Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros Informáticos

# Unifying methodologies for graphical models with Gaussian parametrization

PhD Thesis

**Irene Córdoba**
MSc Artificial Intelligence

2020

Departamento de Inteligencia Artificial
Escuela Técnica Superior de Ingenieros Informáticos

# Unifying methodologies for graphical models with Gaussian parametrization

**Irene Córdoba**
MSc Artificial Intelligence

Supervised by:

**Pedro Larrañaga**
PhD Computer Science
**Concha Bielza**
PhD Computer Science

2020

# Tribunal

- **Presidente**: Serafín Moral
- **Secretario**: Juan Antonio Fernández del Pozo
- **Vocal**: Antonio Salmerón
- **Vocal**: José M. Peña
- **Vocal**: Robert Castelo

# Agradecimientos

Esta tesis ha sido financiada con la ayuda predoctoral FPU15/03797 del Ministerio de Ciencia, Innovación y Universidades.

Si en 2014 Pedro y Concha no me hubieran invitado a unirme a su grupo de investigación, probablemente yo ni siquiera hubiera hecho un doctorado, así que esta tesis existe gracias a ellos. No sólo debido a la oportunidad que me brindaron, si no también a la confianza que durante todos estos años han depositado en mi trabajo (incluso cuando ni yo misma la tenía), y a la libertad con la que me han permitido desarrollarlo.

Mis compañeros del grupo de investigación (los presentes y los que ya se fueron) han sabido proveer distensión y desconexión de la vorágine investigadora. Merece una mención especial Gherardo Varando, que ha sido un apoyo tanto profesional como personal. Empezamos a colaborar cuando él estaba a punto de doctorarse, y hemos continuado hasta el día de hoy. Sin él, gran parte del contenido de esta tesis sería mucho menos interesante.

Mi familia (la presente y la que ya se fue) es una fuente constante de inspiración y cariño. En especial, a mis padres José Luis y María les agradezco la dedicación y el esfuerzo que han invertido en todas las etapas de mi educación (y de todo lo demás). Si tuviera que enumerar todo lo positivo que me han aportado y que ha hecho que yo esté hoy aquí, el resto del documento empequeñecería demasiado. Esta tesis, como todos mis logros, es suya.

Finalmente, a mi pequeña familia, que comencé a construir a la vez que inicié la tesis: Eduardo, Ramón y Juan. No puedo encontrar palabras que expresen lo que habéis significado y significáis día a día. Sois los mejores compañeros que hubiera podido tener en este viaje. Vosotros dais a la vida todo el sentido que puede tener. Mis tres chicos.

# Resumen

Los modelos gráficos representan independencias condicionales de una distribución multivariante mediante aristas ausentes en un grafo, que típicamente es dirigido, no dirigido o mixto. Esta modelización compacta permite descomponer la inferencia estadística en computaciones eficientes sobre el correspondiente grafo. Es por ello que los modelos gráficos se originaron en la intersección entre la estadística y la inteligencia artificial, siendo las redes de Markov (grafo no dirigido) y las redes Bayesianas (grafo dirigido acíclico) los representantes clásicos. Hoy en día los modelos gráficos se aplican extensamente y una cantidad significativa de investigación se dedica a ellos, incluyendo las clásicas redes de Markov y Bayesianas.

Las redes de Markov Gaussianas y las redes Bayesianas Gaussianas, a pesar de no ser modelos equivalentes, comparten una intersección común consistente en los grafos cordales (o grafos dirigidos acíclicos sin v-estructuras). Un método habitual para la selección del modelo en ambas clases es el contraste de hipótesis, y supone la selección del grafo que parametriza el modelo. Las aristas ausentes en ambos modelos se representan mediante un patrón de ceros en la matriz inversa de covarianza o de correlación parcial (redes de Markov Gaussianas) o en su descomposición de Cholesky (redes Bayesianas Gaussianas). Después, sus parámetros son estimados por máxima verosimilitud. Como alternativa, existen en el estado del arte métodos de regularización para ambas clases de modelos, que simultáneamente realizan la selección y estimación del modelo.

Un método popular para la selección del modelo mediante contraste de hipótesis es el algoritmo PC, que se puede aplicar tanto para redes de Markov Gaussianas como para redes Bayesianas Gaussianas. Este método depende fundamentalmente de dos parámetros: el tipo de test estadístico y el nivel de significatividad al que se contrastan las hipótesis. Sin embargo, el enfoque actual en la literatura es usar un test Gaussiano para una transformación de la correlación parcial, y una búsqueda en rejilla para su nivel de significatividad. Por contra, cuando se usa un procedimiento automático para afinar los parámetros, como la optimización Bayesiana, se muestra cómo se mejora significativamente el rendimiento de la selección del modelo cuando se emplea un test no usado habitualmente en la literatura. Es más, estos procedimientos automáticos de afinación de parámetros permiten seleccionar un nivel de significatividad optimizado para cada tipo de test.

A la validación de metodologías para selección de modelos gráficos Gaussianos le afecta profundamente, además de cómo se hace la afinación de parámetros, cómo se simulan los modelos de test sintéticos. Se puede mostrar que las metodologías que tratan esta tarea en el estado del arte, tanto para redes de Markov Gaussianas como para redes Bayesianas Gaussianas, están sesgadas hacia ciertas regiones, influenciando así significativamente sobre los resultados de validación. Sería por tanto deseable disponer de un proceso para

muestrear uniformemente modelos gráficos Gaussianos. En concreto, las redes Bayesianas Gaussianas y las redes de Markov Gaussianas están íntimamente relacionadas con la matriz de correlación parcial, por lo que métodos de muestreo uniforme de dicho conjunto, llamado *elliptope*, pueden ser un punto de partida. Se propone un nuevo método tipo Metrópolis para muestrar uniformemente del *elliptope*, extensible de manera directa a modelos gráficos Gaussianos cordales. Sin embargo, en el caso general, se debe usar un método de ortogonalización parcial para las redes de Markov Gaussianas, y no queda garantizado que los resultados sean uniformes. Pese a esta dificultad, se muestra cómo constituye una metodología de simulación alternativa de modelos gráficos Gaussianos que ilustra cómo resultan profundamente afectados los resultados de validación, y por tanto cómo los experimentos de simulación se deben examinar cuidadosamente, si no se usa muestreo uniforme.

Finalmente, ya se ha mencionado que el grafo asociado tanto con las redes Bayesianas Gaussianas como con las redes de Markov Gaussianas está codificado directamente en la matriz de correlación parcial o de covarianza inversa, o en su descomposición de Cholesky. Otro modelo gráfico Gaussiano, el grafo de covarianza, se puede leer del patrón de ceros en una matriz de covarianza. Sin embargo, no exiten trabajos en la literatura que propongan un modelo gráfico Gaussiano sobre el factor de Cholesky de una matriz de covarianza. Se muestra cómo este modelo es un análogo de la red Bayesiana Gaussiana, de la misma manera que un grafo de covarianza lo es de una red de Markov Gaussiana. Cuando las variables siguen un orden conocido, este nuevo modelo gráfico Gaussiano se puede estimar fácilmente como una factorización de la matriz de covarianza restringida a tener muchos ceros. Esto ya se ha tratado en la literatura, pero solamente mediante una transformación del modelo a regresión. Este vacío puede llenarse usando un enfoque de pérdida matricial regularizada que penaliza directamente la función de verosimilitud, u otras funciones de pérdida de interés. Se muestra cómo este modelo de aprendizaje produce una mejor recuperación del patrón de ceros así como resultados competitivos en escenarios reales.

# Abstract

Graphical models represent conditional independences of a multivariate distribution by absent edges in a graph, which typically is directed, undirected or mixed. This compact modelling allows to decompose statistical inference into efficient computations over the associated graph. As such, graphical models originated mainly at the interface between statistics and artificial intelligence, with Markov networks (undirected graph) and Bayesian networks (acyclic digraph) being the classic representatives. Nowadays graphical models are widely applied and a significant amount of research is devoted to them.

Gaussian Markov networks and Gaussian Bayesian networks, although not being equivalent models, share a common intersection consisting of chordal graphs (or acyclic digraphs with no v-structures). A typical approach for model selection in both model classes is hypothesis testing, which amounts to selecting the graph that parametrizes the model. Absent edges in both models are represented by a zero pattern in the inverse covariance or partial correlation matrix (Gaussian Markov networks) or in its Cholesky decomposition (Gaussian Bayesian networks). Afterwards, their parameters are estimated by maximum likelihood. Alternatively, there exist state-of-the-art regularisation methods for both model classes, which simultaneously perform model selection and estimation.

A popular method for model selection via hypothesis testing is the PC algorithm, which can be applied for both Gaussian Markov networks and Gaussian Bayesian networks. This method mainly depends on two parameters: the statistical test type and the significance level at which the hypotheses are tested. However, the usual approach in the literature is to use a Gaussian test for a transformation of the partial correlation, and a grid search for its significance level. By contrast, when using an automatic procedure for parameter tuning, such as Bayesian optimization, it is shown how model selection performance is significantly improved when employing an uncommonly used test in the literature. Furthermore, these automatic parameter tuning procedures allow to select a significance level optimized for each test type.

Validation of methodologies for Gaussian graphical model selection is also deeply affected, apart from how parameter tuning is performed, by how synthetic test models are simulated. It can be shown that state-of-the-art methodologies addressing this task are biased towards certain regions, thereby significantly influencing validation results. It would be therefore desirable to have a uniform sampling procedure for Gaussian graphical models. In particular, both Gaussian Bayesian networks and Gaussian Markov networks are intimately related with the partial correlation matrix, thereby uniform sampling methods for such set, called elliptope, can be a departing point. A novel Metropolis uniform sampling from the elliptope is proposed, which can be straightforwardly extended to chordal Gaussian graphical models. However, in the general case, a partial orthogonalization method has to be used for Gaussian Markov networks, and the results are not

guaranteed to be uniform. Despite this difficulty, it is shown to be an alternative simulation methodology for Gaussian graphical models which also illustrates how validation results are deeply affected, and therefore how simulated experiments need to be carefully examined, when not using uniform sampling.

Finally, it has already been mentioned that the graph associated with both Gaussian Bayesian networks and Gaussian Markov network models is directly encoded in the partial correlation or inverse covariance matrix, or in its Cholesky decomposition. Another Gaussian graphical model, the covariance graph, can be read from a zero pattern in the covariance matrix. However, there is no work in the literature that proposes a Gaussian graphical model over the Cholesky factor of a covariance matrix. It is shown that this model is an analogue of the Gaussian Bayesian network, in the same way that a covariance graph is of a Gaussian Markov network. When the variables follow a known order, this new Gaussian graphical model can be easily estimated as a sparse Cholesky factorization of the covariance matrix. This has been previously addressed in the literature, but only via a regression transformation of the model. This gap can be filled by using a regularized matrix loss approach that directly penalizes the likelihood function, or other losses of interest. It is shown how this learning model yields better zero-pattern recovery as well as competitive results in real scenarios.

# Contents

# Chapter 1

# Introduction

Graphical or Markov models provide a graphical way of modelling a multivariate distribution, which at the same time allows to decompose statistical inference. They are essentially determined by three aspects:

- The statistical family of distributions under consideration.
- The graphs allowed for representing the distribution.
- How graph separation properties are related to statistical independences in the probability distribution, the so-called *Markov properties* of the graphical model.

With such interdisciplinary nature, it is not a surprise that historically graphical models originated at the interface between many fields, including physics, statistics and artificial intelligence. Indeed, they can be traced to the Ising model for ferromagnetic materials (Kindermann and Snell, 1980; Isham, 1981), Markov random fields (Grimmett, 1973; Besag, 1974; Moussouris, 1974), or path analysis for genetics (Wright, 1934). Wermuth (1976a) hinted that the work of Dempster (1972) was a Gaussian undirected graphical model, by noting the similarities with log-linear contingency tables (Darroch et al., 1980). In addition, she implicitly introduced Gaussian acyclic directed graphical models as zero values for coefficients in linear recursive regressions (Wermuth, 1980), which she further compared with Dempster's models and path analysis. The explicit terminology *graphical model*, however, was not introduced until Darroch et al. (1980) linked log-linear models for contingency tables with discrete Markov fields. In parallel, within the artificial intelligence community, Pearl (1988) actively developed acyclic directed and undirected graphical models, terming them *Bayesian* and *Markov networks*, respectively. It was the birth of graphical model's research (see also Córdoba et al., 2020c, §2).

Nowadays graphical models are still widely applied and a significant amount of research is devoted to them (Maathuis et al., 2018). This thesis contains an exposition of some contributions to the statistical analysis of graphical models for the multivariate Gaussian distribution, commonly called Gaussian graphical models, from a unifying perspective, that is, taking into account how the proposed methodologies can be transferred throughout different Gaussian graphical models.

## 1.1 Aims and scope

The contributions of this thesis exclusively focus on statistical methodology related to Gaussian graphical models. Therefore it will generally be assumed that random vectors follow a multivariate Gaussian distribution. In any case, this will usually be explicitly stated whenever defining a random vector. With the scope restricted to this assumption, the following hypotheses are formulated in this thesis:

- State-of-the-art learning methods for Gaussian graphical models can be improved by using modern methodologies for parameter optimisation.
- Different ways of simulating Gaussian graphical models may deeply affect synthetic evaluation of state-of-the-art learning methods. In particular, special instances of Gaussian graphical models can be uniformly sampled to allow fair synthetic model comparison.
- The analogies between parameters of Gaussian Markov and Bayesian networks (Dempster, 1969; Wermuth et al., 2006) can be used for defining an alternative Gaussian graphical model.

Based on such hypotheses, the main thesis objectives are therefore:

1. To show how modern parameter optimisation techniques can improve the performance of a state-of-the-art learning algorithm for Gaussian graphical models.
2. To develop simulation methods for Gaussian graphical models different from the traditional ones, and show how synthetic validation can be deeply affected.
3. To detail the analogies between parametrizations of Gaussian Markov and Bayesian networks, and show how they motivate the introduction of a new Gaussian graphical model.

## 1.2 Contributions

In this thesis the objectives previously outlined have been achieved. In particular,

- In Córdoba et al. (2018a) the state-of-the-art PC algorithm (Spirtes et al., 2000) improves its performance when selecting the parameters with Bayesian optimization (Garrido-Merchán and Hernández-Lobato, 2019), thereby fulfilling Objective 1.
- Objective 2 has yielded several contributions. It is primarily addressed in Córdoba et al. (2018c), where a new simulation method for Gaussian Markov networks is proposed, and it is shown how it deeply affects different validation scenarios where several state-of-the-art learning methods are tested. In parallel, as a first step towards providing unbiased simulation for Gaussian graphical models, in Córdoba et al. (2018b) a new method for uniform sampling of correlation matrices is detailed. This method is extended in Córdoba et al. (2020a), providing uniform sampling for special types of Gaussian Markov and Bayesian networks.

- In Córdoba et al. (2020b) a new learning method for the sparse Cholesky decomposition of a covariance matrix is proposed. Motivated by the analogy with sparse inverse covariance decompositions, and Gaussian Markov and Bayesian networks, a new Gaussian graphical model is introduced, thereby fulfilling Objective 3.

Furthermore, the review Córdoba et al. (2020a), unrelated with the above methodological objectives, constitutes another contribution of this thesis to state of the art of Gaussian graphical models.

## 1.3 Document structure

This thesis is organized as follows. Chapter 2 contains an overview of the main statistical theory related to Gaussian graphical models that will be necessary to follow the thesis remainder, specially Markov and Bayesian networks. Maximum likelihood estimation, stepwise model selection, and penalized learning are covered. Chapter 3 contains an illustration of how the PC algorithm's model selection performance is improved by automatically tuning its parameters with Bayesian optimisation. This also serves as an introduction to another significant issue that affects numerical validation of Gaussian graphical model selection methods: how synthetic models are simulated. Chapter 4 contains an exemplification of how to uniformly sample from the set of correlation matrices, intimately related to Gaussian graphical models. Its results are then combined with a partial orthogonalization method in Chapter 5, providing thus several alternatives to traditional synthetic model simulation in Gaussian graphical model literature. It also contains an illustration of how these different simulation methodologies can deeply affect validation results and conclusions, confirming what was previously theorised in Chapter 3. Finally, the attention is switched in Chapter 6 to the sparse parametrization of the covariance matrix in a multivariate Gaussian distribution, motivated by the inverse analogue, which plays a key role for Gaussian Markov and Bayesian networks. The conclusions and open research that could be drawn from this thesis are exposed in Chapter 7. Finally, it is closed by a bibliography containing details of all the references made throughout the text.

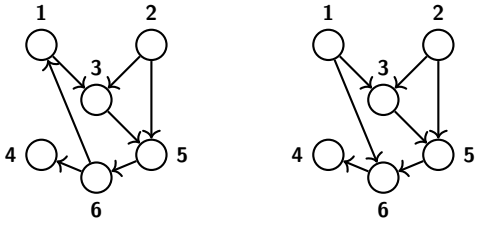
# Chapter 2

# Background

This chapter contains a review of the main statistical theory about Gaussian graphical models, specially Markov and Bayesian networks. Maximum likelihood estimation, stepwise model selection, and penalized learning are covered. The reader already familiarized with this theory can go quickly through this chapter and just focus on familiarizing with the notation.

## 2.1 Graph preliminaries

A graph $\mathcal{G} = (V, E)$, where $V$ is the vertex set and $E \subseteq V \times V$ is the edge set, is called *undirected* if $(u, v) \in E \iff (v, u) \in E$, and *directed* or *digraph* otherwise. A cycle of length $k \geq 2$ in $u \in V$ is an ordered sequence of vertices $u(= v_0), v_1, \ldots, v_{k-1}, u(= v_k)$ where $(v_{i-1}, v_i) \in E$ for $i \in \{1, \ldots, k\}$ and $v_0, \ldots, v_{k-1}$ are distinct. An *acyclic digraph* is a directed graph with no cycles (see Figure 2.1(a) where $1, 3, 5, 6, 1$ is a cycle of length 4, and Figure 2.1(b) for an acyclic digraph). The set of neighbours of a vertex $v \in V$




(a) Cyclic digraph (b) Acyclic digraph

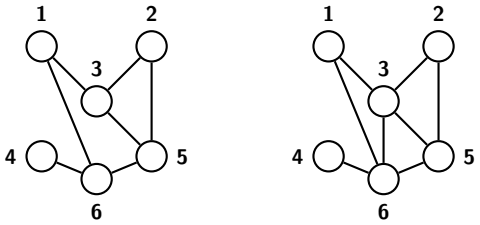
Figure 2.1: Directed graphs.

is defined as $\text{ne}(v) = \{u \in V : (u, v) \in E\}$. If the graph is acyclic directed, such set is commonly called the *parent set* and denoted as $\text{pa}(v)$.

The remainder of this section contains an introduction to chordal graphs, which are key for this thesis and graphical models in general, and some relationships they have with acyclic digraphs.

### 2.1.1 Chordal graphs

An undirected graph $\mathcal{G} = (V, E)$ is called *chordal* if all cycles $(u =)v_0, \ldots, v_k, (= u)$ of length $k \geq 4$ have an edge joining two non-consecutive nodes, that is, if there exist $i, j \in \{0, \ldots, k\}$ with $|i - j| \neq 1$ such that $(v_i, v_j) \in E$. Such edge is commonly denominated a *chord*. As an example, in Figure 2.2(a), the cycle $1, 3, 5, 6, 1$ does not have a chord, whereas the graph in Figure 2.2(b) is chordal. Observe that solid lines are used for undirected graphs because direction is unimportant. Because of the graphical representation that arises, chordal graphs are often called *triangulated*. A *chordal cover*




(a) Non-chordal graph (b) Chordal graph

Figure 2.2: Undirected graphs.

or *triangulation* of an undirected graph $\mathcal{G} = (V, E)$ is a chordal graph $\overline{\mathcal{G}} = (V, \overline{E})$ that contains $\mathcal{G}$ as a sub-graph, that is, such that $E \subseteq \overline{E}$. For example, the graph in Figure 2.2(b) is a chordal cover of that in Figure 2.2(a).

Chordal graphs may be characterized in alternative ways (Lauritzen, 1996). Some of them will be used in the thesis remainder, and thus appear in Proposition 2.1.1. They build upon further graph concepts that will be explained below.

An undirected graph $G = (V, E)$ is complete if every two vertices are connected, that is, if $\mathrm{ne}(u) = V \setminus \{u\}$ for every $u \in V$. Furthermore, a vertex subset $C \subseteq V$ that induces a complete sub-graph $\mathcal{G}_C = (C, E_C = E \cap (C \times C))$ is called a *clique*. Let $C_1, \ldots, C_k$ be a sequence of vertex subsets in $\mathcal{G}$, not necessarily cliques. Define for each $i = 1, \ldots, k$ the subsets $H_i = C_1 \cup \cdots \cup C_i$, $R_i = C_i \setminus H_i$ and $S_i = H_{i-1} \cap C_i$. The sequence $C_1, \ldots, C_k$ is said to be *perfect* if for all $i$ the sets $S_i$ are cliques, and $S_i \subseteq C_j$ when $i > 1$ for some $j < i$. A *perfect numbering* of the vertices in $V$ is an ordering $v_1 \prec \cdots \prec v_k$ such that $C_j = (\{v_j\} \cup \mathrm{ne}(v_j)) \cap \{v_1, \ldots, v_j\}$ with $j = 1, \ldots, k$ is a perfect sequence of vertex subsets.

