embedding $\tilde{S}$ with 130,046 non-zero elements and 3650 cliques. Table 1 shows the distribution of the sizes of the clique subsets U_k and S_k (defined in (11)).

On a 2.8GHz Pentium IV PC with 2GB RAM it took approximately 12.7 seconds to compute the inverse of X using Matlab's sparse Cholesky factorization. It took only 0.32 seconds to compute only the elements $(X^{-1})_{ij}$ for $(i,j) \in S$, using the chordal embedding and the method of §3.3, implemented using BLAS and LAPACK.

5 Gradient methods for the primal problem

We now consider optimization methods for solving the ML problem

$$\text{minimize} \quad f(x) = -\log\det K(x) + \mathbf{tr}(K(x)\bar{\Sigma}). \tag{30}$$

This is an unconstrained convex minimization problem, and small and medium size problems are effectively solved via Newton's method. For larger problems, however, the cost of evaluating and factoring the Hessian (defined in (28)) becomes prohibitive, and gradient methods are better suited. As we have seen in the previous section, the gradient of the objective function can be efficiently evaluated, even when the matrix dimension n or the number of variables q is large, by exploiting

Range I	#cliques with $\lvert U_k\rvert \in I$	#cliques with $\lvert S_k\rvert \in I$
1–30	3599	3646
31–60	22	1
61–90	9	2
91–120	3	1
121–150	10	0
151–180	1	0
181–210	1	0
211–240	4	0
241–270	1	0

Table 1: Distribution of the sizes of the clique subsets U_k and S_k in a clique tree for the chordal embedding of (3). For each bin I, we show the number of cliques with $|U_k| \in I$ and the number of cliques with $|S_k| \in I$.

sparsity. In this section we discuss the implementations of three popular gradient methods and compare their performance.

5.1 Coordinate descent

In the coordinate descent algorithm we solve (30) one variable at a time. At each iteration, the gradient of f is computed, and the variable x_k with $k = \operatorname{argmax} |\partial f(x)/\partial x_k|$ is updated, by minimizing f over x_k while keeping the other variables fixed. This coordinate-wise minimization is repeated until convergence. The method is also known as steepest descent in ℓ_1-norm, and its convergence follows from standard results in unconstrained convex minimization [8, §9.4.2] [6, page 206]. Similar ideas were applied to covariance selection in the early work by Wermuth and Scheidt [33], and by Speed and Kiiveri [30].

Coordinate descent is a natural choice for the covariance selection problem, because each iteration is very cheap. We have already described in §4.2 an efficient method to evaluate the gradient $\nabla f(x)$, using a sparse Cholesky factorization of $K(x)$. In the rest of this section we discuss the minimization of f over one variable, and the update of the Cholesky factorization of $K(x)$ following a coordinate step.

Suppose we want to update x as $x := x + se_k$, where s minimizes

$$\begin{aligned} f(x+se_k) &= \mathbf{tr}(K(x+se_k)\bar{\Sigma}) - \log\det(K(x+se_k)) \\ &= \mathbf{tr}(K(x)\bar{\Sigma}) + 2s\bar{\Sigma}_{ij} - \log\det(K(x) + s(e_ie_j^T + e_je_i^T)) \end{aligned} \tag{31}$$

for $(i,j) = (i_k, j_k)$. To simplify the determinant we use the formula for the determinant of a 2 by 2 block matrix: If A and D are nonsingular, then

$$\det \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \det D \det(A - BD^{-1}C) = \det A \det(D - CA^{-1}B).$$

Therefore, with $\Sigma = K(x)^{-1}$,

$$\begin{aligned}
\det(K(x) + s(e_i e_j^T + e_j e_i^T)) &= \det \begin{bmatrix} K(x) & se_i & se_j \\ e_j^T & -1 & 0 \\ e_i^T & 0 & -1 \end{bmatrix} \\
&= \det K(x) \det \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + s \begin{bmatrix} e_j^T \\ e_i^T \end{bmatrix} \Sigma \begin{bmatrix} e_i & e_j \end{bmatrix} \right) \\
&= \det K(x) \det \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + s \begin{bmatrix} \Sigma_{ij} & \Sigma_{jj} \\ \Sigma_{ii} & \Sigma_{ij} \end{bmatrix} \right).
\end{aligned}$$

Next we note that

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + s \begin{bmatrix} \Sigma_{ij} & \Sigma_{jj} \\ \Sigma_{ii} & \Sigma_{ij} \end{bmatrix} = \begin{bmatrix} 0 & \Sigma_{ii}^{-1/2} \\ \Sigma_{jj}^{-1/2} & 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + t \begin{bmatrix} \rho & 1 \\ 1 & \rho \end{bmatrix} \right) \begin{bmatrix} 0 & \Sigma_{jj}^{1/2} \\ \Sigma_{ii}^{1/2} & 0 \end{bmatrix}$$

where $\rho = \Sigma_{ij}/(\Sigma_{ii}\Sigma_{jj})^{1/2}$ and $t = (\Sigma_{ii}\Sigma_{jj})^{1/2} s$, so

$$\begin{aligned}
\det \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + s \begin{bmatrix} \Sigma_{ij} & \Sigma_{jj} \\ \Sigma_{ii} & \Sigma_{ij} \end{bmatrix} \right) &= \det \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + t \begin{bmatrix} \rho & 1 \\ 1 & \rho \end{bmatrix} \right) \\
&= ((1 + t\rho)^2 - t^2).
\end{aligned}$$