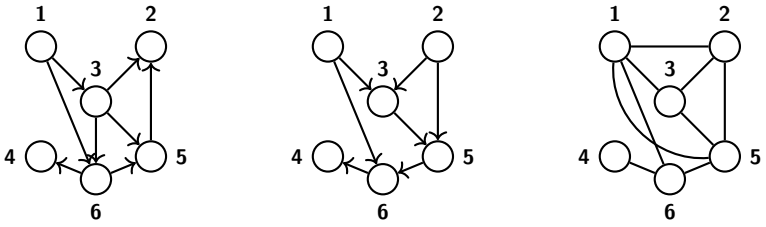
**Proposition 2.1.1** (Lauritzen, 1996)**.** *The following statements are equivalent for an undirected graph $\mathcal{G}$:*

- *$\mathcal{G}$ is chordal.*
- *The vertices of $\mathcal{G}$ admit a perfect numbering.*
- *The maximal cliques of $\mathcal{G}$ can be numbered to form a perfect sequence.*

Note that, following Proposition 2.1.1, in a chordal graph $\mathcal{G} = (V, E)$ an immediate perfect ordering of $V$ may be formed from a perfect sequence of the maximal cliques $C_1, \ldots, C_k$ by taking first those vertices in $C_1$, then those in $R_2$, $R_3$ and so on.

### 2.1.2 Acyclic digraphs and chordal graphs

An acyclic digraph $\mathcal{G} = (V, E)$ is contained by two special undirected graphs, its skeleton and moral graph, which will be explained subsequently. The undirected graph $\mathcal{G}^U = (V, E^U = E \cup \{(v, u) : (u, v) \in E\}$ is called the *skeleton* of $\mathcal{G}$, and, conversely, $\mathcal{G}$ is one of its *orientations*. For example, the undirected graph in Figure 2.2(b) is the skeleton of the acyclic digraph in Figure 2.3(a). Let $u, w_1, w_2 \in V$ with $(w_1, u), (w_2, u) \in E$ and




(a) Orientation (b) v-structures (c) Moral graph

Figure 2.3: Chordal graphs and acyclic digraphs.

$(w_1, w_2), (w_2, w_1) \notin E$ (see vertices 1, 2 and 3 in Figure 2.1(b)). Such configurations are usually called *v-structures* and denoted as $w_1 \to u \leftarrow w_2$. The *moral graph* of $\mathcal{G}$ is defined as the undirected graph $\mathcal{G}^m = (V, E^m = E^U \cup \{(w_1, w_2) : w_1 \to u \leftarrow w_2 \text{ for some } u \in V\}$. Figure 2.3(b) has the v-structures $1 \to 3 \leftarrow 2$ and $1 \to 6 \leftarrow 5$, whereas Figure 2.3(a) contains none. The equivalence between an acyclic digraph's moral graph and skeleton is closely related to chordal graphs, as Proposition 2.1.2 states.

**Proposition 2.1.2** (Koller and Friedman, 2009)**.** *The moral graph $\mathcal{G}^m$ and the skeleton $\mathcal{G}^U$ of an acyclic digraph $\mathcal{G}$ coincide if and only if $\mathcal{G}$ has no v-structures. In such case, both $\mathcal{G}^m$ and $\mathcal{G}^U$ are chordal.*

Continuing with the example, since the acyclic digraph of Figure 2.3(a) has no v-structures, its moral graph is its skeleton, Figure 2.2(b). However, the acyclic digraph in Figure 2.3(b) has two v-structures, therefore its moral graph, depicted in Figure 2.3(c), contains two edges more than its skeleton (Figure 2.2(a)). Note that, in general, neither the moral graph nor the skeleton are chordal.

Conversely to Proposition 2.1.2, any chordal undirected graph $\mathcal{G} = (V, E)$ can be oriented into an acyclic digraph with no v-structures. Indeed, a well-known property of acyclic digraphs is that their nodes can be totally ordered such that each node is preceded by its parents in such order; this is usually called *ancestral* or *topological ordering*. It can be shown (Lauritzen, 1996) that if $v_1 \prec \cdots \prec v_p$ is a perfect ordering of $V$, then $\prec$ is also an ancestral ordering for an orientation of $\mathcal{G}$ that contains no v-structures.

## 2.2 Gaussian graphical models

Let $\boldsymbol{X} = (X_1, \ldots, X_p)^t$ be a $p$-variate random vector. In graphical models, the vertex set $V$ of a graph $\mathcal{G} = (V, E)$ is typically used as an index set for $\boldsymbol{X}$, thus $V = \{1, \ldots, p\}$. In the following, for an arbitrary subset $I \subseteq V$, $X_I$ will denote the $|I|$-variate random vector indexed by $I$. Furthermore, the notation of Dawid (1979) for conditional independence will be followed, thus for disjoint $I, J, K \subseteq V$, $\boldsymbol{X}_I \perp\!\!\!\perp \boldsymbol{X}_J \mid \boldsymbol{X}_K$ will mean that $\boldsymbol{X}_I$ and

$\boldsymbol{X}_J$ are independent given $\boldsymbol{X}_K = \boldsymbol{x}_K$, for any value of $\boldsymbol{x}_k$. Note that, in terms of density functions, this means, among other identities, that $f(\boldsymbol{x}_I, \boldsymbol{x}_J \mid \boldsymbol{x}_K) = f(\boldsymbol{x}_I \mid \boldsymbol{x}_K) f(\boldsymbol{x}_J \mid \boldsymbol{x}_K)$ (see also Studený, 2018, §1.3).

### 2.2.1 Markov and Bayesian networks

There are several Markov properties that can be defined for both acyclic directed and undirected graphs (Lauritzen, 1996; Studený, 2018, §1.7 and §1.8). For the multivariate Gaussian distribution they are all equivalent (Pearl, 1988; Lauritzen et al., 1990), therefore hereby only those relevant for this thesis will be described.

If $\mathcal{G} = (V, E)$ is an undirected graph, then the probability distribution of a random vector $\boldsymbol{X}$ is said to be *pairwise Markov* or satisfy the pairwise Markov property with respect to $\mathcal{G}$ if

$$X_i \perp\!\!\!\perp X_j \mid \boldsymbol{X}_{V \setminus \{i,j\}} \text{ for all } (i,j) \notin E. \tag{2.1}$$

Analogously, in an acyclic digraph $\mathcal{G} = (V, E)$, denoting as $\prec$ an ancestral order of $V$, the probability distribution of $\boldsymbol{X}$ is said to be *ordered Markov* or satisfy the ordered Markov property with respect to $G$ if

$$X_i \perp\!\!\!\perp X_j \mid \boldsymbol{X}_{\mathrm{pa}(i)} \text{ for all } (j,i) \notin E \text{ with } j \prec i. \tag{2.2}$$

Given a statistical family $\mathcal{F}$ of distributions, such as the multivariate Gaussian in the case of this thesis, and a graph $\mathcal{G}$, the *graphical* or *Markov model* $\mathcal{M}(\mathcal{G})$ is defined as the set of distributions in $\mathcal{F}$ that satisfy the corresponding Markov property with respect to $\mathcal{G}$, Equation (2.1) or (2.2) depending on whether it is undirected or acyclic directed. They are also called Markov networks and Bayesian networks, respectively. The interested reader may look at Lauritzen and Sadeghi (2018) for an account of other, more complex, graph types and their associated Markov properties, which give rise to other graphical models.

There is a key difference between undirected and acyclic directed graphical models: the former are uniquely parametrized by the graph (Pearl and Paz, 1987), whereas for the latter this uniqueness does not hold. Instead, if $\mathcal{G}$ is an acyclic digraph, then any other that shares the skeleton and v-structures (Verma and Pearl, 1991) will yield the same graphical model. This gives rise to the *Markov equivalence class* of $\mathcal{G}$, $[\mathcal{G}]$, which contains all acyclic digraphs sharing skeleton and v-structures with $\mathcal{G}$.

Finally, Markov equivalence can also be established between acyclic directed and undirected graphical models. A chordal graph is Markov equivalent to an orientation that does not contain v-structures (Frydenberg, 1990). Conversely, if $\mathcal{G}$ is an acyclic digraph and $\mathcal{G}^m$ is its moral graph, then $\mathcal{M}(\mathcal{G}) \subseteq \mathcal{M}(\mathcal{G}^m)$ (Lauritzen et al., 1990). Furthermore, if $\mathcal{G}$ contains no v-structures, then $\mathcal{G}^m$ is chordal and equal to the skeleton (Proposition 2.1.2), therefore the respective Markov models $\mathcal{M}(\mathcal{G})$ and $\mathcal{M}(\mathcal{G}^m)$ coincide.

### 2.2.2 The multivariate Gaussian parametrization

From now on the statistical family $\mathcal{F}$ underlying a graphical model $\mathcal{M}(\mathcal{G})$ over graph $\mathcal{G}$ will be restricted to the multivariate Gaussian, that is, distributions of the form $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Sigma}$ is the covariance matrix, which belongs to the set of positive definite symmetric matrices, $\mathbb{S}^{>0}$. If $\mathcal{G}$ is undirected, the parameters of $\mathcal{M}(\mathcal{G})$ are

in correspondence with the inverse covariance matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$, whereas if it is an acyclic digraph this correspondence is with its Cholesky decomposition (Wermuth, 1976a, 1980; Uhler, 2018, §9.1). Therefore, in the remainder we will assume a zero mean, $\boldsymbol{\mu} = \mathbf{0}$ for a lighter notation. In the following, $\mathbf{M}_{IJ}$ will be the $|I| \times |J|$ sub-matrix of a real $q \times r$ matrix $\mathbf{M}$, where $I \subseteq \{1, \ldots, q\}$ and $J \subseteq \{1, \ldots, r\}$; and $\mathbf{M}_{IJ}^{-1}$ will mean $(\mathbf{M}_{IJ})^{-1}$.

Let $\mathcal{G} = (V, E)$ be an acyclic digraph with vertex set $V = \{1, \ldots, p\}$ indexing a $p$-variate random vector $\boldsymbol{X} = (X_1, \ldots, X_p)^t$. Denote with $\prec$ the ancestral order of $\mathcal{G}$ and for a vertex $i \in V$ let $\mathrm{pr}(i) = \{j \in V : j \prec i\}$ be its predecessor set. Observe that, by definition of ancestral order, $\mathrm{pa}(i) \subseteq \mathrm{pr}(i)$. Assume that $\boldsymbol{X}$ follows a $p$-variate $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. For every $j \in \mathrm{pr}(i) \setminus \mathrm{pa}(i)$ (Anderson, 2003)

$$X_i \perp\!\!\!\perp X_j \mid \boldsymbol{X}_{\mathrm{pa}(i)} \iff \beta_{ij|\mathrm{pr}(i)} = 0, \tag{2.3}$$

where $\beta_{ij|\mathrm{pr}(i)}$ is the $j$-th entry of vector $\boldsymbol{\beta}_{i|\mathrm{pr}(i)} = \mathbf{\Sigma}_{i\,\mathrm{pr}(i)} \mathbf{\Sigma}^{-1}_{\mathrm{pr}(i)\,\mathrm{pr}(i)}$, that is, the coefficient of $X_j$ in the regression of $X_i$ over $\boldsymbol{X}_{\mathrm{pr}(i)}$, following a slight adaptation of the notation by Yule (1907). Thus a Gaussian Bayesian network model, where Equation (2.3) holds, is equivalent to the set of linear regressions (Anderson, 2003)

$$X_i = \sum_{j \in \mathrm{pr}(i)} \beta_{ij|\mathrm{pr}(i)} X_j + \mathcal{E}_i = \sum_{j \in \mathrm{pa}(i)} \beta_{ij|\mathrm{pa}(i)} X_j + \mathcal{E}_i, \tag{2.4}$$

where $i \in V$ and $\mathcal{E}_i$ are zero mean independent Gaussian variables.

The regression coefficients of Equation (2.4) can be arranged in a matrix $\mathbf{B}$ where $b_{ij} = \beta_{ij|\mathrm{pa}(i)} = \beta_{ij|\mathrm{pr}(i)}$ if $j \in \mathrm{pa}(i)$ and zero otherwise. This leads to the matrix form of Equation (2.4), $\boldsymbol{X} = \mathbf{B}\boldsymbol{X} + \boldsymbol{\mathcal{E}}$ where $\boldsymbol{\mathcal{E}} \sim \mathcal{N}(\mathbf{0}, \mathbf{D})$ with $\mathbf{D}$ diagonal. We can rearrange it and take variances, arriving at

$$\mathbf{\Omega} = \mathbf{\Sigma}^{-1} = \mathbf{U}\mathbf{D}^{-1}\mathbf{U}^t = \mathbf{W}\mathbf{W}^t, \tag{2.5}$$

where $\mathbf{U} = (\mathbf{I} - \mathbf{B})^t$, $\mathbf{I}$ is the identity matrix and $\mathbf{W} = \mathbf{U}\sqrt{\mathbf{V}^{-1}}$. Thus defining the set $\mathbb{M}_{\mathcal{G}}$ of matrices with positive diagonal and a zero pattern compatible with $\mathcal{G}$, that is, such that $m_{ji} = 0$ for all $(j, i) \notin E$, $j \neq i$, the Gaussian Bayesian network can be expressed as

$$\mathcal{M}(\mathcal{G}) = \{\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) : \mathbf{\Sigma}^{-1} = \mathbf{W}\mathbf{W}^t,\ \mathbf{W} \in \mathbb{M}_{\mathcal{G}}\}. \tag{2.6}$$

**Remark.** *Let $\tau$ be the permutation of $X_1, \ldots, X_p$ associated with the ancestral order $\prec$. If $\prec$ is the natural order, $1 \prec \cdots \prec p$, which means that $\tau$ is the identity, then matrix $\mathbf{B}$ is strictly lower triangular, $\mathbf{W}$ is the upper Cholesky factor (Eaton, 1983; Horn and Johnson, 2012) of $\mathbf{\Omega}$ and $\mathcal{M}(\mathcal{G})$ is usually called a* linear structural equation model *(Drton, 2018) or* recursive regression system *(Wermuth, 1980). In a general Gaussian Bayesian network $\mathcal{M}(\mathcal{G})$, however, this will not be the case, and $\tau(\mathbf{W})$ will be the upper Cholesky factor of $\tau(\mathbf{\Omega})$, but neither $\mathbf{B}$ nor $\mathbf{W}$ will be strictly lower and upper triangular, respectively. In fact, if $\tilde{\mathbf{\Omega}}$ is the matrix obtained from $\mathbf{\Omega}$ by reordering rows and columns following the reverse of $\prec$, also known as* fill-in free *or* perfect elimination ordering *(Roverato, 2000), and $\mathbf{N}$ is its lower Cholesky factor, it can be verified that $\mathbf{N}^t$ is equal to the transpose of $\tau(\mathbf{U})$ with respect to its anti-diagonal.*

Conversely to Equation (2.3), we also have for every $i, j \in V$ (Anderson, 2003)

$$X_i \perp\!\!\!\perp X_j \mid \boldsymbol{X}_{V \setminus \{i,j\}} \iff \omega_{ij} = 0, \tag{2.7}$$

where $\omega_{ij}$ is the $(i,j)$ entry of the inverse covariance matrix $\mathbf{\Omega}$. This means that if $\mathcal{G} = (V, E)$ is an undirected graph and we consider the set $\mathbb{S}^{>0}_{\mathcal{G}} = \mathbb{S}^{>0} \cap \mathbb{M}_{\mathcal{G}}$, that is, the set of positive definite matrices whose zeros are compatible with $\mathcal{G}$, then we may express a Gaussian Markov network as

$$\mathcal{M}(\mathcal{G}) = \{\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) : \mathbf{\Sigma}^{-1} \in \mathbb{S}^{>0}_{\mathcal{G}}\}. \tag{2.8}$$

## 2.3 Maximum likelihood estimation

The expression of Gaussian Bayesian and Markov networks in Equations (2.6) and (2.8), respectively, hints that maximum likelihood estimation in each model is going to differ significantly. In the former case, it amounts to simple linear regression coefficients and variances, whereas in the latter a sparse positive definite matrix is sought.

When the graphical model $\mathcal{M}(\mathcal{G})$ is a Gaussian Markov network, which is a regular exponential family (Barndorff-Nielsen, 1978; Uhler, 2018), in general there are no closed formulas for the maximum likelihood estimators. Consider $N$ independent observations from a distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) \in \mathcal{M}(\mathcal{G})$ arranged in a $N \times p$ matrix $\mathbf{x}$. Let $\mathbf{Q}_{\mathcal{G}}$ be the projection of $\mathbf{Q} = \mathbf{X}^t\mathbf{X}$ onto $\mathbb{S}^{\geq 0}_{\mathcal{G}} = \mathbb{M}_{\mathcal{G}} \cap \mathbb{S}^{\geq 0}$, that is, $\mathbf{Q}_{\mathcal{G}}$ has zeros compatible with $\mathcal{G}$ and entries coinciding with $\mathbf{Q}$ otherwise. The sufficient statistic for $\mathcal{M}(\mathcal{G})$ is $\mathbf{Q}_{\mathcal{G}}$, whose closed convex support is $\mathbb{S}^{\geq 0}_{\mathcal{G}}$. Therefore, a maximum likelihood estimator for $\mathbf{\Sigma}$ exists if and only if $\mathbf{Q}_{\mathcal{G}}$ is extendable to a positive definite matrix. This condition for existence is not easily established: for a general graph the problem is still open, see Uhler (2018) for an up-to-date overview of the advances made so far.

By contrast, obtaining the maximum likelihood estimates in a multivariate Gaussian distribution following a Bayesian network model is easy, since theory from multivariate linear regression can be applied (Anderson, 2003). In particular, if $\mathcal{G}$ is an acyclic digraph, recall that if $\boldsymbol{X} = (X_1, \ldots, X_p)^t \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) \in \mathcal{M}(\mathcal{G})$, then the statistical model for $\boldsymbol{X}$ coincides with the system of linear regression equations in Equation (2.4). Thus, if $\mathbf{X}$ is the corresponding $N \times p$ random sample matrix, usual least squares estimation may be used,

$$\begin{aligned}
\hat{\boldsymbol{\beta}}^t_{i|\,\mathrm{pa}(i)} &= \mathbf{Q}_{i\,\mathrm{pa}(i)}\mathbf{Q}^{-1}_{\mathrm{pa}(i)\,\mathrm{pa}(i)}, \\
\hat{d}_{ii} &= \frac{1}{N}(q_{ii} - \hat{\boldsymbol{\beta}}^t_{i|\,\mathrm{pa}(i)}\mathbf{Q}^t_{i\,\mathrm{pa}(i)}),
\end{aligned}$$

where $\mathbf{Q} = \mathbf{X}^t\mathbf{X}$, $q_{ii}$ is the $i$-th diagonal entry in $\mathbf{Q}$, $\hat{d}_{ii}$ is the estimator for the $i$-th conditional variance ($i$-th diagonal entry of $\mathbf{D}$ in Equation (2.5)), and $i \in V$.

## 2.4 Stepwise model selection

Model selection for a graphical model $\mathcal{M}(\mathcal{G})$ consists of learning the graph $\mathcal{G} = (V, E)$ that determines it. When a random vector $\boldsymbol{X} = (X_1, \ldots, X_p)^t$ is assumed to follow a multivariate Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) \in \mathcal{M}(\mathcal{G})$, the classical approach is to test conditional independence via partial correlation testing, as follows. Let $i, k \in I \subseteq V = \{1, \ldots, p\}$ and $J \subseteq V \setminus I$. The partial correlation coefficient between $X_i$ and $X_k$ given $\boldsymbol{X}_J$ is

$$\rho_{ik|J} = \frac{\sigma_{ik|J}}{\sqrt{\sigma_{ii|J}\sigma_{kk|J}}}, \tag{2.9}$$

where $\sigma_{ik|J}$, $\sigma_{ii|J}$ and $\sigma_{kk|J}$ are the respective entries in matrix $\mathbf{\Sigma}_{I|J} = \mathbf{\Sigma}_{II} - \mathbf{\Sigma}_{IJ}\mathbf{\Sigma}_{JJ}^{-1}\mathbf{\Sigma}_{JI}$, the conditional covariance matrix. $X_i$ and $X_k$ are independent given $\boldsymbol{X}_J = \boldsymbol{x}_J$, for any value $\boldsymbol{x}_J$, if and only if $\rho_{ik|J} = 0$ (Anderson, 2003).

When selecting the graph for a Gaussian Markov network model, a starting graph $\mathcal{G} = (V, E)$ (usually the complete one over $V$) is fixed, and sub-graphs raising from one edge removal are tested (Wermuth, 1976b), that is $H_0 : \rho_{ij|V\setminus\{i,j\}} = 0$ is tested for every $(i, j) \in E$. The edge leading to the highest likelihood ratio statistic is then removed, and the procedure is iterated backwardly until no edge can be significantly removed. Under $H_0$, the likelihood ratio statistic distribution can be approximated by a product of univariate Beta distributions (Eriksen, 1996; Lauritzen, 1996).

The above method, however, becomes impractical for high values of $p$ and the sample size $N$. Alternatively, the first phase of the PC algorithm (Spirtes and Glymour, 1991) may be used. This method, although targeted at selecting a Gaussian Bayesian network model $\mathcal{M}(\mathcal{G})$, proceeds by first estimating the skeleton $\mathcal{G}^U$ and then orienting it. Starting from the complete undirected graph, at iteration $l$ for each edge $(i, j) \in E$ such that $|\text{ne}(i) \setminus \{j\}| \geq l$, a subset $J \subseteq \text{ne}(i) \setminus \{j\}$ of cardinality $|J| \leq l$ is selected. Then the hypothesis $H_0 : \rho_{ij|J} = 0$ is tested, and if it cannot be rejected, edges $(i, j)$,$(j, i)$ are removed from $\mathcal{G}^U$. This procedure is outlined in Algorithm 1.