If we also define $\bar{\rho} = \bar{\Sigma}_{ij}/(\Sigma_{ii}\Sigma_{jj})^{1/2}$, then we can write (31) as

$$\begin{aligned}
f(x + se_k) &= f(x) + 2t\bar{\rho} - \log((1 + t\rho)^2 - t^2) \\
&= f(x) + 2t\bar{\rho} - \log(1 + t(\rho + 1)) - \log(1 + t(\rho - 1)),
\end{aligned}$$

so it is clear that in order to minimize $f(x + se_k)$ over s, we need to minimize the function

$$g(t) = 2t\bar{\rho} - \log((1 + t\rho)^2 - t^2), \qquad \mathbf{dom}\, g = \begin{cases} (-\infty, 1/2) & \rho = -1 \\ (-(1+\rho)^{-1}, (1-\rho)^{-1}) & -1 < \rho < 1 \\ (-1/2, \infty) & \rho = 1. \end{cases}$$

If $\rho = -1$, then g is bounded below if and only if $\bar{\rho} < 0$, in which case the optimal solution is $t = (1 + \bar{\rho})/(2\bar{\rho})$. If $\rho = 1$, then g is bounded below if and only if $\bar{\rho} > 0$, in which case the optimal solution is $t = (1 - \bar{\rho})/(2\bar{\rho})$. If $-1 < \rho < 1$, then g is bounded below for all values of $\bar{\rho}$, and we can find the minimum by setting the derivative equal to zero:

$$\bar{\rho} = \frac{1}{t + (1 + \rho)^{-1}} + \frac{1}{t - (1 - \rho)^{-1}}.$$

This gives a quadratic equation in t,

$$\bar{\rho}(1 - \rho^2)t^2 - (1 - \rho^2 + 2\rho\bar{\rho})t + \rho - \bar{\rho} = 0,$$

with exactly one root in the interval $(-(1 + \rho)^{-1}, (1 - \rho)^{-1})$ (the unique root if $\bar{\rho} = 0$, the smallest root if $\bar{\rho} > 0$, and the largest root if $\bar{\rho} < 0$). Hence, we obtain a simple analytical expression for the optimal step size s that minimizes (31).

After calculating Δx_k, the Cholesky factorization of the matrix

$$K(x + \Delta x_k e_k) = K(x) + \Delta x_k (e_i e_j^T + e_j e_i^T), \tag{32}$$

can be updated very efficiently given the factorization of $K(x)$. We note that (32) may be written equivalently as

$$K(x + \Delta x_k e_k) = K(x) + uu^T - vv^T \tag{33}$$

where

$$u = \begin{cases} \sqrt{\Delta x_k/2}\,(e_i + e_j) & \Delta x_k \geq 0 \\ \sqrt{-\Delta x_k/2}\,(e_i - e_j) & \Delta x_k < 0, \end{cases} \qquad v = \begin{cases} \sqrt{\Delta x_k/2}\,(e_i - e_j) & \Delta x_k \geq 0 \\ \sqrt{-\Delta x_k/2}\,(e_i + e_j) & \Delta x_k < 0. \end{cases}$$

The expression (33) shows that we can update the factorization of K by making a symmetric rank-one update (if $v = 0$), or a symmetric rank-one downdate (if $u = 0$), or a rank-one update followed by a rank-one downdate. We can therefore use one of several updating and downdating methods available in the literature [18, page 611],[17].

5.2 Conjugate gradient method

The second algorithm in the comparison is the Fletcher-Reeves conjugate gradient algorithm (see [25, page 120]) with a backtracking line search using cubic interpolation [25, page 56].

We use a simple diagonal preconditioner and minimize $g(z) = f(\mathbf{diag}(H)^{1/2}z)$ instead of f, where

$$H = 2\left(\bar{\Sigma}_{II} \circ \bar{\Sigma}_{JJ} + \bar{\Sigma}_{IJ} \circ \bar{\Sigma}_{JI}\right) \tag{34}$$

and $\bar{\Sigma}$ is the sample covariance. To justify this choice, we first note that if we knew the optimal $x^\star$, then an ideal preconditioner would be to minimize $f(Uz)$ where $U = \nabla^2 f(x^\star)^{-1/2}$. The expression (28) shows that computing the Hessian $\nabla^2 f(x^\star)$ requires knowledge of $K(x^\star)^{-1}$, and from the optimality conditions (10) we note that $(K(x^\star)^{-1})_{ij} = \bar{\Sigma}_{ij}$ for $(i,j) \notin S$. So while we do not know $x^\star$ or $K(x^\star)$, we do know some entries of $K(x^\star)^{-1}$. A reasonable and inexpensive estimate of $\nabla^2 f(x^\star)$ is therefore to replace K^{-1} in the expression (28) with the sample covariance $\bar{\Sigma}$. This justifies using H instead of $\nabla^2 f(x^\star)$. Finally, since factoring H is too expensive, we do not use $H^{1/2}$ but $\mathbf{diag}(H)^{1/2}$.