**Algorithm 1** The PC algorithm for recovering the skeleton of a Gaussian Bayesian network model $\mathcal{M}(\mathcal{G})$

**Input:** Sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ from $\mathcal{M}(\mathcal{G})$
**Output:** Undirected graph

$V \leftarrow \{1, \ldots, p\}$
$\hat{E} \leftarrow V \times V \setminus \{(i, i) : i \in V\}$
$\hat{\mathcal{G}}^U \leftarrow (V, \hat{E})$ *// Complete undirected graph*
$l \leftarrow -1$
**repeat**
  $l \leftarrow l + 1$
  **repeat**
    Select $(i, j) \in \hat{E}$ such that $|\text{ne}(i) \setminus \{j\}| \geq l$
    **repeat**
      Choose new $J \subseteq \text{ne}(i) \setminus \{j\}$ with $|J| = l$
      **if** $H_0 : \rho_{ij|J} = 0$ cannot be rejected **then**
        $\hat{E} \leftarrow \hat{E} \setminus \{(i, j), (j, i)\}$
      **end if**
    **until** $(i, j)$, $(j, i)$ are deleted or all $J \subseteq \text{ne}(i) \setminus \{j\}$ with $|J| = l$ are tested
  **until** all $(i, j) \in \hat{E}$ such that $|\text{ne}(i) \setminus \{j\}| \geq l$ are tested
**until** $|\text{ne}(i) \setminus \{j\}| < l$ for all $(i, j) \in \hat{E}$
**return** $\hat{\mathcal{G}}^U$

The methodology for the PC algorithm is based on what are called *faithful* distributions. A distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ following a Gaussian Markov network model $\mathcal{M}(\mathcal{G})$, where $\mathbf{\Omega} = \mathbf{\Sigma}^{-1} \in \mathbb{S}_{\mathcal{G}}^{>0}$ (Equation (2.8)), is faithful to $\mathcal{G}$ if $\omega_{ij} = 0$ for all $(i, j) \notin E$ (note that the converse always holds because $\mathbf{\Omega} \in \mathbb{S}_{\mathcal{G}}^{>0}$). For faithful distributions, if $\rho_{ij|J} = 0$, then $\rho_{ij|K} = 0$ for all $K \subseteq J$; this is what inspires iterative backward testing in Algorithm 1.

However the faithful condition is not enough for the PC algorithm's consistency, issue that has been and is currently thoroughly studied (Robins et al., 2003; Zhang and Spirtes, 2003; Kalisch and Bühlmann, 2007; Uhler et al., 2013). The PC algorithm is a versatile method since it allows to estimate either an undirected or acyclic directed Gaussian graphical model. In the latter case, the output of Algorithm 1 is oriented following a set of rules (see for example Kalisch and Bühlmann, 2007), and a graph estimate of the Markov equivalence class (Andersson et al., 1997) is returned.

## 2.5 Regularization

Certain regularization techniques that impose a zero pattern in vectors or matrices are useful alternative methods for selecting a Gaussian graphical model, as will be shown in the remainder.

A multivariate Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ belonging to a Gaussian Markov network $\mathcal{M}(\mathcal{G})$ can be learned by solving the following optimization problem

$$\arg \min_{\mathbf{\Omega} \in \mathbb{S}_{\mathcal{G}}^{\succ 0}} \operatorname{tr}(\mathbf{\Omega}\mathbf{Q}) - N \ln \det(\mathbf{\Omega}) + \lambda f(\mathbf{\Omega}),$$

where $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$, $\mathbf{Q} = \mathbf{X}^t\mathbf{X}$ with $\mathbf{X}$ the $N \times p$ sample matrix, $\lambda > 0$, tr and det are respectively the matrix trace and determinant functions, and $f$ is a sparsity inducing the penalty function. Yuan and Lin (2007a) were the first to pursue this approach, and they chose the $l_1$ norm penalization, also called *lasso* (Tibshirani, 1996), over the off-diagonal elements of $\mathbf{\Omega}$. Banerjee et al. (2008) instead included the diagonal; however, since $1/\omega_{uu} = \sigma_{uu|V\setminus\{u\}}$, this choice for the penalty favours larger values for the error variances in the regression of $X_u$ on the rest of variables (Bühlmann and van de Geer, 2011). Nonetheless, this latter estimator is the one chosen in the extensively used graphical lasso algorithm (Friedman et al., 2008).

Alternatively, if $\boldsymbol{X} = (X_1, \ldots, X_p)^t$ is a Gaussian random vector then $\beta_{ij|V\setminus\{i\}} = -\omega_{ij}/\omega_{ii}$ for each $i, j \in V$ (Anderson, 2003), which means that linear regressions can also be used for learning the Gaussian Markov network model. Specifically, let $\boldsymbol{x}_i$ denote a sample of size $N$ corresponding to variable $X_i$, $i \in V$, then $\boldsymbol{\beta}_{i|V\setminus\{i\}}$ is estimated as the solution of

$$\arg \min_{\boldsymbol{\beta} \in \mathbb{R}^{p-1}} \|\boldsymbol{x}_i - (\boldsymbol{x}_1 \cdots \boldsymbol{x}_{i-1}\boldsymbol{x}_{i+1} \cdots \boldsymbol{x}_p)\boldsymbol{\beta}\|_2^2 + \lambda f(\boldsymbol{\beta}),$$

where $(\cdots)$ denotes vector concatenation into a matrix, $\lambda > 0$, $f$ is a sparsity inducing penalty function and $\|\cdot\|_2$ is the Euclidean or $l_2$ norm. If $f$ is the $l_1$ norm, then the result is a consistent estimator of the edge set for certain choice of $\lambda$ under a rather restrictive condition (Meinshausen and Bühlmann, 2006; Zhao and Yu, 2006; Zou, 2006; Yuan and Lin, 2007b). Some variants have thus been proposed that under milder assumptions achieve model selection consistency (Meinshausen and Yu, 2009) or other attractive, so-called *oracle*, properties (van de Geer and Bühlmann, 2009); see Bühlmann and van de Geer (2011, §7) for a review. It is not known whether the conditions required for consistency of penalized likelihood estimators (Lam and Fan, 2009; Ravikumar et al., 2011) are strictly stronger than those for regression-based approaches, as some examples (Meinshausen, 2008) seem to indicate.

For selecting a Gaussian Bayesian network model $\mathcal{M}(\mathcal{G})$, the likelihood minimization problem becomes (recall Equation (2.6))

$$\arg \min_{\mathbf{W} \in \mathbb{M}_{\mathcal{G}}} \operatorname{tr}(\mathbf{W}\mathbf{W}^t\mathbf{Q}) - 2N \ln \sum_{i=1}^{p} w_{ii} + \lambda f(\mathbf{W}),$$

where $\lambda > 0$. When $f(\mathbf{W}) = |\{w_{ji} \neq 0\}|$ ($l_0$ regularization, van de Geer and Bühlmann, 2013; Aragam and Zhou, 2015), this estimator has been suggested as an alternative for the PC algorithm, in order to avoid the restrictive strong faithfulness assumption that guarantees uniform model selection consistency (Zhang and Spirtes, 2003). However, it is unclear how the assumptions of both methods are related (Uhler et al., 2013). Naturally, the penalized regression approach may also be used as an alternative (recall Equation (2.4)) if the ancestral variable ordering is known (Shojaie and Michailidis, 2010; Yu and Bien, 2017). The estimators would be those solving, for each $i \in V$, and assuming the natural order $1 \prec \cdots \prec p$ is already ancestral for notational simplicity,

$$\arg \min_{\boldsymbol{\beta} \in \mathbb{R}^{i-1}} \|\boldsymbol{x}_i - (\boldsymbol{x}_1 \cdots \boldsymbol{x}_{i-1})\boldsymbol{\beta}\|_2^2 + \lambda f(\boldsymbol{\beta}).$$

# Chapter 3

# Model selection with the PC algorithm: parameter tuning

The PC algorithm, explained in the previous chapter, is used for model selection of both Gaussian Bayesian and Markov networks. Two main parameters govern it: the statistical test over partial correlations, line 11 of Algorithm 1, and its significance level $\alpha$. Typically the usual test choice is Gaussian (Bühlmann and van de Geer, 2011), based on the Fisher's Z transformation,

$$\mathrm{Z}(t) = \frac{1}{2} \ln \left( \frac{1+t}{1-t} \right). \tag{3.1}$$

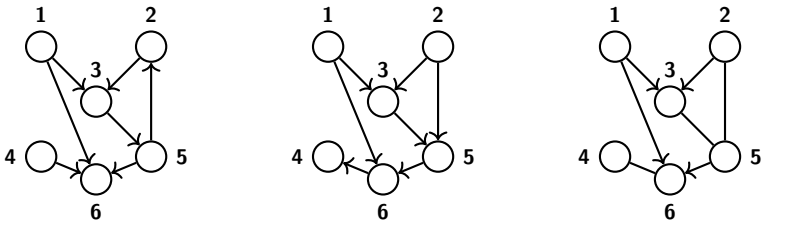
Denoting as $\hat{\rho}_{ij|J}$ the sample partial correlation coefficient, then $\sqrt{N - |J| - 3}(\mathrm{Z}(\hat{\rho}_{ij|J}) - \mathrm{Z}(\rho_{ij|J})) \sim \mathcal{N}(0,1)$ (Anderson, 2003). By contrast, $\alpha$ is usually fixed after a grid search (Kalisch and Bühlmann, 2007) or directly set by expert knowledge (Colombo and Maathuis, 2014) to a value ranging within $(0, 0.05]$. Nevertheless, such an approach for parameter selection can suffer from human bias, leading to suboptimal model selection. In this chapter a more principled approach is considered: the use of Bayesian optimisation (Shahriari et al., 2016). This is a natural choice when optimising the parameters of a function expensive to evaluate and without a closed-form expression (Snoek et al., 2012). In particular, predictive entropy search (Hernández-Lobato et al., 2014) will be used, since it achieves competitive results in a wide range of scenarios.

## 3.1 Evaluation of the output

When applying the PC algorithm to Gaussian Markov network selection, standard error rates, such as the true positive rate, can be used for evaluation. These rates simply take into account the original undirected graph $\mathcal{G} = (V, E)$, used for data simulation, and the estimated one $\hat{\mathcal{G}} = (V, \hat{E})$. Then, the edge sets $E$ and $\hat{E}$ are compared element-wise, considering the presence of an edge a positive case.

However, if selecting a Gaussian Bayesian network model $\mathcal{M}(\mathcal{G})$, an additional orientation step needs to be executed over the output of Algorithm 1, which is just an estimate of the acyclic digraph skeleton, $\hat{\mathcal{G}}^U$. Recall that two acyclic digraphs $\mathcal{G}_1$ and $\mathcal{G}_2$ yield the same Bayesian network model whenever they share skeleton and v-structures. Therefore, a mixed graph with every edge undirected except those corresponding to v-structures

uniquely characterizes a Markov equivalence class (Andersson et al., 1997). For example, the mixed graph of Figure 3.1(c) is the graph representation that corresponds to the acyclic digraphs of Figures 3.1(a) and 3.1(b). The second step of the PC algorithm is an application of several orientation rules (Meek, 1995), and it returns an estimate of the mixed graph representation of a Markov equivalence class, $[\hat{\mathcal{G}}]$. Importantly, observe that if no v-structures are estimated, then the output of this step is an undirected graph that coincides with $\hat{\mathcal{G}}^U$ and $\mathcal{M}(\hat{\mathcal{G}}) = \mathcal{M}(\hat{\mathcal{G}}^U)$ for every acyclic digraph $\hat{\mathcal{G}} \in [\hat{\mathcal{G}}]$.




(a) Acyclic digraph (b) Markov equivalent (c) Class representation

Figure 3.1: Markov equivalence among acyclic digraphs.

It is clear that when selecting Bayesian network models with the two phases of the PC algorithm, classic measures are more difficult to apply because of the mixed edge set output. The usual approach instead is to use the *structural Hamming distance* (SHD), firstly defined by Tsamardinos et al. (2006), which counts the number of operations that have to be performed over two mixed graphs, representing the respective Markov equivalence classes, so that they coincide (Algorithm 2).

**Algorithm 2** Calculation of the structural Hamming distance

**Input:** Learned mixed graph $\hat{\mathcal{G}}$ and true graph $\mathcal{G}$
**Output:** SHD$(\hat{\mathcal{G}}, \mathcal{G})$

$s \leftarrow 0$
**for** every edge $e$ different in $\hat{\mathcal{G}}$ than $\mathcal{G}$ **do**
  **if** $e$ is missing in $\hat{\mathcal{G}}$ **then**
    $s \leftarrow s + 1$
  **end if**
  **if** $e$ is extra in $\hat{\mathcal{G}}$ **then**
    $s \leftarrow s + 1$
  **end if**
  **if** $e$ has incorrect direction in $\hat{\mathcal{G}}$ **then**
    $s \leftarrow s + 1$ *// Includes edges of different type (undirected/directed)*
  **end if**
**end for**
**return** $s$

## 3.2 Parameter values

As discussed above, there are two main parameters governing the PC algorithm's output: the statistical test over partial correlations and its significance level $\alpha$. For the former,

apart from the Gaussian test based on Fisher's Z function (Equation (3.1)), there are other choices available in the literature.

Under hypothesis $H_0 : \rho_{ij|J} = 0$ the distribution of $\sqrt{N - |J| - 2}\hat{\rho}_{ij|J} / \sqrt{1 - \hat{\rho}^2_{ij|J}}$ is a Student's $t$ with $N - |J| - 2$ degrees of freedom (Anderson, 2003). Furthermore, if $\mathcal{G}^U = (V, E)$ and $\mathcal{G}^U_0 = (V, E_0)$ are the skeletons with and without edge $(i, j)$, respectively, and $T_L$ is the likelihood ratio statistic for the respective nested graphical models (also proportional to the mutual information), then $-2\ln(T_L)$ is asymptotically distributed as a $\chi^2$ distribution with $|E| - |E_0|$ degrees of freedom (Wilks, 1938). In addition to these standard tests for $H_0$, many more are available in the literature that could be considered (Edwards, 2000).

The significance level $\alpha$ is the probability of rejecting $H_0$ when it is true. In particular, for Gaussian graphical models the alternative hypothesis is $H_1 : \rho_{ij|J} \neq 0$, therefore tests are two sided (Anderson, 2003). For example, in the case of Fisher's Z transformation of $\hat{\rho}_{ij|J}$, the $p$-value $\sqrt{N - |J| - 3}|\mathrm{Z}(\hat{\rho}_{ij|J})|$ is compared against $\Phi^{-1}(1 - \alpha/2)$, where $\Phi$ is the distribution function of $\mathcal{N}(0, 1)$. If it is smaller or equal, then there is no statistical evidence to reject $H_0$ and edge $(i, j)$ is removed. Therefore, the probability of incorrectly maintaining edge $(i, j)$, also called Type I error (Bühlmann and van de Geer, 2011), is precisely $P_{H_0}(\sqrt{N - |J| - 3}|\mathrm{Z}(\hat{\rho}_{ij|J})| > \Phi^{-1}(1 - \alpha/2)) = \alpha$. For the other tests, the procedure would be analogous, but comparing with the respective asymptotic distribution instead of a standard Gaussian. In every case, the significance level $\alpha$ is typically set to a positive value smaller or equal than 0.05.

## 3.3 Experiments

As an illustration of the impact that automatic parameter tuning has over Gaussian Bayesian and Markov network model selection, the SHD measure will be optimised, parametrized by the significance level $\alpha$ and the statistical test.[1] The former will range over $10^{-5}$ and 0.1, whereas possible tests will be those available in the extensively used $R$ (R Core Team, 2020) package *bnlearn* (Scutari, 2010), which includes the three standard alternatives previously explained, as well as an improved $\chi^2$ test based on the shrinkage James-Stein estimator for the likelihood ratio $T_L$ (Hausser and Strimmer, 2009).

The simulation scope will therefore be restricted to Gaussian Bayesian network models $\mathcal{M}(\mathcal{G})$ where the acyclic digraph $\mathcal{G} = (V, E)$ will have a node set of size ranging over values 25, 50, 75 and 100, and average neighbour set size of $n \in \{2, 8\}$. The sample size of simulated data from those models will be $N \in \{10, 50, 100, 500\}$, thereby giving a total of 32 different model selection scenarios, which are moderately high-dimensional, sparse, and representative of those from relevant literature on the PC algorithm (Kalisch and Bühlmann, 2007; Colombo and Maathuis, 2014). A total of 40 replicas for each of the 32 model selection scenarios will be used, and for generating each Gaussian distribution the methodology of Kalisch and Bühlmann (2007) will be followed, implemented in the $R$ package *pcalg* (Kalisch et al., 2012). Since the acyclic digraphs will have different node size, the SHD validation measure has to be normalized with respect to the maximum edge number $p(p - 1)/2$.

[1] Code for reproducing the experiments and figures of this section is publicly available at `https://github.com/EduardoGarrido90/bopc`.

Regarding Bayesian optimisation, the predictive entropy search is averaged across 10 Monte Carlo iterations, and the transformation described by Garrido-Merchán and Hernández-Lobato (2020) is used in order to deal with the test type, which is a categorical parameter. The results are compared with a random search strategy and with the expert criterion, taken from Kalisch and Bühlmann (2007), who recommend a value of $\alpha = 0.01$ after a grid search, and use the Fisher's Z partial correlation test. At each iteration, Bayesian optimisation provides a candidate solution which corresponds to the best observation made so far, and the search is stopped after 30 evaluations of the objective, where some stability in the results can be appreciated. The *Spearmint*[2] tool was used.

The average normalized SHD results obtained of the described simulation setting are shown in Figure 3.2. We show the relative difference in log-scale with respect to the best observed result, since after normalisation the SHD differences become small because of the combinatorial network space. Therefore, the lower the values obtained, the better. We show the mean and standard deviation of this measure along the 40 replicas of the experiment, for each of the three methods compared: Bayesian optimisation (BO), random search (RS) and expert criteria (EC). We can see that EC is easily improved after only 10 iterations of BO and RS. Furthermore, BO outperforms RS providing significantly better results as more evaluations are performed. Importantly, the standard deviation of the results of BO is fairly small in the last iterations. This means that BO is very robust with respect to the different replicas of the experiments.

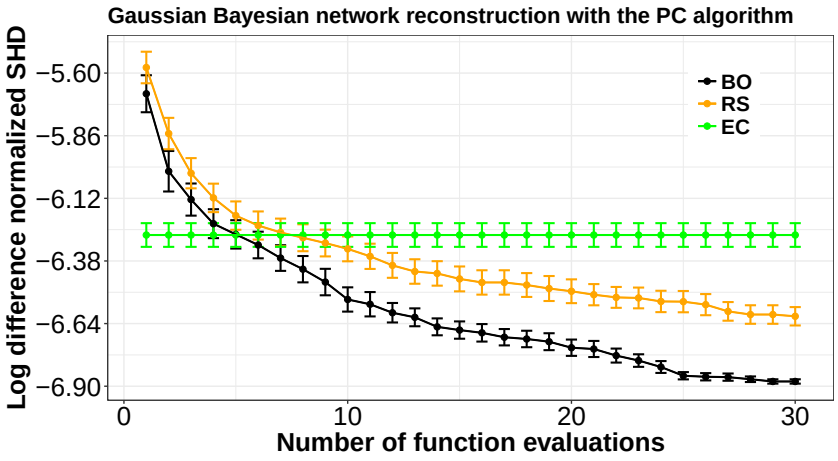



Figure 3.2: Logarithmic difference with respect to the best observed average normalized SHD obtained in 40 replicas of the 32 considered Gaussian Bayesian network models.

Since the expert criterion is outperformed, it is of interest to explore the parameter suggestions delivered by Bayesian optimisation. Figure 3.3 contains two histograms summarising the suggested parameters by Bayesian optimisation in the last iteration. Observe that the most frequently recommended test is the shrinkage $\chi^2$, while the significance level recommendation is concentrated at values lower than 0.025.

These results are very interesting from the viewpoint of model selection in graphical models. The first observation is that the optimal value obtained for the significance level is fairly close to that suggested in Kalisch and Bühlmann (2007). However, the SHD results are arguably better when using Bayesian optimisation rather than when using

[2]https://github.com/HIPS/Spearmint

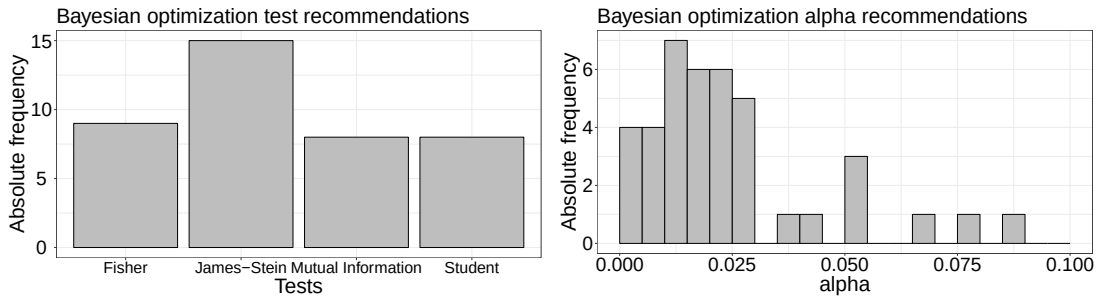



Figure 3.3: Histograms with the recommended parameters by Bayesian optimisation in the last iteration. `alpha`: Significance level; `Fisher`: Gaussian test based on the Fisher's Z transform of the partial correlation coefficient; `James-Stein`: $\chi^2$ improved test based on a shrinkage estimator of the nested likelihood ratio; `Mutual Information`: $\chi^2$ test for the likelihood ratio/mutual information of the nested models; `Student`: Student's test for the untransformed partial correlation coefficient.

parameters from an expert criterion. This may be explained by the second interesting result, namely, that the shrinkage $\chi^2$ test is suggested more often than the extensively used classic Fisher's Z partial correlation test. Therefore, in the context of sparse, high-dimensional networks, where we may have $p > N$ (such as in the above experimental set-up and the one in Kalisch and Bühlmann (2007)), it may be better to focus on the statistical test selection, rather than on carefully adjusting the significance level. In the literature, however, this is often done the other way around, and more effort is put on carefully adjusting the significance level.