5.3 Limited-memory BFGS method

The third method is the limited-memory Broyden-Fletcher-Goldfarb-Reeves (LBFGS) quasi-Newton method of [25, page 226]. Quasi-Newton methods are similar to Newton's method but use an approximation of the Hessian (or inverse Hessian) formed based on gradient evaluations. In the standard BFGS method an $n \times n$ dense matrix (or a triangular factor) is propagated as an approximate inverse Hessian. In the limited-memory BFGS (LBFGS) with limit m only the most recent m gradient evaluations are used. If m is much smaller than the number of variables, the LBFGS method is less expensive and requires less memory than the full BFGS method.

The BFGS and LBFGS methods require an initial approximation of the inverse Hessian. We experimented with two choices: the identity matrix and $\mathbf{diag}(H)^{-1}$ where H is defined in (34).

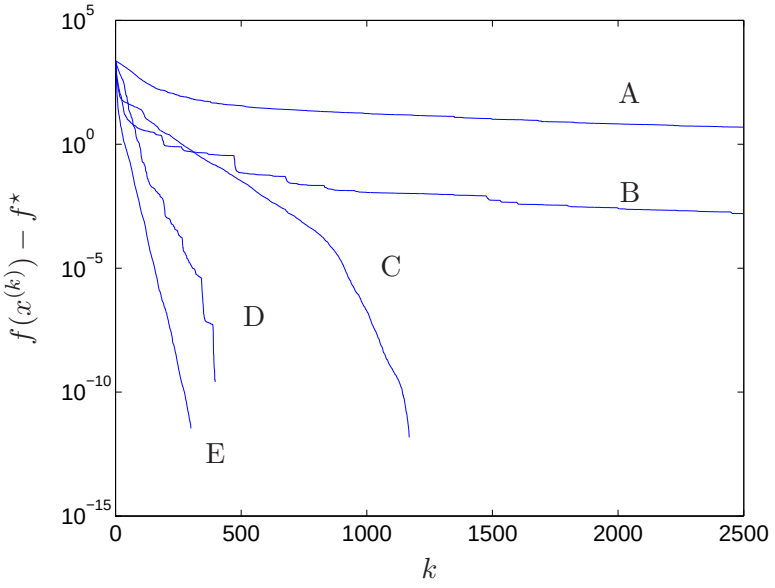

Figure 4: Convergence of five gradient methods on an example problem with $n = 200$ and $q = 1134$. A: coordinate steepest descent, B: conjugate gradient without preconditioner, C: conjugate gradient with diagonal preconditioner, D: BFGS with an identity matrix for the initial Hessian estimate, E: BFGS with diagonal initial Hessian estimate.

5.4 Numerical results

In this section we compare the different methods on a collection of randomly generated problems of varying dimensions and difficulty (measured by condition number of the Hessian at optimality).

In the first experiment we randomly generated a sparse 200×200 matrix K^* and constructed a problem (30) with K^* as solution by taking $\bar{\Sigma}_{ij} = ((K^*)^{-1})_{ij}$ for $(i, j) \in S$. The number of variables (*i.e.*, number of non-zero lower triangular elements in the inverse covariance) was $q = 1134$ and the condition number of the Hessian $\nabla^2 f(x^\star)$ at optimality was approximately $2 \cdot 10^5$. Figure 4 shows the convergence of five gradient methods: coordinate descent, conjugate gradient with and without preconditioner, and the (full) BFGS method. For comparison, Newton's method solved this problem in 11 iterations, but has a much higher cost per iteration and is significantly slower than the pre-started BFGS method.

Table 2 shows the number of iterations required to reach an accuracy $\|\nabla f(x_k)\| < 10^{-5}$ for the conjugate gradient and LBFGS methods with different values of m. We compare the conjugate gradient method without a preconditioner (CG), conjugate gradient with the diagonal preconditioner based on (34) (P-CG), the BFGS method (BFGS), limited memory BFGS with different limits (LBFGS), and finally limited memory BFGS with the initial Hessian estimate described in §5.3. The table clearly shows the advantage of using a preconditioner. It also shows that the quasi-Newton methods perform better than the conjugate gradient methods. In particular, the prestarted limited memory BFGS performs well over a wide range of problems using only a modest amount of memory.

The last example compares BFGS and LBFGS with $m = 5, 20, 100$ for an example with $n = 100$

	$n = 100$, $q = 425$			$n = 100$, $q = 425$			$n = 100$, $q = 425$		
Cond. number ρ	8E2	6E2	7E2	2E4	7E4	1E4	1E5	2E5	2E5
CG	261	255	286	1181	2313	1342	> 3000	> 3000	> 3000
P-CG	290	120	170	231	309	445	690	1507	1658
BFGS	105	116	116	481	823	473	619	NP	NP
LBFGS $m = 10$	180	147	204	1368	1947	1235	> 3000	> 3000	> 3000
LBFGS $m = 50$	134	140	152	973	1397	921	2200	> 3000	> 3000
LBFGS $m = 100$	106	110	118	803	1218	802	2089	> 3000	> 3000
P-LBFGS $m = 10$	79	76	82	188	275	417	1369	887	1055
P-LBFGS $m = 50$	57	68	64	218	285	369	> 3000	486	705
P-LBFGS $m = 100$	57	67	63	196	420	486	NP	420	486

	$n = 200$, $q = 1100$			$n = 200$, $q = 1100$			$n = 200$, $q = 1100$		
Cond. number ρ	2E2	2E2	3E2	1E4	9E4	2E4	2E5	9E5	1E5
CG	161	132	142	1070	2839	1291	> 3000	> 3000	> 3000
P-CG	98	63	282	273	409	1182	1367	> 3000	1495
BFGS	102	88	108	1907	911	1493	2090	NP	1316
LBFGS $m = 10$	127	107	115	972	1541	1154	> 3000	> 3000	2678
LBFGS $m = 50$	107	92	112	1614	1075	926	> 3000	> 3000	2660
LBFGS $m = 100$	102	88	108	1777	1050	972	> 3000	> 3000	2199
P-LBFGS $m = 10$	74	47	65	264	319	414	1043	658	1015
P-LBFGS $m = 50$	60	44	58	219	297	349	730	660	783
P-LBFGS $m = 100$	60	44	58	185	265	289	656	723	748

Table 2: Number of iterations for a collection of randomly generated problems. Each column shows the average number of iterations for three randomly generated instances with q variables (or lower-triangular nonzeros). The generated problems have different condition number of the Hessian at optimality, shown in each column. Runs marked 'NP' encountered numerical problems.

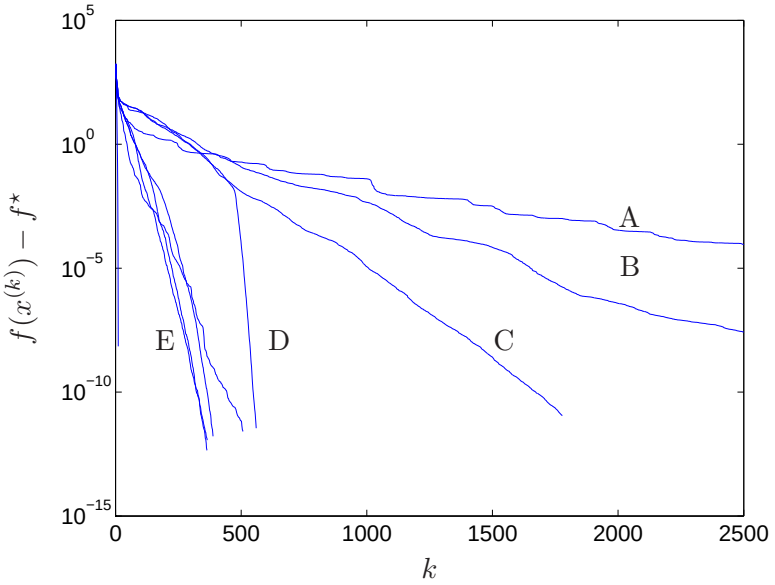

Figure 5: Convergence plots for A: LBFGS $m = 5$, B: LBFGS $m = 20$, C: LBFGS $m = 100$, D: BFGS, E: BFGS and LBFGS $m = 5, 20, 100$ and diagonal initialization.

and $q = 478$. The results are shown in figure 5. With an identity matrix as starting value we notice a large difference for different values of m; in fact, we only observe superlinear convergence for the full BFGS method. If we use the diagonal starting value for H as explained in §5.3 there is no significant difference between the different values of m in the range 5–100.

6 Gradient methods for the dual problem

In this section we discuss the possibility of solving the dual problem (9) by a gradient method. The dual problem can be written as an unconstrained problem with $n(n+1)/2 - q$ variables where $q = |S|$ (see (8)). In practice, this is a very large number. However, we can use matrix completion theory to substantially reduce the size of the problem.

Let $\tilde{S}$ be a chordal embedding of the sparsity pattern S, and let $q_e = |\tilde{S} \setminus S|$ be the number of lower-triangular positions added in the embedding. We denote by $\tilde{S}^c$ the complement of $\tilde{S}$,

$$\tilde{S}^c = \{(i,j) \mid i \geq j,\ (i,j) \notin \tilde{S}\}.$$

We will index the entries in $\tilde{S} \setminus S$ as (r_k, s_k) and the entries in in $\tilde{S}^c$ as $(\tilde{r}_k, \tilde{s}_k)$, so that

$$\tilde{S} \setminus S = \{(r_k, s_k) \mid k = 1, \ldots, q_e\}, \qquad \tilde{S}^c = \{(\tilde{r}_k, \tilde{s}_k) \mid k = 1, \ldots, n(n+1)/2 - (q + q_e)\}.$$

In this notation (9) reduces to the problem of minimizing the function

$$g(y,z) = -\log\det Z(y,z)$$

where $Z(y,z)$ is the symmetric matrix with lower-triangular elements

$$Z(y,z)_{ij} = \begin{cases} \bar{\Sigma}_{ij} & (i,j) \in S \\ y_k & (i,j) = (r_k, s_k), \quad k = 1, \ldots, q_{\mathrm{e}} \\ z_k & (i,j) = (\tilde{r}_k, \tilde{s}_k), \quad k = 1, \ldots, n(n+1)/2 - q - q_{\mathrm{e}}. \end{cases}$$