# Chapter 4

# Uniform sampling of correlation matrices

There is an important remark regarding how model selection is approached in simulation experiments for Gaussian graphical models. In the previous chapter parameter tuning was addressed, but simulation methodology needs also to be examined. Recall that both the Gaussian Bayesian network model $\mathcal{M}(\mathcal{G})$ and the distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) \in \mathcal{M}(\mathcal{G})$ for each experiment replica were simulated following the methodology by Kalisch and Bühlmann (2007), extensively used for Gaussian Bayesian network models (Kalisch and Bühlmann, 2008; Colombo and Maathuis, 2014; Goudie and Mukherjee, 2016). Edges in $\mathcal{G}$ are sampled as ones in a strictly upper triangular matrix $\mathbf{U}$ (Equation (2.5)) from a Bernoulli distribution with success probability $d = n/(p-1)$, with $n$ and $p$ being neighbour and vertex set sizes in $\mathcal{G}$, respectively. The value $d$ can be thought of as an indicator of the network density: smaller $d$ values mean sparser networks. Afterwards, ones in the upper triangle of $\mathbf{U}$ are replaced by values from a uniform distribution over the $[0.1, 1]$ interval, and $\mathbf{\Sigma}^{-1} = \mathbf{U}\mathbf{U}^t$. It is easy to see that inverse covariance matrices thus generated have an increasing diagonal, thereby restricting the simulation space, see Figure 4.1. This issue could also be influencing model selection performance results for the PC algorithm.

As a first approximation to the above-described problem, in this chapter a novel Metropolis-Hastings algorithm for uniform sampling of correlation matrices is described. These are intimately related to Gaussian graphical models, both acyclic directed and undirected: letting $\mathbf{S} = \sqrt{\text{diag}(\mathbf{\Sigma})}$ be the diagonal matrix of standard deviations, then the correlation matrix is $\mathbf{R} = \mathbf{S}^{-1}\mathbf{\Sigma}\mathbf{S}^{-1}$. The set of correlation matrices is known to form a convex body $\mathcal{E}$ called *elliptope* (Laurent and Poljak, 1996), whose volume has been explicitly computed by Lewandowski et al. (2009). Uniform sampling therefore amounts to sampling with respect to the volume measure over the elliptope (Diaconis et al., 2013). Existing methods in the literature for this task are either based on vine representations of the correlation matrix (Joe, 2006; Lewandowski et al., 2009), or on spherical parametrizations of its Cholesky decomposition (Pourahmadi and Wang, 2015). By contrast, the method proposed in this chapter is intuitive, combining the upper Cholesky factorization (Eaton, 1983; Horn and Johnson, 2012) with Markov chain Monte Carlo theory (Robert and Casella, 2004), and allows for a direct application to Gaussian graphical models, as will be discussed in subsequent chapters.

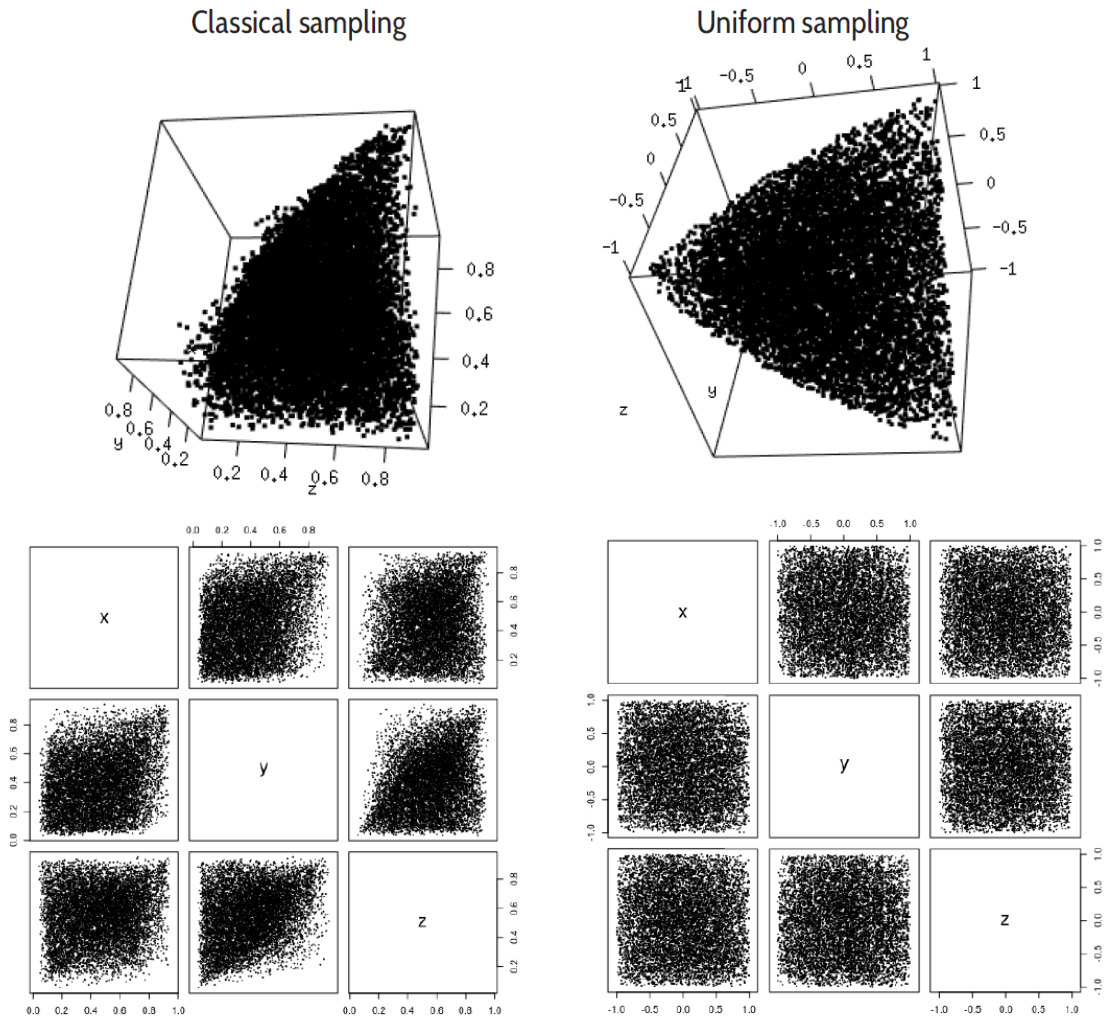



Figure 4.1: Illustration of Kalisch and Bühlmann (2007) simulation methodology for $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$ when $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ belongs to a Gaussian Bayesian network $\mathcal{M}(\mathcal{G})$ (left). By contrast, uniform sampling (right) explores the whole space. The diagonal of $\mathbf{\Omega}$ has been fixed to 1, and the skeleton of $\mathcal{G}$ complete, therefore the upper right plot depicts the set of 3-dimensional correlation matrices.

## 4.1 Cholesky parametrization and uniform sampling

Let $\mathcal{U}_1$ denote the set of upper triangular $p \times p$ matrices with positive diagonal entries and unitary row vectors. Consider now

$$\begin{array}{rccc} \Phi: & \mathcal{U}_1 & \longrightarrow & \mathbb{S}^{>0} \\ & \mathbf{U} & \longmapsto & \mathbf{U}\mathbf{U}^t, \end{array}$$

which is the upper Cholesky parametrization of the elliptope $\mathcal{E}$, that is, $\Phi(\mathcal{U}_1) = \mathcal{E}$. In order to sample uniformly from $\mathcal{E}$ via parametrization $\Phi$ (Diaconis et al., 2013), the approach is to sample from $\mathcal{U}_1$ following a density proportional to its Jacobian (Eaton, 1983)

$$\det\left(\frac{\partial\Phi(\mathbf{U})}{\partial\mathbf{U}}\right) \propto \prod_{i=1}^{p-1} u_{ii}^i, \tag{4.1}$$

where $u_{ii}$ is the $i$-th diagonal element of $\mathbf{U}$ and $u_{pp}$ has been omitted because it is equal to 1. Following such density, the induced distribution on $\mathcal{E}$ by $\Phi$ is the uniform measure (Diaconis et al., 2013).

Observe that Equation (4.1) factorises across the rows of $\mathbf{U}$, therefore each row can be sampled independently. Furthermore, $\boldsymbol{u}_i$ belongs to the positive hemisphere

$$\mathcal{S}_+^{p-i} = \{\boldsymbol{v} \in \mathbb{R}^{p-i+1} : \boldsymbol{v}^t\boldsymbol{v} = 1 \text{ and } v_1 > 0\}, \tag{4.2}$$

as it has its first $i-1$ entries equal to zero and $u_{ii} > 0$. Therefore row-wise sampling amounts to generating vectors $\boldsymbol{v} \in \mathcal{S}_+^{p-i}$ with respect to the density $f(\boldsymbol{v}) \propto v_1^i$. This sampling procedure is described in Algorithm 3.

**Algorithm 3** Uniform sampling in $\mathcal{E}$

**Input:** Sample size $N$
**Output:** Uniform sample from $\mathcal{E}$ of size $N$

**for** $n = 1, \dots, N$ **do**
  $\mathbf{U}^n \leftarrow \mathbf{0}$
  **for** $i = 1, \dots, p$ **do**
    $\boldsymbol{v} \leftarrow$ sample from $f(\boldsymbol{v}) \propto v_1^i$ on $\mathcal{S}_+^{p-i}$
    $J \leftarrow \{i, \dots, p\}$
    $\mathbf{U}_{iJ}^n \leftarrow \boldsymbol{v}$
  **end for**
**end for**
**return** $\{\Phi(\mathbf{U}^1), \dots, \Phi(\mathbf{U}^N)\}$

## 4.2 Metropolis sampling from the positive hemisphere

The step 4 of Algorithm 3 can be performed with a Metropolis scheme (Robert and Casella, 2004). In the induced Markov chain, a new state is generated as a normalized perturbation of the current vector, specifically,

$$\tilde{\boldsymbol{v}} = \frac{\boldsymbol{v} + \boldsymbol{\varepsilon}}{\|\boldsymbol{v} + \boldsymbol{\varepsilon}\|}, \tag{4.3}$$

where $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_\varepsilon^2\mathbf{I})$ is a $(p-i+1)$-dimension random vector. With such transformation, the induced proposal density $q(\tilde{\boldsymbol{v}} \mid \boldsymbol{v})$ is a projected Gaussian over the $(p-i)$-dimensional unit sphere $\mathcal{S}^{p-i}$ (Mardia and Jupp, 1999), with parameters $\boldsymbol{v}$ and $\sigma_\varepsilon^2\mathbf{I}$, whose expression is simplified in Proposition 4.2.1.

**Proposition 4.2.1.** *The expression for the proposal density is*

$$q(\tilde{\boldsymbol{v}}|\boldsymbol{v}) \propto \int_0^\infty r^{p-i} \exp\left(-\frac{1}{2}\left(r^2 - 2r\frac{\boldsymbol{v}^t\tilde{\boldsymbol{v}}}{\sigma_\varepsilon}\right)\right) dr. \tag{4.4}$$

*Proof.* Let $\boldsymbol{X} = (X_1, \dots, X_k) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and consider $\boldsymbol{Y} = \boldsymbol{X}/\|\boldsymbol{X}\|$, that is, the projection onto the unit sphere $\mathcal{S}^{k-1}$ of $\boldsymbol{X}$. Define the quantities

$$k_1 = \boldsymbol{Y}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{Y},\ k_2 = \boldsymbol{\mu}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{Y},\ k_3 = \boldsymbol{\mu}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}. \tag{4.5}$$

The density of $\boldsymbol{Y}$ on the sphere $\mathcal{S}^{k-1}$ is (Pukkila and Rao, 1988)

$$\begin{aligned} f(\boldsymbol{Y}) &\propto \exp\left(-\frac{1}{2}\left(k_1 - k_2^2 k_3^{-1}\right)\right) \int_0^\infty r^{k-1} \exp\left(-\frac{1}{2}(r - k_2 k_3^{-1/2})^2\right) dr \\ &\propto \exp\left(-\frac{1}{2}\left(k_1 - k_2^2 k_3^{-1}\right)\right) \int_0^\infty r^{k-1} \exp\left(-\frac{1}{2}(r^2 + k_2^2 k_3^{-1} - 2r k_2 k_3^{-1/2})\right) dr \quad (4.6) \\ &\propto \exp\left(-\frac{1}{2}k_1\right) \int_0^\infty r^{k-1} \exp\left(-\frac{1}{2}(r^2 - 2r k_2 k_3^{-1/2})\right) dr. \end{aligned}$$

If $k = p - i + 1$, being $p$ the correlation matrix dimension and $i$ the row number, $\boldsymbol{Y} = \tilde{\boldsymbol{v}}$ and $\boldsymbol{\mu} = \boldsymbol{v}$, Equation (4.5) simplifies to

$$\begin{aligned} k_1 &= \tilde{\boldsymbol{v}}^t (\sigma_\varepsilon^2 \mathbf{I})^{-1} \tilde{\boldsymbol{v}} = \sigma_\varepsilon^{-2} \tilde{\boldsymbol{v}}^t \tilde{\boldsymbol{v}} = \sigma_\varepsilon^{-2} \\ k_2 &= \boldsymbol{v}^t (\sigma_\varepsilon^2 \mathbf{I})^{-1} \tilde{\boldsymbol{v}} = \sigma_\varepsilon^{-2} \boldsymbol{v}^t \tilde{\boldsymbol{v}} \\ k_3 &= \boldsymbol{v}^t (\sigma_\varepsilon^2 \mathbf{I})^{-1} \boldsymbol{v} = \sigma_\varepsilon^{-2} \boldsymbol{v}^t \boldsymbol{v} = \sigma_\varepsilon^{-2}, \end{aligned}$$

where the fact that $\boldsymbol{v}^t \boldsymbol{v} = \tilde{\boldsymbol{v}}^t \tilde{\boldsymbol{v}} = 1$ has been used. Substituting in Equation (4.6), the desired result is obtained,

$$\begin{aligned} q(\tilde{\boldsymbol{v}} \mid \boldsymbol{v}) &\propto \exp\left(-\frac{1}{2}\sigma_\varepsilon^{-2}\right) \int_0^\infty r^{p-i} \exp\left(-\frac{1}{2}\left(r^2 - 2r\frac{\boldsymbol{v}^t \tilde{\boldsymbol{v}}}{\sigma_\varepsilon}\right)\right) dr \\ &\propto \int_0^\infty r^{p-i} \exp\left(-\frac{1}{2}\left(r^2 - 2r\frac{\boldsymbol{v}^t \tilde{\boldsymbol{v}}}{\sigma_\varepsilon}\right)\right) dr. \end{aligned}$$

□

The density for the proposal $q(\tilde{\boldsymbol{v}}|\boldsymbol{v})$ in Equation (4.4) is a function of the scalar product $\boldsymbol{v}^t \tilde{\boldsymbol{v}}$, therefore it is symmetric because the roles of $\boldsymbol{v}$ and $\tilde{\boldsymbol{v}}$ can be exchanged, and the Hastings correction (Robert and Casella, 2004) can be omitted from the sampling scheme. Thus, the acceptance probability at each step of the algorithm becomes

$$\min\left(1, \frac{f(\tilde{\boldsymbol{v}})}{f(\boldsymbol{v})}\right) = \min\left(1, \mathbb{I}_{\geq 0}(\tilde{v}_1)\left(\frac{\tilde{v}_1}{v_1}\right)^i\right),$$

where $\tilde{v}_1$ is the first component of the proposed vector $\tilde{\boldsymbol{v}}$ and $\mathbb{I}_{\geq 0}$ denotes the indicator function of the positive real numbers. Therefore the Metropolis sampling over $\mathcal{S}^{p-i}$ would follow the steps outlined in Algorithm 4.

## 4.3 Theoretical convergence properties

If the Metropolis chain previously constructed is irreducible and aperiodic, then it converges to its stationary distribution (see Robert and Casella, 2004, Theorem 7.4). The first condition holds because the proposal $q(\tilde{\boldsymbol{v}} \mid \boldsymbol{v})$ is strictly positive for all $\boldsymbol{v}, \tilde{\boldsymbol{v}} \in \mathcal{S}_+^{p-1}$. A sufficient condition for aperiodicity is that the probability of remaining in the same state for the next step is strictly positive, that is, $P(f(\boldsymbol{v}) \geq f(\tilde{\boldsymbol{v}})) > 0$. Expanding this,

$$P(f(\boldsymbol{v}) \geq f(\tilde{\boldsymbol{v}})) = P\left(v_1^i \geq \mathbb{I}_{\geq 0}(\tilde{v}_1)\tilde{v}_1^i\right) = P\left(v_1 \geq \tilde{v}_1, \tilde{v}_1 \geq 0\right) + P\left(\tilde{v}_1 \leq 0\right),$$

**Algorithm 4** Metropolis sampling of a vector $\boldsymbol{v}$ in $\mathcal{S}_+^{p-i}$ from $f(\boldsymbol{v}) \propto v_1^i$

**Input:** Values for $p$ and $i$; burn-in time $t_b$
**Output:** A vector $\boldsymbol{v}$ sampled from $\mathcal{S}_+^{p-i}$

$\boldsymbol{v} \leftarrow p-i+1$-dimensional observation from $\mathcal{N}(\mathbf{0}, \mathbf{I})$
$v_1 \leftarrow |v_1|$
$\boldsymbol{v} \leftarrow$ normalize $\boldsymbol{v}$
**for** $t = 0, \ldots, t_b + 1$ **do**
  $\boldsymbol{\varepsilon} \leftarrow p-i+1$-dimensional observation from $\mathcal{N}(\mathbf{0}, \sigma_\varepsilon^2 \mathbf{I})$
  $\tilde{\boldsymbol{v}} \leftarrow \boldsymbol{v} + \boldsymbol{\varepsilon}$
  $\tilde{\boldsymbol{v}} \leftarrow$ normalize $\tilde{\boldsymbol{v}}$
  $\delta \leftarrow$ random uniform observation on $[0, 1]$
  **if** $\tilde{v}_1 \geq 0$ **and** $\delta \leq (\tilde{v}_1 / v_1)^i$ **then**
    $\boldsymbol{v} \leftarrow \tilde{\boldsymbol{v}}$
  **end if**
**end for**
**return** $\boldsymbol{v}$

where, using the fact that $v_1 > 0$,

$$P(\tilde{v}_1 \leq 0) = P(\varepsilon_1 \leq 0) - P(-v_1 \leq \varepsilon_1 \leq 0) = \frac{1}{2} - \int_0^{v_1} \frac{1}{\sqrt{2\pi}\sigma_\varepsilon} \exp\left(-\frac{s^2}{2\sigma_\varepsilon^2}\right) ds > 0. \quad (4.7)$$

Therefore $P(f(\boldsymbol{v}) \geq f(\tilde{\boldsymbol{v}}))$ is strictly positive, the chain is aperiodic, and Algorithm 4 converges to $f$.

Some additional insights can be gained on the algorithm's convergence when the variance $\sigma_\varepsilon^2$ increases. From Equation (4.4), note that the density $q(\tilde{\boldsymbol{v}} \mid \boldsymbol{v})$ approaches to a constant,

$$\lim_{\sigma_\varepsilon \to \infty} q(\tilde{\boldsymbol{v}} \mid \boldsymbol{v}) = q(\tilde{\boldsymbol{v}}) \propto \int_0^\infty r^{p-i} \exp\left(-\frac{r^2}{2}\right) dr,$$

that is, to the uniform distribution over the unit sphere $\mathcal{S}^{p-i}$. The expression for such limiting proposal, which coincides with the inverse sphere volume, is

$$\lim_{\sigma_\varepsilon \to \infty} q(\tilde{\boldsymbol{v}} \mid \boldsymbol{v}) = \frac{\Gamma((p-i+1)/2)}{2\pi^{(p-i+1)/2}}, \quad (4.8)$$

where $\Gamma$ is the gamma function (Anderson, 2003). In this scenario, where the sampled values do not depend on the previous state, the resulting sampling algorithm is called *independent* Metropolis and satisfies desirable convergence properties, in particular, the chain is uniformly ergodic (see Robert and Casella, 2004, Theorem 7.8).

## 4.4 Experiments

For reproducibility, the scripts used for generating the data and figures described throughout this section are publicly available at `https://github.com/irenecrsn/rcor`. Also, an implementation of Algorithm 3 can be found in the *R* (R Core Team, 2020) package *gmat*[1], function *chol_mh*.

[1]CRAN latest release at `https://CRAN.R-project.org/package=gmat`, with version under development available at `https://github.com/irenecrsn/gmat`.

### 4.4.1 Empirical convergence monitoring

The convergence of Algorithm 4 can also be empirically monitored. As an illustration, the focus of this section is on the high-dimensional case, $p = 1000$, and the acceptance ratio, that is, the percentage of times the proposed value has been accepted. This latter quantity can therefore be thought of as an approximation for $P(f(\boldsymbol{v}) \leq f(\tilde{\boldsymbol{v}}))$. In Figure 4.2 the acceptance ratio is depicted as a function of the row number $i$, which influences the form of target density $f(\boldsymbol{v}) \propto v_1^i$, and, complementarily, of the perturbation variance $\sigma_\varepsilon^2$.