The key idea of the reformulation is as follows. For fixed y, we can compute the optimal z by solving a maximum-determinant matrix completion problem with a chordal sparsity pattern $\tilde{S}$. This means that the convex function

$$h(y) = \inf_z g(y,z),$$

can be evaluated by solving the matrix completion problem

$$\begin{array}{ll} \text{minimize} & -\log\det Z \\ \text{subject to} & Z_{ij} = \bar{\Sigma}_{ij}, \quad (i,j) \in S \\ & Z_{r_k s_k} = y_k, \quad k = 1, \ldots, q_{\mathrm{e}}. \end{array} \tag{35}$$

We can solve this problem by computing the factorization (22) and taking $h(y) = -\log\det D$. The same factorization provides the inverse of $Z(y, z(y))$, and thus the gradient of h, which is given by

$$\nabla h(y) = \nabla_y g(y, z(y))$$

where $z(y) = \mathrm{argmin}_z\, g(y,z)$. The gradient of g is

$$\frac{\partial g(y,z)}{\partial y_k} = -2\left(Z(y,z)^{-1}\right)_{r_k s_k}$$

(see §4), so evaluating ∇h requires computing the entries (r_k, s_k) in $Z(y, z(y))^{-1}$.

Example We illustrate the method by a large scale problem. We use the first 40,000 nodes of a dataset from the WebBase project [12]. The dataset consists of a directed graph where the nodes represent webpages and the edges represent links between webpages. We removed orientation of the edges, *i.e.*, we interpreted the graph as undirected. This resulted in a large sparse problem with $n = 40,000$, $q = 84,771$ and $q_e = 3510$.

We randomly generated a sparse sample covariance matrix on the chordal sparsity pattern. The generated sparse covariance matrix was positive definite on the sparsity pattern, as opposed to just having a positive definite completion. We solved this problem instance using a limited-memory BFGS method storing the past $m = 50$ search directions. On a 2.8 GHz Pentium IV PC the covariance selection problem was solved in 81 L-BFGS iterations, taking a total of 4976 seconds with the inverse Hessian estimate preset to the identity.

7 Topology selection and MAP estimation

7.1 Akaike and Bayes information criterion

In the topology selection problem we are given several possible sparsity patterns S_k, $k = 1, \ldots, K$, and wish to select the 'best' pattern and the corresponding ML estimates for Σ. This problem can

be addressed with standard techniques for model selection. The most common methods are the Akaike information criterion and the Jeffrey-Schwarz criterion or Bayesian information criterion [1, 29, 9, 10]. We explain the idea for zero-mean distributions $\mathcal{N}(0, \Sigma)$.

Let $\Sigma_{\mathrm{ml},k}$, $k = 1, \ldots, K$, be the ML covariance estimate for the sparsity pattern S_k, *i.e.*, the solution of the problem

$$\begin{array}{ll} \text{maximize} & L(\Sigma) = (N/2)\left(\log\det\Sigma^{-1} - \mathbf{tr}(\Sigma^{-1}\bar{\Sigma})\right) \\ \text{subject to} & (\Sigma^{-1})_{ij} = 0, \quad (i,j) \in S_k. \end{array}$$

(L is the log-likelihood function (3) for a zero-mean distribution.) The Akaike information criterion (AIC) selects the model with the largest value of

$$L(\Sigma_{\mathrm{ml},k}) - q_k, \tag{36}$$

where q_k is the number of nonzero entries in the lower-triangular part of Σ^{-1}. It is also the number of variables in the unconstrained formulation of the ML estimation problem (7). It can be shown that for large sample sizes N the quantity $-L(\Sigma_{\mathrm{ml},k})+q_k$ converges to the Kullback-Leibler distance between the true and the estimated distribution [9, 10].

For small sample sizes the AIC tends to overestimate q_k, and it is preferable to use the second order bias-corrected expression

$$L(\Sigma_{\mathrm{ml},k}) - q_k - \frac{q_k(q_k+1)}{N - q_k - 1} \tag{37}$$

instead of the quantity (36) [9]. The Bayesian information criterion (BIC) selects the model with the largest value of

$$L(\Sigma_{\mathrm{ml},k}) - \frac{\log N}{2} q_k. \tag{38}$$

Thus, in the AIC and BIC the log-likelihood function is augmented with a penalty term that depends on the number of parameters q_k. For the basic AIC and the BIC the penalty is proportional to the number of parameters, and the two criteria differ only in the constant of proportionality.

When the number of possible sparsity patterns K is not too large, the AIC- and BIC-optimal models can be computed by solving K ML estimation problems of the form considered in §5–§6.

7.2 Maximum a posteriori probability estimation

A related problem is maximum a posteriori probability (MAP) estimation of the distribution, *i.e.*, the problem

$$\begin{array}{ll} \text{maximize} & L(\Sigma, \mu) + \log(p(\Sigma, \mu)) \\ \text{subject to} & (\Sigma^{-1})_{ij} = 0, \quad (i,j) \in S, \end{array} \tag{39}$$

where $p(\Sigma, \mu)$ is the prior distribution of the parameters μ, Σ. The additional term in the objective can also be interpreted as a regularization term. Interesting choices for p are densities that result in sparse graph topologies (*i.e.*, covariance estimates with many zero elements in Σ^{-1}), while preserving convexity of the optimization problem (39). An example of such a distribution is an exponential distribution on the nonzero entries of Σ^{-1}. We give the details for zero-mean normal models $\mathcal{N}(0, \Sigma)$.