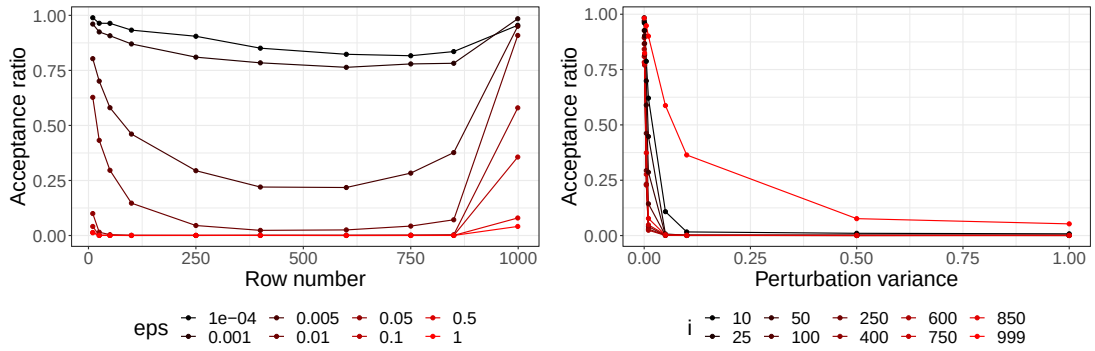



Figure 4.2: Acceptance ratio as a function of the row number (left) and the perturbation variance (right). `eps`: perturbation variance $\sigma_\varepsilon^2$; `i`: row number.

It is noticeable how as $\sigma_\varepsilon^2$ increases the proposed value is rejected more often, which could be already expected by looking at Equation (4.7) above, where the second term goes to zero as $\sigma_\varepsilon^2$ increases, yielding $\lim_{\sigma_\varepsilon^2 \to \infty} P(f(\boldsymbol{v}) \leq f(\tilde{\boldsymbol{v}})) \leq 1/2$. Furthermore, recall that as $\sigma_\varepsilon^2$ increases the proposal distribution is more similar to the uniform density on the $(p-i)$-dimensional sphere (Equation (4.8)), which also hints the higher rejection rate since only those values in the positive hemisphere are accepted. The row number $i$ also has a significant influence on the acceptance ratio, because as it increases the target density $f(\boldsymbol{v}) \propto v_1^i$ approaches a delta function, and the dimensionality of $\boldsymbol{v} \in \mathcal{S}^{p-i}$ decreases.

Because all of the above, it is reasonable to conclude that the larger $i$ is, the smaller $\sigma_\varepsilon^2$ should be for achieving a high acceptance ratio, since new candidate states should be, with high probability, very close to the current state in order to be accepted. This is further illustrated in Figure 4.3, where the contour lines of the acceptance ratio surface as a function of $\sigma_\varepsilon^2$ and the row number $i$ are depicted. Observe that small values for $\sigma_\varepsilon$ always lead to high acceptance ratios, which, although being desirable in the delta situation explained above, can also be a sign of slow convergence in different scenarios. By contrast, a low acceptance ratio can be expected when approaching to a delta in moderately high dimensions, as is the case for row numbers approximately between 250 and 750, where almost all except extremely small values for $\sigma_\epsilon^2$ yield low acceptance ratios. As a conclusion, in Monte Carlo methods there is no recipe for assuring fast convergence (Robert and Casella, 2004), even when the chain is theoretically assured to converge. Thus, parameter selection can be approached by monitoring procedures such as the ones illustrated throughout this section, or automatic methodologies such as Bayesian optimisation, as in the previous chapter.

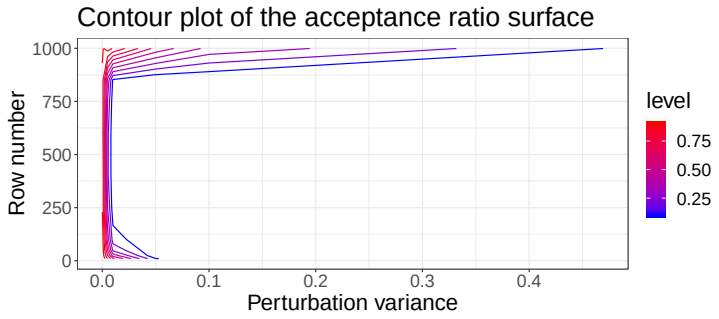



Figure 4.3: Contour lines of the acceptance ratio surface. `level`: magnitude of the acceptance ratio.

### 4.4.2 Comparative analysis

In this section Algorithm 3 will be compared against the existing state-of-the-art alternatives: the c-vine and onion methods from Lewandowski et al. (2009) and the spherical or polar parametrization by Pourahmadi and Wang (2015).

Since all of them sample from the same uniform distribution over the elliptope, some results are expected to be similar across the methods. As an illustration, recall that a closed formula for the elliptope's volume $\text{vol}(\mathcal{E})$ was computed by Lewandowski et al. (2009). Therefore, for a sample $\mathbf{R}_1, \ldots, \mathbf{R}_N$ from $\mathcal{E}$, the expected number of matrices $m_{\mathcal{Q}}$ in a hypercube $\mathcal{Q}$ of edge length $d$ inside $\mathcal{E}$ can be obtained as follows,

$$m_{\mathcal{Q}} = \mathbb{E}\left(\sum_{n=1}^{N} \mathbb{I}_{\mathcal{Q}}(\mathbf{R}_n)\right) = N\frac{\text{vol}(\mathcal{Q})}{\text{vol}(\mathcal{E})} = \frac{Nd^p}{\text{vol}(\mathcal{E})}.$$

Table 4.1 shows the empirical $\mathcal{Q}$ volume obtained by each method for the cube centred at the origin of edge length 0.2 inside an elliptope of $p = 3$ dimensions over 50 repetitions of $N = 5000$ samples from $\mathcal{E}$. The expected theoretical volume is $m_{\mathcal{Q}} \approx 16.21$, and it can be seen that all methods yield an empirical volume close to that magnitude.

| Uniform sampling method for $\mathcal{E}$ | Empirical volume for $\mathcal{Q}$ |
|---|---|
| `chol` | 16.36 |
| `c-vine` | 16.34 |
| `onion` | 16.02 |
| `polar` | 16.00 |

Table 4.1: Empirical volumes of $\mathcal{Q}$ for each method. `chol`: our proposal; `c-vine`, `onion`: methods by Lewandowski et al. (2009); `polar`: method by Pourahmadi and Wang (2015).

Now methods will be compared in terms of computational performance. Specifically, each method will generate 5000 correlation matrices of dimension $p = 10, 20, \ldots, 100$. Methods from Lewandowski et al. (2009) are available via function *genPositiveDefMat* from the *R* package *clusterGeneration*[2], whereas for the method by Pourahmadi and Wang (2015), the *R* package *randcorr* (Makalic and Schmidt, 2020) has been used. The

[2] `https://CRAN.R-project.org/package=clusterGeneration`

experiment has been executed on a machine equipped with Intel Core i7-5820k, 3.30 GHz×12 and 16 GB of RAM.

The time experiment results are shown in Figure 4.4. Notice that the proposed Metropolis method is the fastest. This can be explained as it takes advantage of the direct representation provided by the Cholesky factorization, as well as the simple form of the target distribution and proposed values on each iteration.

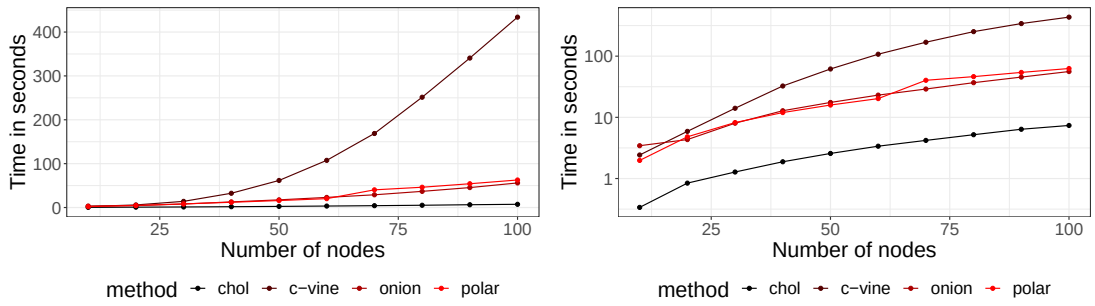



Figure 4.4: Execution time of available methods for uniform sampling of correlation matrices, both in linear (left) and logarithmic (right) scale. `chol`: our proposal; `c-vine`, `onion`: methods by Lewandowski et al. (2009); `polar`: method by Pourahmadi and Wang (2015), as implemented by Makalic and Schmidt (2020).

# Chapter 5

# On Gaussian graphical model simulation

Gaussian graphical model selection methods are often validated on synthetic models, usually obtained by generating a symmetric positive definite matrix compatible with some, possibly also synthetic, graph. Gaussian Bayesian networks were analysed at the end of Chapter 3, where it was explained how the commonly used simulation methodology of Kalisch and Bühlmann (2007) yields an inverse covariance matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$ which has a diagonal with increasing values. In the case of Gaussian Markov networks, where the zeros are present directly in $\mathbf{\Omega}$ instead of its Cholesky decomposition, diagonally dominant matrices are employed (Lin et al., 2009; Arvaniti and Claassen, 2014; Stojkovic et al., 2017) to guarantee the positive definiteness of $\mathbf{\Omega}$. In this chapter, the limitations of these approaches when validating model selection methods will be discussed, as well as some alternatives will be provided. The uniform sampling Metropolis method from the previous chapter will be adapted for uniform sampling of chordal Gaussian Markov networks or Gaussian Bayesian networks with no v-structures. The R scripts for replicating the experiments described throughout this chapter are available online at `https://github.com/irenecrsn/ggmsim`.

The results may also be applied to another Gaussian graphical model: the covariance graph (Cox and Wermuth, 1993; Kauermann, 1996)

$$\mathcal{M}(\mathcal{G}) = \{\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) : \mathbf{\Sigma} \in \mathbb{S}_{\mathcal{G}}^{>0}\}, \tag{5.1}$$

where $\mathcal{G} = (V, E)$ is an undirected graph. Observe that this model is the same as a Gaussian Markov network (Equation (2.8)), but with the graphical constraints over the covariance matrix directly instead of its inverse. Therefore, marginal independences are captured instead of conditional ones, since for each random vector $\boldsymbol{X}$ following a covariance graph distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$,

$$(i, j) \notin E \implies \sigma_{ij} = 0 \implies X_i \perp\!\!\!\perp X_j.$$

A covariance graph $\mathcal{M}(\mathcal{G})$ and a Gaussian Markov network model $\mathcal{M}(\mathcal{H})$ coincide if and only if $\mathcal{G}$ and $\mathcal{H}$ consist of the same complete, disconnected sub-graphs (Wermuth and Sadeghi, 2012; Drton and Richardson, 2008; Jensen, 1988). This implies in practice that the statistical independences in most multivariate Gaussian distributions can be represented only by one of the two models.

## 5.1 Classical simulation methodologies

When a symmetric matrix $\mathbf{M}$ satisfies that $|m_{ii}| > \sum_{j \neq i} |m_{ij}|$ for each $i \in \{1, \ldots, p\}$, commonly called *diagonal dominance*, then $\mathbf{M}$ is guaranteed to be positive definite (Horn and Johnson, 2012). Thus a simple method to generate a matrix in $\mathbb{S}_{\mathcal{G}}^{>0}$ consists in generating a random symmetric matrix in $\mathbb{M}_{\mathcal{G}}$ and then choosing diagonal elements so the final matrix is diagonally dominant. This procedure is outlined in Algorithm 5. The usual approach for generating the initial matrix in line 1 is to use independent and identically distributed non-zero entries.

**Algorithm 5** Simulation of a diagonally dominant matrix in $\mathbb{S}_{\mathcal{G}}^{>0}$

**Input:** Undirected graph $\mathcal{G} = (V, E)$ with $|V| = p$
**Output:** Matrix belonging to $\mathbb{S}_{\mathcal{G}}^{>0}$
1: $\mathbf{M} \leftarrow$ symmetric matrix in $\mathbb{M}_{\mathcal{G}}$
2: **for** $i = 1, \ldots, p$ **do**
3: $\quad m_{ii} \leftarrow \sum_{j \neq i} |m_{ij}|+$ random positive perturbation
4: **end for**
5: **return** $\mathbf{M}$

Diagonal dominance has been extensively used in the literature mainly due to its simplicity and the ability to control the generated matrix singularity. In particular, it is possible to control its minimum eigenvalue (Honorio et al., 2012) as follows. Let $\mathcal{G}$ be an undirected graph, $\mathbf{M}$ a symmetric matrix in $\mathbb{M}_{\mathcal{G}}$, and $\varepsilon > 0$ the desired lower-bound on the eigenvalues. If $\lambda_{min}$ is the minimum eigenvalue of $\mathbf{M}$, then $\mathbf{M} + (\lambda_{min}^{-} + \varepsilon)\mathbf{I}$ belongs to $\mathbb{S}_{\mathcal{G}}^{>0}$ and has eigenvalues greater or equal to $\epsilon$, where $\lambda_{min}^{-} = \max(-\lambda_{min}, 0)$ denotes the negative part of $\lambda_{min}$ and $\mathbf{I}$ is the identity matrix. Similarly, one can control the condition number, with respect to the Frobenious norm, of the generated matrix (Cai et al., 2011): if $\kappa_0 > 1$ is the desired condition number and $\lambda_{max} > 0$ the maximum eigenvalue of $\mathbf{M}$, then

$$\mathbf{M} + \frac{\lambda_{max} - \kappa_0 \lambda_{min}}{\kappa_0 - 1} \mathbf{I}$$

belongs to $\mathbb{S}_{\mathcal{G}}^{>0}$ and has condition number equal to $\kappa_0$. The main drawback of diagonally dominant inverse covariance matrices is that off-diagonal elements, often interpreted as link strengths, are extremely small with respect to the diagonal entries and thus model selection becomes a challenge, thereby compromising the synthetic validation (Schäfer and Strimmer, 2005a,b; Krämer et al., 2009; Cai et al., 2011).

On the other hand, the simulation methodology proposed by Kalisch and Bühlmann (2007), extensively used for Gaussian Bayesian network models (Kalisch and Bühlmann, 2008; Colombo and Maathuis, 2014; Goudie and Mukherjee, 2016), also yields matrices satisfying a pattern. In particular, assume that the acyclic digraph $\mathcal{G} = (V, E)$ has already been simulated, with each edge drawn from a Bernoulli with success probability $d$ and an arbitrary ancestral order $\prec$. Then, a matrix $\mathbf{U}$ is filled with zeros for absent edges of $\mathcal{G}$, whereas those corresponding to the diagonal or edges are replaced by values from a uniform distribution over the $[0.1, 1]$ interval, following Algorithm 6. The inverse covariance matrix is thus obtained as $\mathbf{\Omega} = \mathbf{\Sigma}^{-1} = \mathbf{U}\mathbf{U}^t$, that is, the error variables of Equation (2.4) are assumed to have unit variance and therefore $\mathbf{D} = \mathbf{I}$ in Equation (2.5). Matrix $\mathbf{\Omega}$ has an interesting property regarding its diagonal: it consists of increasing

values, which is proved for completeness in Proposition 5.1.1. However, as was shown in Figure 4.1 for the case of correlation matrices, this methodology leaves a significant region of the space unexplored.

**Algorithm 6** Simulation of matrix in $\mathbb{M}_{\mathcal{G}}$

**Input:** Acyclic digraph $\mathcal{G} = (V, E)$ with $|V| = p$
**Output:** Matrix belonging to $\mathbb{M}_{\mathcal{G}}$

```
U ← 0
for i = 1, ..., p do
    for j ∈ pa(i) do
        u_ji ← random uniform observation over [0.1, 1]
    end for
    u_ii ← random uniform observation over [0.1, 1]
end for
return U
```

**Proposition 5.1.1.** *Let* $\mathbf{U}$ *be a matrix output by Algorithm 6, and denote as* $\tau$ *the permutation associated with the ancestral order* $\prec$ *of* $\mathcal{G}$*. Then matrix* $\tau(\mathbf{\Omega}) = \tau(\mathbf{U})\tau(\mathbf{U})^t$ *is expected to have an increasing diagonal, that is,* $\mathbb{E}(\omega_{\tau(k)\tau(k)}) < \mathbb{E}(\omega_{\tau(j)\tau(j)})$ *for all* $k \prec j$*.*

*Proof.* Denote as $\mathrm{ch}(j) = \{i \in V : (j, i) \in E\}$ the children set of vertex $j \in V$. Observe that $\tau(\mathbf{U})$ is upper triangular, and there are $\tau(j) - 1$ candidate parent nodes for $\tau(j)$, whereas the candidate children nodes are $p - \tau(j)$. Also note that the $j$-th row of $\tau\mathbf{U}$ contains information about $\mathrm{ch}(j)$, whereas column $i$ informs about $\mathrm{pa}(i)$. Non-zero entries of $\mathbf{U}$ are sampled from a uniform distribution over the same interval, which means that for every $j \in \{1, \ldots, p\}$ and $i \in \mathrm{ch}(j)$, $\mathbb{E}(u_{jj}) = \mathbb{E}(u_{ji}) = C$. Therefore the expected values for $\mathbf{\Omega}$ diagonal entries are

$$\mathbb{E}(\omega_{jj}) = \sum_{i \in \mathbb{E}(\mathrm{ch}(j))} \mathbb{E}(u_{ji})^2 + \mathbb{E}(u_{jj})^2 = d(p - \tau(k))C^2 + C^2,$$

and thus if $k \prec j$

$$\mathbb{E}(\omega_{\tau(k)\tau(k)}) = d(p - \tau(k))C^2 + C^2 < d(p - \tau(j))C^2 + C^2 = \mathbb{E}(\omega_{\tau(j)\tau(j)}).$$

□

## 5.2 A partial orthogonalization simulation method

First, as noted by several authors, matrices sampled with Algorithm 5 suffer from what could be called *weak link strength*, referring to low absolute value of the off-diagonal entries with respect to those in the diagonal (Schäfer and Strimmer, 2005a,b; Krämer et al., 2009; Cai et al., 2011). In order to overcome such issue, an alternative method which does not rely on diagonal dominance will be subsequently described. It is based on the following simple idea. Consider an arbitrary full rank matrix $\mathbf{Q}$, then its product $\mathbf{QQ}^t$, which is always positive definite and symmetric, belongs to $\mathbb{S}_{\mathcal{G}}^{>0}$, with $\mathcal{G} = (V, E)$ an undirected graph, if and only if $\boldsymbol{q}_i^t \boldsymbol{q}_j = 0$ for every $(i, j) \notin E$, where $\boldsymbol{q}_i$ and $\boldsymbol{q}_j$ are the

$i$-th and $j$-th rows of $\mathbf{Q}$, respectively. That is, if for each $(i,j) \in E$, the $i$-th and $j$-th rows of $\mathbf{Q}$ are orthogonal.

Thus, given an undirected graph $\mathcal{G} = (V, E)$ and an arbitrary matrix $\mathbf{Q}$ of full rank, in order to obtain another in $\mathbb{S}_{\mathcal{G}}^{>0}$ a simple procedure is to iteratively orthogonalise the rows of $\mathbf{QQ}^t$ corresponding to elements in $E$. Recall that the set of positive definite matrices with ones along the diagonal, or correlation matrices, is called elliptope (Laurent and Poljak, 1996) and denoted as $\mathcal{E}$. Since a Gaussian distribution can be equivalently parametrized by its (inverse) covariance matrix or the scaled (partial) correlation matrix, instead of simulating from $\mathbb{S}_{\mathcal{G}}^{>0}$, one could sample directly from $\mathcal{E}_{\mathcal{G}}$, the set of (partial) correlation matrices complying with a given undirected graph $\mathcal{G}$. For the diagonal dominance method of Algorithm 5, the output would just be normalised, retaining a dominant diagonal. Alternatively, the pseudocode for partially orthogonalising the rows of $\mathbf{Q}$ and obtaining a matrix in $\mathcal{E}_{\mathcal{G}}$ is described procedure can be found in Algorithm 7.

**Algorithm 7** Simulation of a matrix in $\mathcal{E}_{\mathcal{G}}$ using partial orthogonalization

**Input:** Undirected graph $\mathcal{G} = (V, E)$ with $|V| = p$
**Output:** Matrix belonging to $\mathcal{E}_{\mathcal{G}}$

1: $\mathbf{Q} \leftarrow$ arbitrary matrix of full rank
2: **for** $i = 1, \ldots, p$ **do**
3: orthogonalize $\boldsymbol{q}_i$ with respect to the span of $\{\boldsymbol{q}_j : (i,j) \notin E \text{ and } j < i\}$
4: normalize $\boldsymbol{q}_i$
5: **end for**
6: **return** $\mathbf{QQ}^t$

### 5.2.1 Numerical and computational properties

If the entries of matrix $\mathbf{Q}$ are initially simulated as independent and identically distributed centred sub-Gaussian, then its condition number $\kappa(\mathbf{Q}) \geq p$ with high probability (Rudelson and Vershynin, 2009). Therefore, in such case the condition number of the matrices $\mathbf{QQ}^t$ returned by Algorithm 7 will satisfy $\kappa(\mathbf{QQ}^t) \geq p^2$ as the graph structure becomes denser, as shown in Figure 5.1.