An exponential prior distribution on the nonzero elements of Σ^{-1} results in a penalty term $\sum_{k=1}^{q} |(\Sigma^{-1})_{i_k j_k}|$. In the simpler notation of problem (7) we obtain the regularized problem

$$\text{minimize} \quad -\log\det K(x) + \mathbf{tr}(K(x)\bar{\Sigma}) + \rho \sum_{k=1}^{q} |x_k| \tag{40}$$

where $\rho > 0$. Similar ℓ_1-regularized ML estimation problems are considered in [4] which includes additional bounds on the condition number of the solution, and in [20] where the goal of the regularization term is to trade variance for bias in the estimates.

Equivalently, one obtains the trade-off curve between $-\log\det K(x) + \mathbf{tr}(K(x)\bar{\Sigma})$ and $\sum_k |x_k|$ by solving

$$\begin{array}{ll} \text{minimize} & -\log\det K(x) + \mathbf{tr}(K(x)\bar{\Sigma}) \\ \text{subject to} & \sum_{k=1}^{q} |x_k| \leq \gamma \end{array} \tag{41}$$

for different values of γ.

The regularized ML problems (40) and (41) are convex optimization problem that can be solved efficiently using interior-point methods [24, 8], in combination with the large-scale numerical techniques discussed earlier in the paper. For example, we can write (40) as a constrained optimization problem with differentiable objective and constraint functions:

$$\begin{array}{ll} \text{minimize} & -\log\det K(x) + \mathbf{tr}(K(x)\bar{\Sigma}) + \gamma \mathbf{1}^T y \\ \text{subject to} & -y \preceq x \preceq y, \end{array}$$

with $y \in \mathbf{R}^q$ an auxiliary variable. A barrier method solves this problem by repeatedly minimizing the function

$$\phi_t(x, y) = t\left(-\log\det K(x) + \mathbf{tr}(K(x)\bar{\Sigma}) + \gamma \mathbf{1}^T y\right) - \sum_{k=1}^{q} \log(y_k^2 - x_k^2)$$

for a sequence of increasing values of t (see [8, chapter 11] for details). These unconstrained minimization problems can be solved by any of the methods discussed in §5–§6.

7.3 Examples

In the first experiment we take $N = 50$ samples of the zero-mean normal distribution $\mathcal{N}(0, \hat{\Sigma})$ with inverse covariance (or concentration) matrix

$$\hat{\Sigma}^{-1} = \begin{bmatrix} 1 & -1/2 & 0 & 1/3 & 0 \\ -1/2 & 1 & 1/2 & 0 & 0 \\ 0 & 1/2 & 1 & 1/3 & 0 \\ 1/3 & 0 & 1/3 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

For a model size $n = 5$ we can easily enumerate all $2^{n(n-1)/2} = 1024$ possible sparsity patterns or graph topologies S_k and compute a ML estimate $\Sigma_{\mathrm{ml},k}$ for each of them. The top curve in figure 6 shows, for each q, the log-likelihood value of the best scoring topology over all sparsity patterns with q nonzero lower triangular entries, *i.e.*,

$$\max_{k:q_k=q} L(\Sigma_{\mathrm{ml},k}).$$

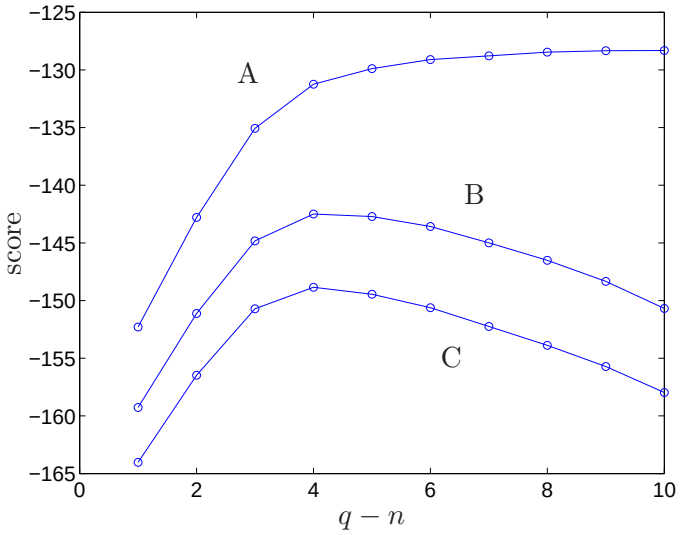

Figure 6: Highest scoring log-likelihood values (A) and corresponding AIC (B) and BIC scores (C) as a function of the number of nonzero entries in the strict lower triangular part of Σ^{-1} (*i.e.*, as a function of the number of edges in the estimated normal graph). The AIC and BIC have a maximum at four, which is the correct number of edges in the graph. They also select the correct topology.

The other curves show the corresponding AIC and BIC scores (37) and (38). In this example, and in most other instances of the same problem, the AIC and BIC identify the correct topology (*i.e.*, the sparsity pattern of $\hat{\Sigma}^{-1}$). The log-likelihood curve on the other hand increases monotonically with q, without distinct breakpoint.