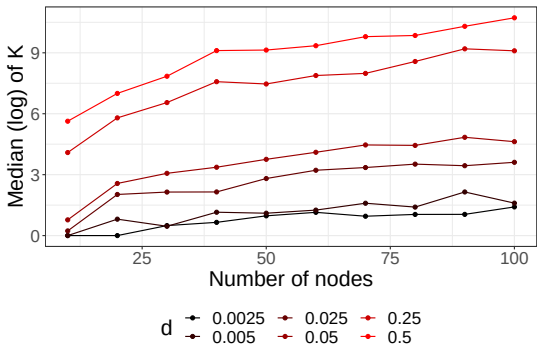



Figure 5.1: Logarithm of the condition number median. `K`: condition number of the matrix.

The step 3 of Algorithm 7 can be implemented in several ways, for example by using a modified Gram-Schmidt orthogonalization procedure, as reflected in Algorithm 8. The loop in line 1 constructs a set of orthogonal vectors $\tilde{\boldsymbol{q}}_j$ which span the same subspace than the original rows in $\{\boldsymbol{q}_j : j \notin \text{ne}(i) \text{ and } j < i\}$. This orthogonal base is later used in the loop at line 9 for ensuring that $\boldsymbol{q}_i$ is jointly orthogonal to all the vectors.

**Algorithm 8** Modified Gram-Schmidt orthogonalization for a row $i$ of $\mathbf{Q}$

**Input:** Undirected graph $\mathcal{G}$, row number $i$ and matrix $\mathbf{Q}$
**Output:** Row $\boldsymbol{q}_i$ orthogonal to $\{\boldsymbol{q}_j : j \notin \text{ne}(i) \text{ and } j < i\}$

1: **for** $j = 1, \ldots, i-1$ **do**
2:  **if** $j \notin \text{ne}(i)$ **then**
3:  $\tilde{\boldsymbol{q}}_j \leftarrow \boldsymbol{q}_j$
4:  **for** $k < j$ and $k \notin \text{ne}(i)$ **do**
5:  orthogonalise $\tilde{\boldsymbol{q}}_j$ with respect to $\tilde{\boldsymbol{q}}_k$
6:  **end for**
7:  **end if**
8: **end for**
9: **for** $j = 1, \ldots, i-1$ **do**
10:  **if** $j \notin \text{ne}(i)$ **then**
11:  orthogonalise $\boldsymbol{q}_i$ with respect to $\tilde{\boldsymbol{q}}_j$
12:  **end if**
13: **end for**
14: **return** $\boldsymbol{q}_i$

The computational complexity of Algorithm 8 is mainly given by the loop in line 1, which in the worst case scenario is $O(i^2 p)$ because $|\text{ne}(i)| \leq i - 1$. Therefore, the overall worst case complexity of Algorithm 7 becomes $O(p^4)$ when using a modified Gram-Schmidt procedure. Figure 5.2 contains the execution time of both Algorithms 5 and 7, when sampling 5000 matrices for the each $d$ value. This experiment has been executed on a machine equipped with Intel Core i7-5820k, 3.30 GHz×12 and 64 GB of RAM, and both methods are available in the *R* (R Core Team, 2020) package *gmat*[1], functions *diagdom* and *port*, respectively.

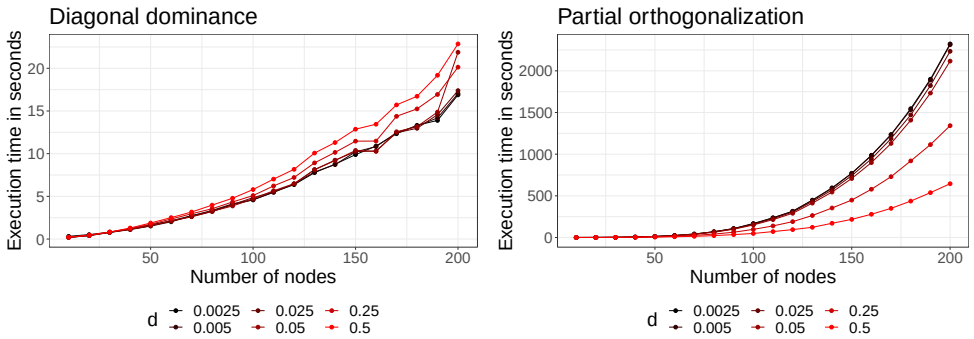



Figure 5.2: Execution time to simulate 5000 matrices.

[1]CRAN latest release at `https://CRAN.R-project.org/package=gmat`, with version under development available at `https://github.com/irenecrsn/gmat`.

It can be observed that the diagonal dominance method is two orders of magnitude faster than the proposed partial orthogonalization method, which is somewhat expected given its relative simplicity. Furthermore, the computational cost of the partial orthogonalization method depends on the structure density $d$. For small values of $d$ the undirected graph contains a lot of disconnected vertices and thus the loop in line 1 of Algorithm 8 is repeated for many matrix rows, being closer to the worst case scenario of $\mathcal{O}(p^4)$.

### 5.2.2 Link strength comparison

In this section, 10 random Erdös-Rényi (Erdös and Rényi, 1959) undirected graphs $\mathcal{G}_1, \ldots, \mathcal{G}_{10}$ are generated for different values of the vertex set size $p$ and density $d$. Afterwards, 10 matrices in $\mathbb{S}^{>0}_{\mathcal{G}_n}$ are sampled using both Algorithms 5 and 7, for $n \in \{1, \ldots, 10\}$. Thus, in total 100 matrices are sampled. Both methods need to generate an arbitrary matrix as an initial step, and for that independent and identically distributed entries are sampled from a uniform distribution over $[0, 1]$. Since the literature traditionally mentions a concern over link strength in the generated matrix $\mathbf{M}$, as previously mentioned, the average of the maximum ratio $R = \max_{j \neq i} |m_{ij}|/|m_{ii}|$ is computed, with results shown in Figure 5.3. Both methods show a comparable behaviour, with the main difference being that the partial orthogonalization method effectively avoids zero values for average $R$ even in dense settings, which are usually found in applications (Krämer et al., 2009). Also for every value of $d$, the partial orthogonalization method yields a higher average $R$ than diagonally dominant matrices.

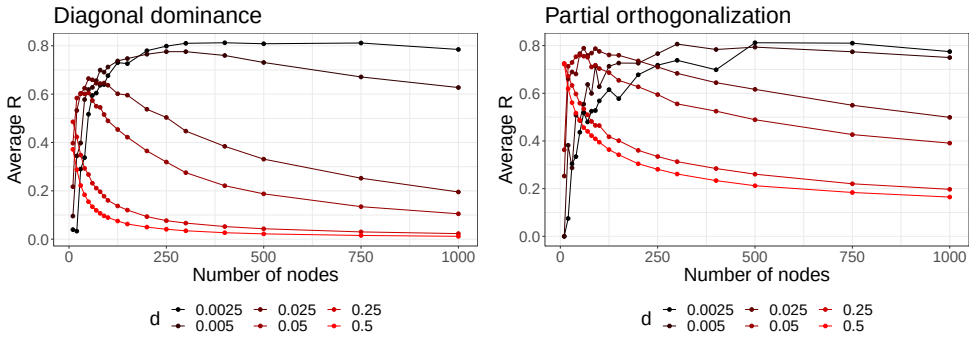



Figure 5.3: Average of $R$ as a function of $p$ for different graph densities $d$.

The above mentioned conclusion is complementarily drawn from Figure 5.4, where the performance of both methods is jointly plotted for the two extreme density values: $d = 0.0025$ (very sparse) and $d = 0.5$ (very dense), with a shade indicating the standard error of the mean. In the sparse scenario both methods perform reasonably good, however in the dense case the diagonal dominance method performance is deeply affected early, being almost zero for $p > 125$, approximately. The partial orthogonalization method, however, manages to maintain an arguably reasonable value for the average ratio.

## 5.3 Uniform sampling for chordal models

Recall that if $\mathcal{G} = (V, E)$ is a chordal undirected graph, then there exists an orientation $\mathcal{G}^{\rightarrow} = (V, E^{\rightarrow})$ that has no v-structures and with ancestral order $\prec$ equal to the perfect

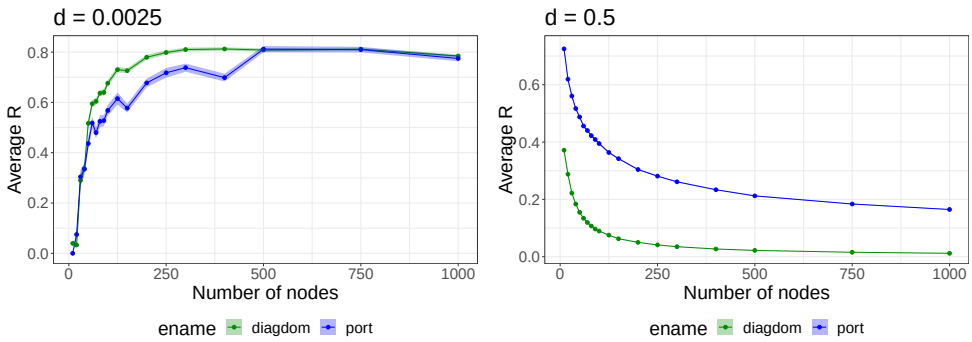



Figure 5.4: The average of $R$ as a function of $p$, for the two extreme values of $d$: very sparse matrices (left) and very dense matrices (right). `diagdom`: Diagonal dominance method; `port`: Partial orthogonalization method; `ename`: Experiment name.

ordering of $\mathcal{G}$, and the respective graphical models $\mathcal{M}(\mathcal{G})$ and $\mathcal{M}(\mathcal{G}^{\rightarrow})$ coincide (Chapter 2). In the Gaussian case this implies in particular that $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) \in \mathcal{M}(\mathcal{G})$ if and only if $\mathbf{\Omega} = \mathbf{\Sigma}^{-1} \in \mathbb{S}_{\mathcal{G}}^{>0}$ and $\mathbf{\Omega} = \mathbf{W}\mathbf{W}^t$ with $\mathbf{W} \in \mathbb{M}_{\mathcal{G}^{\rightarrow}}$ (Wermuth, 1980). Therefore the upper Cholesky factorisation may be used to parametrize $\mathcal{E}_{\mathcal{G}}$, similarly to what was done in Chapter 4. Specifically, denoting, for an acyclic digraph $\mathcal{G}$, as $\mathbb{M}_{\mathcal{G}}^1$ the subset of $\mathbb{M}_{\mathcal{G}}$ where rows are unitary, then the parametrization is

$$\begin{array}{rccc} \Phi: & \mathbb{M}_{\mathcal{G}}^1 & \rightarrow & \mathcal{E}_{\mathcal{G}} \\ & \mathbf{W} & \mapsto & \mathbf{W}\mathbf{W}^t. \end{array}$$

The Jacobian of $\Phi$ has been obtained by Roverato (2000) (note the similarity with Equation (4.1))

$$\det\left(\frac{\partial \Phi(\mathbf{W})}{\partial \mathbf{W}}\right) \propto \prod_{i=1}^{p} w_{ii}^{|\mathrm{pa}(i)|+1}, \tag{5.2}$$

where $\mathrm{pa}(i)$ is the parent set of vertex $i$ in $\mathcal{G}^{\rightarrow}$. As in Chapter 4, it factorises across rows $\boldsymbol{w}_i$ of $\mathbf{W}$, thus they can be sampled independently (Algorithm 9). Specifically, $\boldsymbol{w}_i$ contains non-zero entries corresponding to children of $i$ in $\mathcal{G}^{\rightarrow}$, $\mathrm{ch}(i) = \{j : (i,j) \in E^{\rightarrow}\}$. Therefore, $\boldsymbol{w}_i \in \mathcal{S}_+^{|\mathrm{ch}(i)|}$ (Equation (4.2)) and Algorithm 4 may be used, substituting $p-i$ with $|\mathrm{ch}(i)|$. This procedure is outlined in Algorithm 9.

**Algorithm 9** Uniform sampling in $\mathcal{E}_{\mathcal{G}}$ for a chordal graph $\mathcal{G}$

**Input:** Chordal graph $\mathcal{G} = (V, E)$ with $|V| = p$
**Output:** A matrix uniformly sampled from $\mathcal{E}_{\mathcal{G}}$

$\mathcal{G}^{\rightarrow} \leftarrow$ orientation of $\mathcal{G}$ with no v-structures
$\mathbf{W} \leftarrow \mathbf{0}$
**for** $i = 1, \ldots, p$ **do**
  $\boldsymbol{v} \leftarrow$ sample from $f(\boldsymbol{v}) \propto v_1^{|\mathrm{pa}(i)+1|}$ on $\mathcal{S}_+^{|\mathrm{ch}(i)|}$ // $\mathrm{pa}(i)$ *and* $\mathrm{ch}(i)$ *computed in* $\mathcal{G}^{\rightarrow}$
  $\mathbf{W}_{i\,\mathrm{ch}(i)} \leftarrow \boldsymbol{v}_{-1}$ // *Vector* $v$ *except its first entry*
  $w_{ii} \leftarrow v_1$
**end for**
**return** $\Phi(\mathbf{W})$

Algorithm 9 allows to sample uniformly from $\mathcal{E}_\mathcal{G}$ when $\mathcal{G}$ is a chordal graph or, equivalently, an acyclic digraph with no v-structures. When the undirected graph $\mathcal{G} = (V, E)$ is not chordal it is not possible to obtain an orientation with no v-structures. Furthermore, if applying Algorithm 9 to a triangulation $\overline{\mathcal{G}} = (V, \overline{E})$ of $\mathcal{G}$ then the resulting matrix will have more non-zero entries than desired, specifically those corresponding to edges in $\overline{E} \setminus E$. As such, a complementary method to partial orthogonalization for sampling from $\mathcal{E}_\mathcal{G}$ for a general graph $\mathcal{G}$ could be an hybrid method combining both approaches: firstly sample a factor $\mathbf{W} \in \mathbb{M}^1_{\overline{\mathcal{G}}}$ using Algorithm 9 for triangulation $\overline{\mathcal{G}}$, and then partially orthogonalize it so that $\Phi(\mathbf{W})$ belongs to $\mathcal{E}_\mathcal{G}$. This method is detailed in Algorithm 10.

**Algorithm 10** Sampling from $\mathcal{E}_\mathcal{G}$ for a general undirected graph $\mathcal{G}$

**Input:** Undirected graph $\mathcal{G} = (V, E)$ with $|V| = p$
**Output:** Matrix belonging to $\mathcal{E}_\mathcal{G}$

$\mathcal{G}^{\rightarrow} \leftarrow$ orientation of a triangulation of $\mathcal{G}$, with no v-structures
$\mathbf{W} \leftarrow \mathbf{0}$
**for** $i = 1, \ldots, p$ **do**
  $\boldsymbol{v} \leftarrow$ sample from $f(\boldsymbol{v}) \propto v_1^{|\mathrm{pa}(i)+1|}$ on $\mathcal{S}_+^{|\mathrm{ch}(i)|}$ // $\mathrm{pa}(i)$ *and* $\mathrm{ch}(i)$ *computed in* $\mathcal{G}^{\rightarrow}$
  $\mathbf{W}_{i\,\mathrm{ch}(i)} \leftarrow \boldsymbol{v}_{-1}$ // *Vector* $v$ *except its first entry*
  $w_{ii} \leftarrow v_1$
**end for**
**for** $i = 1, \ldots, p$ **do**
  orthogonalize $\boldsymbol{w}_i$ with respect to the span of $\{\boldsymbol{w}_j$ s.t. $(i, j) \notin E$ and $j < i\}$
  normalize $\boldsymbol{w}_i$
**end for**
**return** $\Phi(\mathbf{W})$

The two methods described in this section are also available in the *R* package *gmat*, functions *chol_mh* (Algorithm 9) and *port_chol* (Algorithm 10). Recall that at the end of Chapter 3 a visual exploratory analysis was performed and in Figure 4.1 it was shown how Algorithm 6 failed to sample from a significant space region when compared to uniform sampling (Algorithm 9 for acyclic digraphs with no v-structures). A similar analysis can be performed in the undirected case for chordal graphs and Algorithms 5 and 7, as follows.

### 5.3.1 Comparative analysis: Three variables

Consider the simple chordal graph $\mathcal{G} = (V = \{1, 2, 3\}, E = \{(1, 2), (2, 1), (2, 3), (3, 2)\})$, depicted in Figure 5.5.

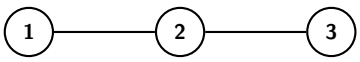



Figure 5.5: Chordal undirected graph with three variables.

A sample of size $N = 5000$ is obtained from $\mathcal{E}_\mathcal{G}$ using Algorithms 5 (diagonal dominance), 7 (partial orthogonalization) and 9 (uniform sampling), with independent $\mathcal{N}(0, 1)$ entries to initialize both Algorithms 5 and 7. Matrices in $\mathcal{E}_\mathcal{G}$ have two non-zero upper triangular entries $(1, 2)$ and $(2, 3)$. Moreover, $\mathcal{E}_\mathcal{G}$ can be represented as the two dimensional

unit ball's interior:

$$\mathcal{E}_{\mathcal{G}} = \left\{ \begin{pmatrix} 1 & x & 0 \\ x & 1 & y \\ 0 & y & 1 \end{pmatrix} : x^2 + y^2 < 1 \right\} \simeq \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 < 1\},$$

yielding the scatter plots shown in Figure 5.6

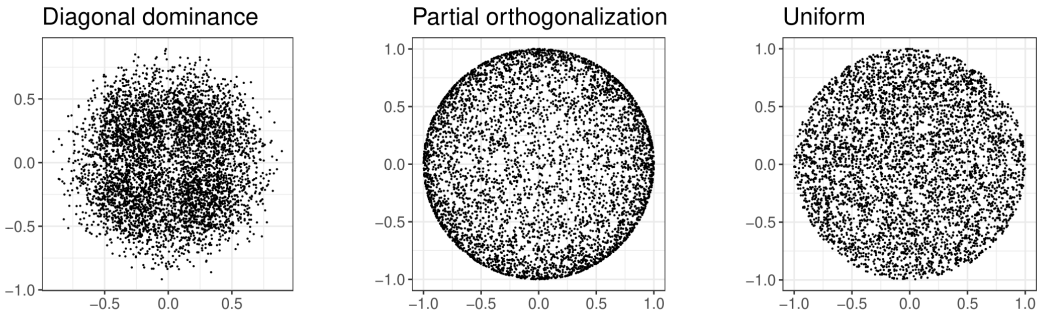



Figure 5.6: Scatter plots of the two non-zero entries for correlation matrices sampled from $\mathcal{E}_{\mathcal{G}}$, with $\mathcal{G}$ as in Figure 5.5.

As expected, the uniform sampling method of Algorithm 9 recovers the whole space uniformly. By contrast, the diagonal dominance method and the partial orthogonalization methods have somehow the opposite behaviour: the former are concentrated on the ball's interior, whereas the latter are more frequently sampled close to its frontier. Furthermore, this implies that partially orthogonalized matrices tend to have large off-diagonal values, while the diagonal dominance method produces matrices with smaller values for the off-diagonal entries, which coincides with what was previously shown in Figures 5.3 and 5.4.

### 5.3.2 Marginal distribution of matrix entries

In order to gain a deeper insight of the above method's behaviour with respect to matrix entries, the chain of 50 vertices (Figure 5.7),

$$\mathcal{G} = (V = \{1, \ldots, 50\}, E = \{(1, 2), (2, 1), (2, 3), (3, 2), \ldots, (49, 50), (50, 49)\}),$$

is considered, which generalizes the graph in Figure 5.5.

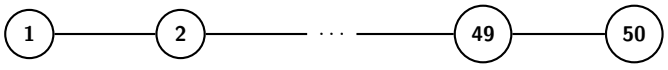



Figure 5.7: Chordal undirected chain with 50 variables and 49 edges

A total of 5000 matrices are sampled from the chain $\mathcal{G}$, and Figure 5.8 contains the marginal densities of the 49 non-zero entries of the generated matrices with the three different methods: diagonal dominance, partial orthogonalization and the general method of Algorithm 10, which in this case amounts to uniform sampling since $\mathcal{G}$ is chordal. It can be observed that the diagonal dominance method produces matrices with off-diagonal entries more concentrated around 0. The partial orthogonalization method produces

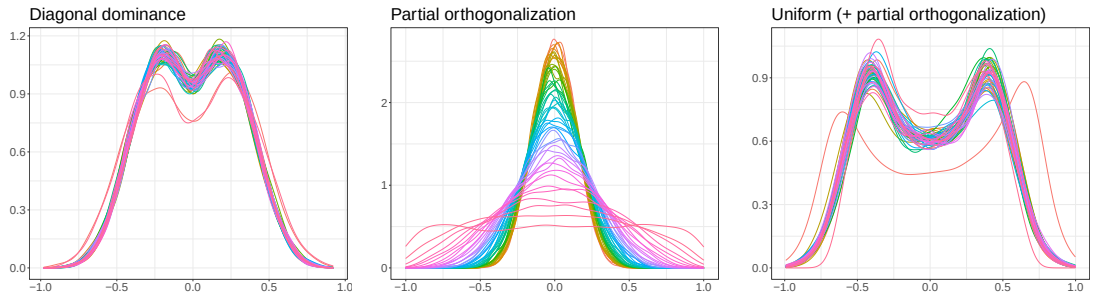



Figure 5.8: Marginal densities of the non-zero entries of matrices sampled from $\mathcal{E}_\mathcal{G}$ with $\mathcal{G}$ the chain of 50 vertices. The first entry in the lower triangle, (2, 1), corresponds to the red colour, while the last entry in the last row of the lower triangle, (49, 48), corresponds to the pink colour.

matrices $\mathbf{M} \in \mathcal{E}_\mathcal{G}$ with the first non-zero entries $m_{12}, m_{23}, m_{34}, \ldots$ more centred around 0 than the last entries $\ldots, m_{48\,49}, m_{49\,50}$. This behaviour is due to the independent and identically distributed entries for initializing factor $\mathbf{Q}$ in Algorithm 7, similarly to what occurred in the case of Gaussian Bayesian networks (Proposition 5.1.1). Intuitively this can be seen as a consequence of the fact that vectors of independent random components are approximately orthogonal in high-dimensions. Indeed, partial orthogonalization part mitigates this fact: the first entries of the matrix, $m_{12}, m_{23}$, are the ones where no orthogonalization is applied by Algorithm 7. On the contrary, uniform sampling correctly produces matrices with the same marginal density for each entry.