In the second example we consider a larger graph with $n = 20$ nodes, which makes an enumeration of all possible graph topologies infeasible. We randomly construct a sparse inverse covariance matrix $\hat{\Sigma}^{-1}$ with 9 strictly lower triangular elements, and generate $N = 100$ samples from $\mathcal{N}(0, \hat{\Sigma})$. The sample covariance matrix for this problem has a dense inverse, and hence cannot be used to estimate the graph topolgy.

We solve the penalized ML problem (41) for a completely interconnected graph, and for different values of γ. Because of the constraint in (41) the solutions K are sparse and become denser as γ is increased. For each γ we identify a sparsity pattern by discarding very small elements of K. We then recompute, for that particular sparsity pattern, the (non-penalized) ML estimate and denote the resulting covariance estimate by $\Sigma_{\text{pml}}(\gamma)$. Figure 7 show the log-likelihood, AIC, and BIC scores of $\Sigma_{\text{pml}}(\gamma)$ as a function of the trade-off parameter γ. The topologies with the best AIC and BIC scores turn out to be almost identical; we choose the estimate corresponding to the best BIC score and call this Σ_{pml}. The sparsity pattern of Σ_{pml}^{-1} is shown in figure 8, together with the sparsity pattern of the true concentration matrix $\hat{\Sigma}^{-1}$. As we see, the two sparsity patterns are quite similar. Table 3 compares the numerical values of selected entries of Σ_{pml}^{-1} with the values of $\hat{\Sigma}^{-1}$.

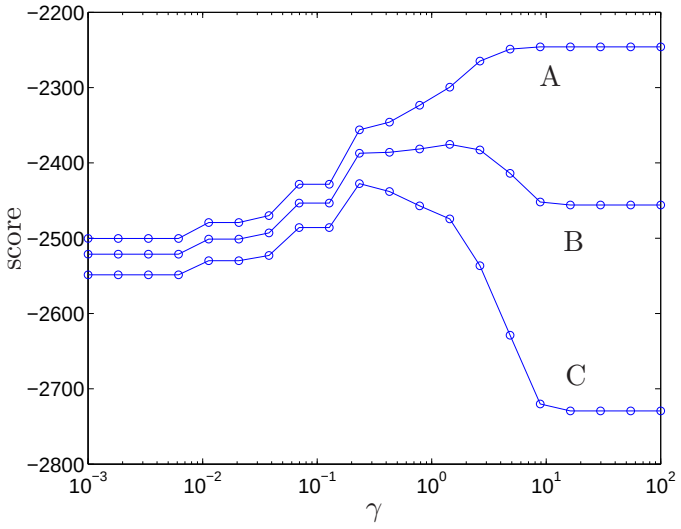

Figure 7: Log likelihood (A), AIC (B) and BIC (C) scores for covariance matrices estimated using the penalized maximum-likelihood method as a function of the trade-off parameter γ.

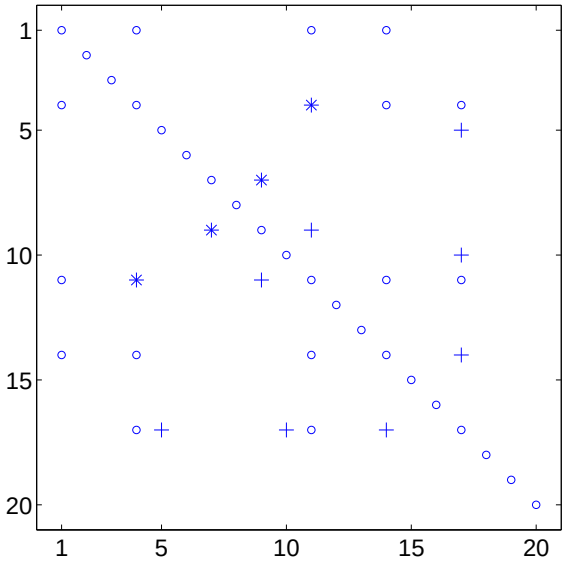

Figure 8: Sparsity patterns of the 'true' concentration matrix $\hat{\Sigma}^{-1}$ and the estimate Σ_{pml}^{-1} obtained via penalized ML estimation. Entries that are nonzero in both the true and the estimated concentration matrix are marked with '$\circ$'. Entries that are nonzero in the estimated concentration matrix but not in the true concentration matrix (false positives) are marked with '+'. Entries that are nonzero in the true concentration matrix but not in the estimate (misses) are marked with '*'.

(i,j)	$(\hat{\Sigma}^{-1})_{ij}$	$(\Sigma_{\mathrm{pml}}^{-1})_{ij}$
(4,1)	0.1182	0.1323
(11,1)	-0.3433	-0.4238
(14,1)	0.1447	0.1777
(11,4)	0.0454	0
(14,4)	0.0324	0.0568
(17,4)	0.3922	0.3504
(17,5)	0	0.0120
(9,7)	-0.0179	0
(11,9)	0	-0.0175
(17,10)	0	-0.0137
(14,11)	-0.0942	-0.1177
(17,11)	0.1350	0.1031
(17,14)	0	0.0192

Table 3: Numerical values of the concentration matrix estimated via penalized ML estimation and the true concentration matrix. The sparsity patterns of the matrices is shown in figure 8.