Finally, a general random undirected graph $\mathcal{G}$ is generated over 50 vertices using the Erdös-Rényi model (Erdös and Rényi, 1959) with a probability of edges equal to 0.05, and again 5000 matrices are sampled from $\mathcal{E}_\mathcal{G}$ using Algorithms 5, 7 and 10. Note that in this case Algorithm 10 will not in general be equal to uniform sampling. The marginal densities of the non-zero entries for the three methods are plotted in Figure 5.9. It can be observed that partial orthogonalization is the more robust method in the sense that its performance is relatively unchanged from Figure 5.8. Diagonal dominance and Algorithm 10, by contrast, now show a higher degree of variability for different entries, whereas in Figure 5.8 the marginal density was similar for all entries.

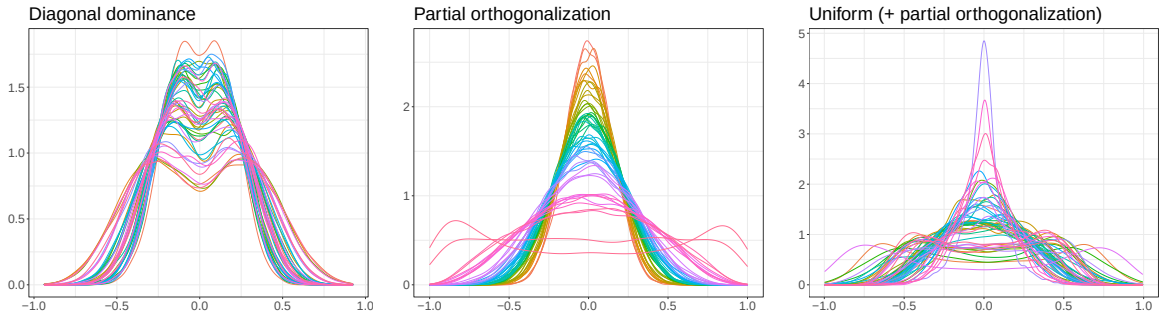



Figure 5.9: Marginal densities of the non-zero entries of matrices sampled from $\mathcal{E}_\mathcal{G}$, where $\mathcal{G}$ is a random graph with 50 vertices and probability of edges 0.05. The first entry in the lower triangle (2, 1) corresponds to the red colour, while the last entry in the last row of the lower triangle (50, 49) corresponds to the pink colour.

## 5.4 Validation of model selection methods

The main motivation for this simulation study are the observations that can be found in the literature on Gaussian graphical models regarding the difficulties of validating the performance of model selection methods (Schäfer and Strimmer, 2005a; Krämer et al., 2009; Cai et al., 2011). As an illustration, the work of Krämer et al. (2009, page 7) has been selected. Therein, the authors highlight how they obtain significantly poorer graph recovery results as the density $d$ of the graphs grows. They simulate the corresponding undirected Gaussian graphical models using the diagonal dominance method (Algorithm 5), so in this section their experiments have been replicated using instead as true models those generated with both partial orthogonalization (Algorithm 7) and the hybrid approach combining uniform sampling (Algorithm 10).

The results can be seen in Figures 5.10 and 5.11, which depict the true positive rate (TPR, also called power by Krämer et al. (2009)) and the positive predictive value (PPV) or precision for $p = 100$ and their sparsest ($d = 0.05$) and densest ($d = 0.25$) scenarios. The different structure learning methods are the same than those studied by Krämer et al. (2009): adaptive $l_1$ regularization (`adalasso`), $l_1$ regularization (`lasso`), partial least squares regression (`pls`), shrinkage estimator of Schäfer and Strimmer (2005b) (`shrink`), and $l_2$ regularization (`ridge`). Note that in the computations for TPR and PPV, the indefinite fraction 0/0 is correctly defined to be equal to 1. For some learning methods such as `shrink` and `pls` this drastically affects their curve when comparing to Krämer et al. (2009).

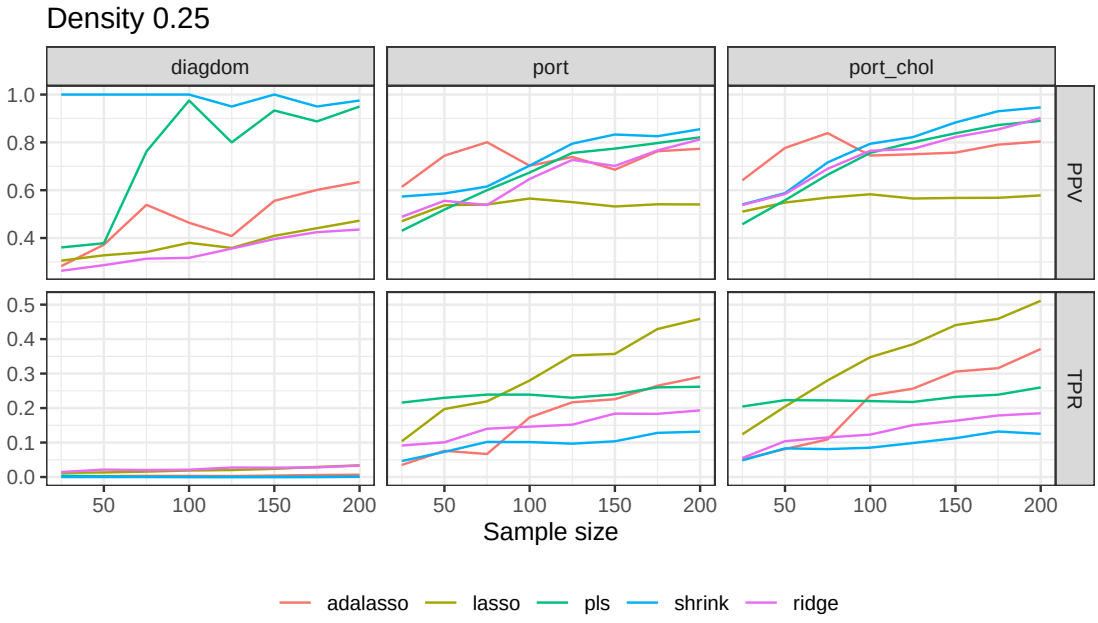



Figure 5.10: TPR and PPV of model selection methods for Gaussian Markov networks validated by Krämer et al. (2009), for the highest density, 0.25. `diagdom`: Diagonal dominance sampling method; `port`: Partial orthogonalization; `port_chol`: Uniform sampling with partial orthogonalization of the Cholesky factor.

Note that there is a significant improvement in the densest case ($d = 0.25$) when using our method (Algorithm 10). All the learning algorithms are close to zero TPR for every

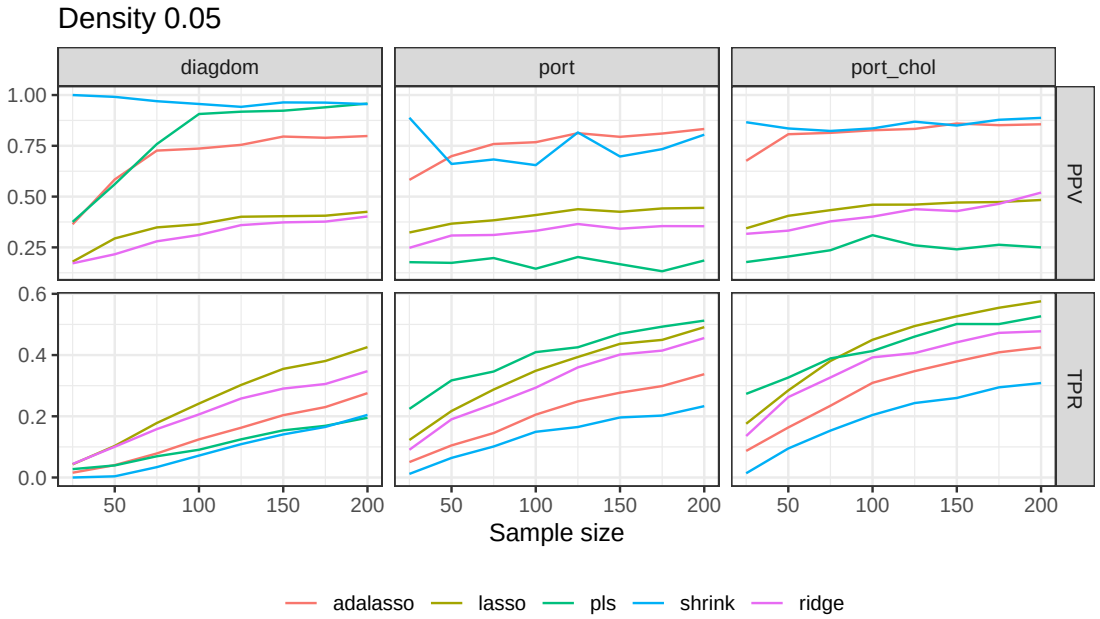



Figure 5.11: TPR and PPV of model selection methods for Gaussian Markov networks validated in (Krämer et al., 2009), for the lowest density, 0.05. `diagdom`: Diagonal dominance sampling method; `port`: Partial orthogonalization; `port_chol`: Uniform sampling with partial orthogonalization of the Cholesky factor.

sample size when validating on diagonally dominant matrices, which highlights a poor performance (the high PPVs are thus not significant). However, when using matrices obtained via partial orthogonalization, some methods (`lasso` and `adalasso`) are able to achieve a TPR of 0.5 approximately. Importantly, partial least squares regression (`pls`) and the shrinkage estimator (`shrink`) greatly improve, whereas when only using diagonal dominance one could erroneously conclude that those methods are not well fitted for dense structure scenarios. In the sparsest scenario ($d = 0.05$) it can be observed that the PPV for partial least squares extremely drops when using our proposed simulation method, while the other algorithms rank similarly using either one. This behaviour is expected: the densest scenario ($d = 0.25$) is not intrinsically difficult, but it indeed poses special difficulties when using diagonally dominant matrices, because correlations are in general small (Figures 5.9 – 5.8) and therefore structure recovery amounts to discriminating an absent edge from an extremely small entry, which is a significantly hard task. The behaviour of partial orthogonalization and the hybrid method combining uniform sampling is rather similar, even though, as was shown in the previous section, in some scenarios they exhibit different numerical properties.

The most important conclusion to draw from these results is that simulation methods for Gaussian graphical models highly influence how the respective model selection methods are ranked. However, it would be incorrect to claim that one of the simulation methodologies is superior to another. Indeed, what is of importance is to choose the correct simulation method for each synthetic validation scenario. For example, if the goal is to assess performance for a wide range of Gaussian Markov networks, then the hybrid method (Algorithm 10) should be used, since it guarantees uniform sampling for chordal models, and therefore unbiased validation. On the contrary, if target models are

known to exhibit small correlations, then using diagonally dominant models would be justified because they have such property. When in doubt, it can be argued to use partial orthogonalization or the hybrid method, because it samples from a wider space range (Figure 5.6), while the diagonal dominance method is largely biased towards matrices away from the space frontier.

# Chapter 6

# Sparse Cholesky covariance parametrization

When a zero pattern is present in the inverse covariance matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$, it represents absent edges in the undirected graph of a Gaussian Markov network (Equation (2.8)). Furthermore, letting $\mathbf{\Omega} = \mathbf{W}\mathbf{W}^t$ be its Cholesky decomposition, a zero pattern in the triangular matrix $\mathbf{W}$ yields the acyclic digraph associated with a Gaussian Bayesian network model (Equation (2.6)), up to a permutation of the variables. As a result, much of the academic focus has been on sparsity in either the inverse covariance matrix or its Cholesky decomposition (Pourahmadi, 1999; d'Aspremont et al., 2008; Friedman et al., 2008; Rothman et al., 2008; Córdoba et al., 2020a). Conversely, a zero pattern in the covariance matrix $\mathbf{\Sigma}$ represents missing edges from the undirected graph of a covariance graph model (Equation (5.1)). However, a structured zero pattern on the Cholesky decomposition $\mathbf{\Sigma} = \mathbf{T}\mathbf{T}^t$ of the covariance matrix has been only addressed by few works. Wermuth et al. (2006) briefly analyse zeros in $\mathbf{T}$ as a tool for better understanding of a higher-level graphical model called covariance chain, which is the main focus of their work. Rothman et al. (2010) directly explore a new regression interpretation of $\mathbf{T}$; however, they focus on a banding structure for $\mathbf{T}$ instead of a general zero pattern. In fact, a significant amount of the paper is devoted to analysing the relationship between the covariance matrix, or its inverse, and banded Cholesky factorization. In contrast, the focus of this chapter is on arbitrary zero patterns in $\mathbf{T}$.

## 6.1 Cholesky decomposition of a covariance matrix

A Gaussian random vector model $\boldsymbol{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ can be equivalently expressed as a system of recursive regressions,

$$X_i = \sum_{j<i} \beta_{ij|1\ldots,i-1} X_j + \varepsilon_i, \tag{6.1}$$

which is the same as Equation (2.4) if $\boldsymbol{X}$ additionally follows a Gaussian Bayesian network model $\mathcal{M}(\mathcal{G})$ and $1 \prec \cdots \prec p$ is assumed to be the ancestral order of $\mathcal{G}$. Taking variances and inverting, the upper Cholesky factorization (Equation (2.5)) is obtained, $\mathbf{\Sigma} = \mathbf{U}\mathbf{D}^{-1}\mathbf{U}^t$. This factorization contains all parameters of the model in Equation (6.1):

$\mathbf{D} = \text{var}(\boldsymbol{\mathcal{E}})$ and

$$\mathbf{U} = \begin{pmatrix} 1 & -\beta_{21|1} & \cdots & -\beta_{p1|1,\ldots,p-1} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & -\beta_{pp-1|1,\ldots,p-1} \\ 0 & \cdots & 0 & 1 \end{pmatrix}. \tag{6.2}$$

If omitting the inversion step from Equation (6.1), then the Cholesky decomposition of the covariance matrix Rothman et al. (2010) is obtained

$$\boldsymbol{\Sigma} = \mathbf{L}\mathbf{D}\mathbf{L}^t = \mathbf{T}\mathbf{T}^t, \tag{6.3}$$

where now $\mathbf{L} = (\mathbf{I}_p - \mathbf{B})^{-1}$ and $\mathbf{T} = \mathbf{L}\sqrt{\mathbf{D}}$ are lower triangular. Observe that Equation (6.3) is a direct analogue of Equation (2.5). Furthermore, the entries of $\mathbf{L}$ are also regression coefficients, as shown in Proposition 6.1.1. Alternative derivations of this result can be found in (Dempster, 1969, p. 158) and (Wermuth et al., 2006, p. 846). The one in Dempster (1969) is computational, based on the sweep matrix operator Beaton (1964), whereas Wermuth et al. (2006) provides a sketch based on a recursive expression for regression coefficients. This proof uses instead simple identities over partitioned matrices.

**Proposition 6.1.1.** *For $i \in \{1, \ldots, p\}$ and $j < i$, the $(i, j)$ entry of matrix $\mathbf{L}$, denoted as $l_{ij}$, is equal to $\beta_{ij|1,\ldots,j}$.*

*Proof.* For each $i \in \{1, \ldots, p\}$, $j < i$ and $J = \{1, \ldots, j\}$, the following partitioned identities hold (Dempster, 1969, Equation (4.2.18)):

$$\begin{aligned} \boldsymbol{\Sigma}_{iJ} &= \mathbf{L}_{iJ}\mathbf{D}_{JJ}\mathbf{L}_{JJ}^t, \\ \boldsymbol{\Sigma}_{JJ} &= \mathbf{L}_{JJ}\mathbf{D}_{JJ}\mathbf{L}_{JJ}^t. \end{aligned}$$

Therefore,

$$\begin{aligned} \boldsymbol{\beta}_{i|J}^t &= \boldsymbol{\Sigma}_{iJ}\boldsymbol{\Sigma}_{JJ}^{-1} \\ &= \mathbf{L}_{iJ}\mathbf{D}_{JJ}\mathbf{L}_{JJ}^t(\mathbf{L}_{JJ}\mathbf{D}_{JJ}\mathbf{L}_{JJ}^t)^{-1} \\ &= \mathbf{L}_{iJ}\mathbf{L}_{JJ}^{-1}. \end{aligned}$$

Furthermore, observe that, since $\mathbf{L}_{JJ}$ is lower triangular with ones along the diagonal, the last column of $\mathbf{L}_{JJ}^{-1}$ is always a vector of zero entries except the last entry, which is 1. This means, in particular, that for each $i \in \{1, \ldots, p\}$, $j < i$ and $J = \{1, \ldots, j\}$, the $j$-th element of row vector $\mathbf{L}_{iJ}\mathbf{L}_{JJ}^{-1}$ is equal to $l_{ij}$, which in turn is equal to the $j$-th entry of $\boldsymbol{\beta}_{i|J}$, $\beta_{ij|J}$. □

The explicit expression in terms of regression coefficients for $\mathbf{L}$ is therefore,

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \beta_{21|1} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \beta_{p1|1} & \cdots & \beta_{pp-1|1,\ldots,p-1} & 1 \end{pmatrix}. \tag{6.4}$$

In terms of model estimation, in $\mathbf{U}$ (Equation (6.2)) each column corresponds to the parameters of a single recursive regression, whereas each entry of matrix $\mathbf{L}$ (Equation (6.4)) corresponds to a different regression model. Fortunately, Rothman et al.

(2010) gave an alternative interpretation for matrix $\mathbf{L}$, as follows. The model of Equation (6.1) is equivalent to a linear model $\boldsymbol{X} = \mathbf{L}\boldsymbol{\mathcal{E}}$ where $\boldsymbol{\mathcal{E}} \sim \mathcal{N}(\mathbf{0}, \mathbf{D})$, since in such case $\boldsymbol{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{L}\mathbf{D}\mathbf{L}^t)$. Unfolding such matrix equation, the following regression system is obtained

$$X_i = \sum_{j=1}^{i-1} l_{ij}\mathcal{E}_j + \mathcal{E}_i, \tag{6.5}$$

which is an analogue of Equation (6.1), but now instead of recursively regressing the original variables, each regression is performed over the ordered *error* terms in Equation (6.1).

**Remark.** *The $(i, j)$ entry of matrix $\mathbf{L}$ for $j < i$, $l_{ij}$, has therefore two interpretations as a regression coefficient:*

1. *It is the coefficient of the error $\mathcal{E}_j$ on the regression of $X_i$ over $\mathcal{E}_1, \ldots, \mathcal{E}_{i-1}$.*
2. *It is the coefficient of variable $X_j$ in the regression of $X_i$ over $X_1, \ldots, X_j$.*

*Furthermore, from Equations* (6.1) *and* (6.5) *variable $\mathcal{E}_i$ has also a dual interpretation: it is the regression error of $X_i$ onto $X_1, \ldots, X_{i-1}$, but also of $X_i$ onto $\mathcal{E}_1, \ldots, \mathcal{E}_{i-1}$.*

## 6.2 A sparse model for the Cholesky factor

If allowing an arbitrary zero pattern in $\mathbf{T}$, a sparse Cholesky decomposition model for the covariance matrix is obtained. The entries of $\mathbf{T}$ are in correspondence with those in $\mathbf{L}$ and $\mathbf{D}$ (see Equation (6.3), $t_{ij} = l_{ij}\sqrt{d_{ii}}$), and therefore sometimes they will be indistinctly used.

### 6.2.1 Hidden variable interpretation

The sparse Cholesky parametrization of the covariance matrix naturally models a hidden variable structure (Chandrasekaran et al., 2012; Yatsenko et al., 2015; Zorzi and Sepulchre, 2016; Basu et al., 2019) over ordered Gaussian observables (Equation (6.5)). Interpreting the *error* terms $\boldsymbol{\mathcal{E}}$ as latent signal sources, then the model is a sort of restricted Gaussian Bayesian network with the following constraints:

- All arcs are from hidden variables $\boldsymbol{\mathcal{E}}$ to the observed ones $\boldsymbol{X}$.
- There is always an arc from $\mathcal{E}_i$ to $X_i$, for all $i \in \{1, \ldots, p\}$.
- For each $i \in \{1, \ldots, p\}$, only variables $\mathcal{E}_1, \ldots, \mathcal{E}_{i-1}$ can have arcs to $X_i$.

Figure 6.1 represents one such restricted Gaussian Bayesian network compatible with the sparse Cholesky factorization model for the covariance. The sparse Cholesky factor for Figure 6.1 would have the following lower triangular pattern

$$\begin{pmatrix} * & \\ 0 & * \end{pmatrix}, \tag{6.6}$$

where an asterisk means a non-zero entry. Observe that the zero value in entry $(3, 1)$ corresponds to the missing edge from $\mathcal{E}_1$ to $X_3$ in Figure 6.1.

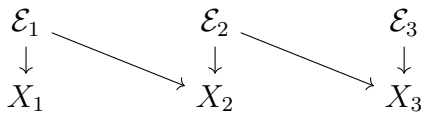



Figure 6.1: Hidden variable model interpretation. The index set notation has been omitted for vertices and variables have been directly used for clarity.