8 Conclusions

We have discussed efficient implementations of convex optimization algorithms for maximum likelihood estimation of normal graphical models with large sparse graphs. The algorithms use a chordal embedding to exploit sparsity when evaluating objective functions and gradients. This allows us to solve problems with several 10,000 nodes. We numerically compared different gradient methods: coordinate steepest descent, conjugate gradient, and limited memory quasi-Newton methods. The best results were achieved by the limited memory BFGS method. We also presented a dual algorithm that exploits results from matrix completion theory and is particularly well suited for problems with sparsity patterns that are almost chordal, *i.e.*, where the chordal embedding adds relatively few edges. Finally, we discussed the problem of topology selection and described a heuristic method for estimating the graph topology via penalized maximum likelihood estimation.

References

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, AC-19(6):716–723, 1974.

[2] P. Amestoy, T. Davis, and I. Duff. An approximate minimum degree ordering. *SIAM Journal on Matrix Analysis and Applications*, 17(4):886–905, 1996.

[3] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. John Wiley & Sons, second edition, 1984.

[4] O. Banerjeee, A. d'Aspremont, and L. El Ghaoui. Sparse covariance selection via robust maximum likelihood estimation. ArXiV cs.CE/0506023, July 2005.

[5] W. W. Barrett, C. R. Johnson, and M. Lundquist. Determinantal formulae for matrix completions with chordal graphs. *Linear Algebra and Its Applications*, 121:265–289, 1989.

[6] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: numerical methods.* Athena Scientific, 1997.

[7] J. R. S. Blair and B. Peyton. An introduction to chordal graphs and clique trees. In A. George, J. R. Gilbert, and J. W. H. Liu, editors, *Graph Theory and Sparse Matrix Computation.* Springer-Verlag, 1993.

[8] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[9] K. P. Burnham and R. D. Anderson. *Model Selection and Inference: A practical Information-Theoretical Approach.* Springer-Verlag, 2nd edition, 2001.

[10] K. P. Burnham and R. D. Anderson. Multimodel inference. Understanding AIC and BIC in model selection. *Sociological Methods & Research*, 33(2):261–304, 2004.

[11] G. R. Cowell, A. P. Dawid, Lauritzen S. L, and Spiegelhalter D. J. *Probabilistic Networks and Expert Systems.* Springer, 1999.

[12] T. Davis. University of florida sparse matrix collection. Available from `http://www.cise.ufl.edu/research/sparse/mat/Kamvar`.

[13] A. P. Dempster. Covariance selection. *Biometrics*, 28(1):157–175, 1972.

[14] A. Dobra, C. Hans, B. Jones, J. R. Nevins, G. Yao, and M. West. Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, pages 196–212, 2004.

[15] A. M. Erisman and W. F. Tinney. On computing certain elements of the inverse of a sparse matrix. *Communications of the ACM*, 18(3):177–179, 1975.

[16] M. Fukuda, M. Kojima, K. Murota, and K. Nakata. Exploiting sparsity in semidefinite programming via matrix completion I: general framework. *SIAM Journal on Optimization*, 11:647–674, 2000.

[17] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders. Methods for modifying matrix factorizations. *Mathematics of Computations*, 28(126):71–89, 1974.

[18] G. H. Golub and C. F. Van Loan. *Matrix Computations.* John Hopkins University Press, 3rd edition, 1996.

[19] R. Grone, C. R. Johnson, E. M Sá, and H. Wolkowicz. Positive definite completions of partial Hermitian matrices. *Linear Algebra and Its Applications*, 58:109–124, 1984.

[20] J. Z. Huang, N. Liu, and M. Pourahmadi. Covariance selection and estimation via penalized normal likelihood. Wharton preprint, 2005.

[21] M. Laurent. Matrix completion problems. In C. A. Floudas and P. M. Pardalos, editors, *Encyclopedia of Optimization*, volume III, pages 221–229. Kluwer, 2001.

[22] S. L Lauritzen. *Graphical Models.* Clarendon Press, Oxford, 1996.

[23] K. Nakata, K. Fujitsawa, M. Fukuda, M. Kojima, and K. Murota. Exploiting sparsity in semidefinite programming via matrix completion II: implementation and numerical details. *Mathematical Programming Series B*, 95:303–327, 2003.

[24] Yu. Nesterov and A. Nemirovsky. *Interior-point polynomial methods in convex programming*, volume 13 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.

[25] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2001.

[26] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[27] D. J. Rose. Triangulated graphs and the elimination process. *Journal of Mathematical Analysis and Applications*, 32:597–609, 1970.

[28] D. J. Rose, R. E. Tarjan, and G. S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing*, 5(2):266–283, 1976.

[29] G. Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6:461–4, 1978.

[30] T. P. Speed and H. T. Kiiveri. Gaussian Markov distributions over finite graphs. *The Annals of Statistics*, 14(1):138–150, 1986.

[31] R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on Computing*, 13(3):566–579, 1984.

[32] N. Wermuth. Linear recursive equations, covariance selection, and path analysis. *Journal of the American Statistical Association*, 75(372):963–972, 1980.

[33] N. Wermuth and E. Scheidt. Algorithm AS 105: Fitting a covariance selection model to a matrix. *Applied Statistics*, 26(1):88–92, 1977.