### 6.2.2 A graphical model extension for unordered variables

In direct analogy with Gaussian Bayesian networks and Equation (2.6), a new graphical model can be defined which is parametrized by the Cholesky factorization of the covariance, up to a permutation: for a given arbitrary acyclic digraph $\mathcal{G} = (V, E)$, such graphical models is defined as

$$\mathcal{M}(\mathcal{G}) = \{\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) : \mathbf{\Sigma} = \mathbf{T}\mathbf{T}^t,\, \mathbf{T}^t \in \mathbb{M}_{\mathcal{G}}\}, \tag{6.7}$$

where recall that $\mathbb{M}_{\mathcal{G}}$ is the set of matrices compatible with $\mathcal{G}$, that is, such that $m_{ji} = 0$ for all $(j, i) \notin E$, $j \neq i$.

**Remark.** *As in the case of Gaussian Bayesian networks, the parameter matrix* $\mathbf{T}$ *in Equation* (6.7) *will only be lower triangular, and thus coincide with the Cholesky factor of* $\mathbf{\Sigma}$*, if the variables are already ancestrally ordered.*

Note that Equation (6.4) holds for an unordered version of $\mathbf{L}$, and thus $t_{ij} \propto \beta_{ij|\,\mathrm{pr}_{\prec}(j)}$ for $j \prec i$. Therefore, a sort of *ordered Markov property* can be retrieved for this new graphical model (comparable to Equation (2.2))

$$X_i \perp\!\!\!\perp X_j | \boldsymbol{X}_{\mathrm{pr}_{\prec}(j)} \text{ for all } (j, i) \notin E \text{ with } j \prec i. \tag{6.8}$$

A simple example of an arbitrary graph would be that in Figure 6.2. In this model, factor $\mathbf{T}$ would be lower triangular after reordering its rows and columns following the ancestral ordering of the graph, $2 \prec 1 \prec 3$, obtaining a new matrix $\tau(\mathbf{T})$,

$$\mathbf{T} = \begin{pmatrix} * & * & 0 \\ 0 & * & 0 \\ * & 0 & * \end{pmatrix}, \tau(\mathbf{T}) = \begin{pmatrix} * & 0 & 0 \\ * & * & 0 \\ 0 & * & * \end{pmatrix}.$$




Figure 6.2: Graph with ancestral order $2 \prec 1 \prec 3$. The index set notation has been omitted for vertices and variables have been directly used for clarity.

In the example of Figure 6.1, where variables already exhibit a natural order, the graph that would represent such interactions would be that in Figure 6.3, whose parameter matrix $\mathbf{T}$ is already lower triangular.

## 6.3 Model estimation

We will first review two regression-based existing estimators for this model that can be found in Rothman et al. (2010), and then will detail our proposed penalized matrix loss estimation method.

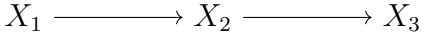



Figure 6.3: Graph corresponding to the model in Figure 6.1. The index set notation has been omitted for vertices and variables have been directly used for clarity.

### 6.3.1 Existing work: Banding and lasso

Throughout this section denote as $\boldsymbol{x}_i$ a sample of size $N$ corresponding to variable $X_i$, where $\boldsymbol{X} = (X_1, \ldots, X_p)$ is assumed to follow the regression model of Equation (6.5).

The banding estimate for $\mathbf{T}$ builds upon the respective for $\mathbf{L}$. The idea is to estimate by standard maximum likelihood only the first $k$ sub-diagonals of $\mathbf{L}$ and set the rest to zero. Specifically, if $b(k) = \max(1, i-k)$ denotes the starting index, with respect to the band parameter $k$, of the $i$-th row vector $\boldsymbol{l}_i = (l_{ib(k)}, \ldots, l_{ii-1})^t$ in matrix $\mathbf{L}$, then, letting $\hat{\boldsymbol{\varepsilon}}_{b(k)} = \boldsymbol{x}_{b(k)}$,

$$\begin{aligned} \hat{\boldsymbol{l}}_i &= \arg\min_{\boldsymbol{l}_i} \|\boldsymbol{x}_i - (\hat{\boldsymbol{\varepsilon}}_{b(k)} \cdots \hat{\boldsymbol{\varepsilon}}_{i-1})\boldsymbol{l}_i\|_2^2 \\ \hat{\boldsymbol{\varepsilon}}_i &= \boldsymbol{x}_i - (\hat{\boldsymbol{\varepsilon}}_{b(k)} \cdots \hat{\boldsymbol{\varepsilon}}_{i-1})\hat{\boldsymbol{l}}_i \\ \hat{d}_{ii} &= \frac{1}{N}\|\hat{\boldsymbol{\varepsilon}}_i\|_2^2, . \end{aligned} \tag{6.9}$$

In order to ensure positive definiteness of all matrices involved in the computations, $k$ must be smaller than $\min(N-1, p)$ (Rothman et al., 2010). Matrix $\widehat{\mathbf{T}} = \widehat{\mathbf{L}}\sqrt{\widehat{\mathbf{D}}}$inherits the band structure from $\widehat{\mathbf{L}}$. The main drawback of this banding estimator is the restrictive zero pattern that it imposes. Note also that this method requires previous selection of the parameter $k$.

An alternative to banding which gives more flexibility over the zero pattern is to use $l_1$ penalization over Equation (6.5),

$$\hat{\boldsymbol{l}}_i = \arg\min_{\boldsymbol{l}_i} \|\boldsymbol{x}_i - (\hat{\boldsymbol{\varepsilon}}_1 \cdots \hat{\boldsymbol{\varepsilon}}_{i-1})\boldsymbol{l}_i\|_2^2 + \lambda\|\boldsymbol{l}_i\|_1 \tag{6.10}$$

where this time $\hat{\boldsymbol{\varepsilon}}_i = \boldsymbol{x}_i - (\hat{\boldsymbol{\varepsilon}}_1 \cdots \hat{\boldsymbol{\varepsilon}}_{i-1})\hat{\boldsymbol{l}}_i$ with $\hat{\boldsymbol{\varepsilon}}_1 = \boldsymbol{x}_1$ and $\boldsymbol{l}_i = (l_{i1}, \ldots, l_{ii-1})^t$, and $\lambda > 0$ is the penalisation parameter. Observe that such penalty could be replaced with any other sparsity inducing penalty over $\boldsymbol{l}_i$.

### 6.3.2 Penalized learning of the covariance Cholesky factor

The above approaches are based on the regression interpretation of the sparse Cholesky factor model, Equation (6.5). By contrast, all of the parameters could be directly estimated by solving one optimization problem. This allows for example to recover maximum likelihood estimates, as well as to be easily extended to the graphical model interpretation (Equation (6.7)) following an approach similar to Zheng et al. (2018).

Denote as $\boldsymbol{\Sigma}(\mathbf{T})$ the parametrization of a covariance matrix $\boldsymbol{\Sigma}$ with its Cholesky factor $\mathbf{T}$ (Equation (6.3)). A sparse model for $\mathbf{T}$ can be learned by solving the following optimization problem

$$\arg\min_{\mathbf{T}} \phi\left(\boldsymbol{\Sigma}(\mathbf{T})\right) + \lambda\|\mathbf{T}\|_1, \tag{6.11}$$

where $\phi(\cdot)$ is a differentiable loss function over covariance matrices, $\lambda > 0$ is the penalisation parameter, and $\|\cdot\|_1$ is the $l_1$-norm for matrices, which induces sparsity on $\mathbf{T}$

(Bach et al., 2012). Note that, as in the regression case, the $l_1$ penalty could be replaced with any other sparsity inducing matrix norm. Solving Equation (6.11) can be done via proximal gradient algorithms, which have optimal convergence rates among first-order methods (Bach et al., 2012) and are tailored for a convex $\phi$ but also competitive in the non-convex case Varando and Hansen (2020). As an illustration, two such smooth loss functions have been selected: the negative Gaussian log-likelihood and the Frobenious norm.

The negative Gaussian log-likelihood for a sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ when $\boldsymbol{\mu}$ is assumed to be zero is proportional to

$$\phi_{NLL}(\boldsymbol{\Sigma}) = \ln\det(\boldsymbol{\Sigma}) + \operatorname{tr}(\boldsymbol{\Sigma}^{-1}\hat{\boldsymbol{\Sigma}}), \tag{6.12}$$

where $\hat{\boldsymbol{\Sigma}} = 1/N \sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^t$ is the maximum likelihood estimator for $\boldsymbol{\Sigma}$. On the other hand, the Frobenious norm loss is

$$\phi_{FR}(\boldsymbol{\Sigma}) = \|\boldsymbol{\Sigma} - \hat{\boldsymbol{\Sigma}}\|_F^2 = \sum_{i=1}^{p}\sum_{j=1}^{p}(\sigma_{ij} - \hat{\sigma}_{ij})^2 \tag{6.13}$$

Both $\phi_{NLL}$ and $\phi_{FR}$ are smooth, and in general $\phi_{NLL}$ renders the optimization problem of Equation (6.11) non-convex (Boyd and Vandenberghe, 2004), whereas $\phi_{FR}$ is a convex function.

### 6.3.3 Computational details of the proximal gradient algorithm

A simplified expression for the gradient of $\phi(\boldsymbol{\Sigma}(\mathbf{T}))$ with respect to $\mathbf{T}$ can be obtained as a function of the one with respect to $\boldsymbol{\Sigma}$. These gradients will be denoted as $\nabla_{\mathbf{T}}\phi$ and $\nabla_{\boldsymbol{\Sigma}}\phi$, respectively.

**Proposition 6.3.1.** *For any differentiable loss function $\phi(\boldsymbol{\Sigma}(\mathbf{T}))$,*

$$\nabla_{\mathbf{T}}\phi = 2\nabla_{\boldsymbol{\Sigma}}\phi\mathbf{T}. \tag{6.14}$$

*Proof.* This proof follows some ideas from Varando and Hansen (2020), Proposition 2.1. By matrix calculus (Petersen and Pedersen, 2008),

$$\frac{\partial\phi(\boldsymbol{\Sigma}(\mathbf{T}))}{\partial t_{ij}} = tr\left(\nabla_{\boldsymbol{\Sigma}}\phi\frac{\partial\boldsymbol{\Sigma}(\mathbf{T})}{\partial t_{ij}}\right), \tag{6.15}$$

since $\nabla_{\boldsymbol{\Sigma}}\phi$ is symmetric. Furthermore, note that (Petersen and Pedersen, 2008)

$$\frac{\partial\boldsymbol{\Sigma}(\mathbf{T})}{\partial t_{ij}} = \frac{\partial\mathbf{T}\mathbf{T}^t}{\partial t_{ij}} = \mathbf{T}\mathbf{E}^{ij} + \mathbf{E}^{ji}\mathbf{T}^t,$$

where $\mathbf{E}^{ij}$ ($\mathbf{E}^{ji}$) has its $(i,j)$ ($(j,i)$) entry equal to one and zero elsewhere. Then, from Equation (6.15),

$$\begin{aligned}\frac{\partial\phi(\boldsymbol{\Sigma}(\mathbf{T}))}{\partial t_{ij}} &= \operatorname{tr}\left(\nabla_{\boldsymbol{\Sigma}}\phi(\mathbf{T}\mathbf{E}^{ij} + \mathbf{E}^{ji}\mathbf{T}^t)\right)\\ &= \operatorname{tr}\left(\nabla_{\boldsymbol{\Sigma}}\phi\mathbf{T}\mathbf{E}^{ij}\right) + \operatorname{tr}\left(\nabla_{\boldsymbol{\Sigma}}\phi\mathbf{E}^{ji}\mathbf{T}^t\right)\\ &= \operatorname{tr}\left(\mathbf{E}^{ji}\mathbf{T}^t\nabla_{\boldsymbol{\Sigma}}\phi\right) + \operatorname{tr}\left(\mathbf{E}^{ji}\mathbf{T}^t\nabla_{\boldsymbol{\Sigma}}\phi\right)\\ &= 2\operatorname{tr}\left(\mathbf{E}^{ji}\mathbf{T}^t\nabla_{\boldsymbol{\Sigma}}\phi\right).\end{aligned}$$

Since $a_{ji} = \operatorname{tr}(\mathbf{E}^{ji}\mathbf{A})$ for any matrix $\mathbf{A}$, the desired result is obtained. □

The above proposition implies that once a loss function $\phi(\mathbf{\Sigma}(\mathbf{T}))$ is fixed, it is only necessary to compute $\nabla_{\mathbf{\Sigma}}\phi$ in order to obtain $\nabla_{\mathbf{T}}\phi$. The gradient for $\phi_{NLL}$ and $\phi_{FR}$ can thus be easily obtained. Standard matrix calculus (Petersen and Pedersen, 2008) gives $\nabla_{\mathbf{\Sigma}}\phi_{NLL} = \mathbf{\Sigma}^{-1} - \mathbf{\Sigma}^{-1}\hat{\mathbf{\Sigma}}\mathbf{\Sigma}^{-1}$. Therefore,

$$\begin{aligned}\nabla_{\mathbf{T}}\phi_{NLL} &= 2\nabla_{\mathbf{\Sigma}}\phi_{NLL}\mathbf{T} \\ &= 2\mathbf{\Sigma}^{-1}(\mathbf{T})(\mathbf{I}_p - \hat{\mathbf{\Sigma}}\mathbf{\Sigma}^{-1}(\mathbf{T}))\mathbf{T} \\ &= 2\mathbf{T}^{-t}(\mathbf{I}_p - \mathbf{T}^{-1}\hat{\mathbf{\Sigma}}\mathbf{T}^{-t}).\end{aligned}$$

Conversely, $\nabla_{\mathbf{\Sigma}}\phi_{FR} = 2(\mathbf{\Sigma} - \hat{\mathbf{\Sigma}})$. Thus $\nabla_{\mathbf{T}}\phi_{FR} = 2(\mathbf{T}\mathbf{T}^t - \hat{\mathbf{\Sigma}})\mathbf{T}$.

## 6.4 Experiments

In all experiments the four estimation methods outlined in the previous section are compared: banding $\mathbf{T}$ (Equation (6.9)), $l_1$ or *lasso* regularization (Equation (6.10)), and the two proposed penalized losses $\phi_{NLL}$ (Equation (6.12)) and $\phi_{FR}$ (Equation (6.13)). These four methods will be denoted in the remainder as band, lasso, grad_lik and grad_frob, respectively. All data was standardized, and therefore for grad_lik and grad_frob the sample correlation matrix was used instead of $\hat{\mathbf{\Sigma}}$. The implementation of our loss optimization methods grad_frob and grad_lik can be found in the *R* package *covchol*[1]. The experiments described throughout this section can be reproduced following the instructions and using the code available at the repository https://github.com/irenecrsn/chol-inv.

### 6.4.1 Simulation

Two different simulation scenarios have been selected. First, because as the work of Rothman et al. (2010) is the most directly related to the proposed sparse covariance Cholesky factorization model, their simulation setting has been replicated for completeness. Therein they select three fixed covariance matrices with either a fixed known banded sparsity pattern or no zeros at all. By contrast, in the second experiment an arbitrary pattern is explored.

In both experiments two statistics have been measured in order to assess both model selection and estimation. These metrics are evaluated over $\mathbf{\Sigma}$ and $\widehat{\mathbf{\Sigma}}$ in the first experiment instead of $\mathbf{T}$ and $\widehat{\mathbf{T}}$, for better comparability with Rothman et al. (2010). Specifically, the F1 score is used for evaluating the zero pattern,

$$\mathrm{F1}(\mathbf{T}, \widehat{\mathbf{T}}) = 2\frac{\mathrm{TPR}(\mathbf{T}, \widehat{\mathbf{T}})\,\mathrm{TDR}(\mathbf{T}, \widehat{\mathbf{T}})}{\mathrm{TPR}(\mathbf{T}, \widehat{\mathbf{T}}) + \mathrm{TDR}(\mathbf{T}, \widehat{\mathbf{T}})}, \tag{6.16}$$

where TPR and TDR are the true positive and discovery rate, respectively,

$$\begin{aligned}\mathrm{TPR}(\mathbf{T}, \widehat{\mathbf{T}}) &= \frac{|\{t_{ij} \neq 0 \text{ and } \hat{t}_{ij} \neq 0\}|}{|\{t_{ij} \neq 0\}|}, \\ \mathrm{TDR}(\mathbf{T}, \widehat{\mathbf{T}}) &= \frac{|\{t_{ij} \neq 0 \text{ and } \hat{t}_{ij} \neq 0\}|}{|\{\hat{t}_{ij} \neq 0\}|};\end{aligned}$$

[1]Version under development: https://github.com/irenecrsn/covchol.

and the induced matrix 1-norm is used for numerical evaluation,

$$\text{NORM}(\mathbf{T}, \widehat{\mathbf{T}}) = \|\mathbf{T} - \hat{\mathbf{T}}\|_1 = \max_{1 \leq j \leq p} \sum_{i=1}^{p} |\hat{t}_{ij} - t_{ij}|.$$

### Fixed covariance matrices

The fixed covariance matrices used in the simulations by Rothman et al. (2010) are:

- The autoregressive model of order 1, where the true covariance matrix $\mathbf{\Sigma}_1$ has entries $\sigma_{ij} = \rho^{|i-j|}$, with $\rho = 0.7$.
- The 4-banded correlation matrix $\mathbf{\Sigma}_2$ with entries $\sigma_{ij} = 0.4\mathbb{I}(|i-j| = 1) + 0.2\mathbb{I}(2 \leq |i-j| \leq 3) + 0.1\mathbb{I}(|i-j| = 4)$ for $i \neq j$, $\mathbb{I}$ being the set indicator function.
- The dense correlation matrix $\mathbf{\Sigma}_3$ with 0.5 in all of its entries except for the diagonal.

Similarly to Rothman et al. (2010), the matrix dimension $p$ ranges from 30 to 500, and the sample size $N$ is fixed to 200, which allows to visualize both the $p > N$ and $p < N$ scenarios. This experiment measures how sparsity inducing methods for learning $\mathbf{T}$ behave in scenarios which are not specially suited for them, except for band and $\mathbf{\Sigma}_2$.

Figure 6.4 shows the results. $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_3$ are both dense matrices, with $\mathbf{\Sigma}_1$ having entries that decay when moving away from the diagonal. Observe that the inexistent sparsity pattern is best approximated by grad_lik and lasso, but interestingly grad_frob and band achieve competitive norm results, sometimes even outperforming the rest. Matrix $\mathbf{\Sigma}_2$ is banded, therefore, as expected, band achieves both the highest F1 measure and lowest norm difference.

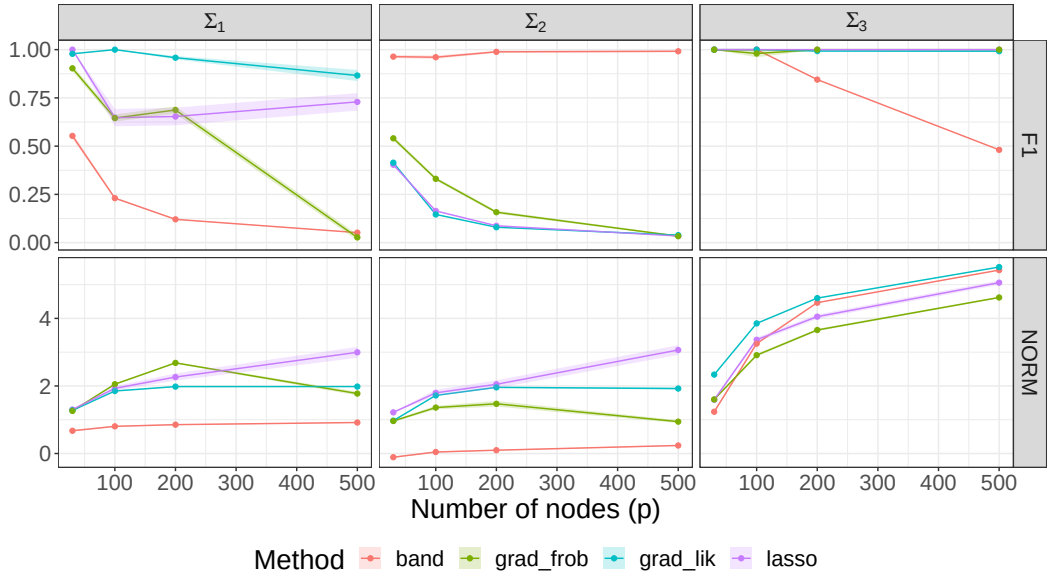



Figure 6.4: Results of the simulation experiment set out in Rothman et al. (2010). Metric NORM is in logarithmic scale for a better comparison between the methods, since there were significant disparities.

**Arbitrary sparsity pattern in the Cholesky factor T**

In this experiment the sparse Cholesky factor $\mathbf{T}$ is simulated using essentially Algorithm 9 with a random acyclic directed orientation to represent zero pattern, that is, the latent structure (see Figure 6.1). Observe that in general this does not yield a uniformly sampled Cholesky factor, but it is more flexible that the standard diagonal dominance procedure. Three Cholesky factors $\mathbf{T}_i$ are generated with a proportion of $i/p$ non-zero entries, where $i \in \{1, 2, 3\}$. Sample size $N$ and matrix dimension $p$ are as in the previous experiment.

Figure 6.5 depicts the results. Note that as the density decreases, the F1 score and matrix norm results slightly worsen, but in general the methods behaviour is maintained. The band estimator exhibits a performance similar to the previous experiment: although achieving a small F1 score, it has a relatively small matrix norm difference. This behaviour is shared in this case with grad_lik, which has in general poor performance. However, the worst performing method is lasso, which neither is able to recover the sparsity pattern, nor gets numerically close to the original Cholesky factor (it has a significantly high value for the norm difference). Conversely, method grad_frob has the best performance, with a significantly high F1 score when compared with the rest and competitive or best norm difference results.

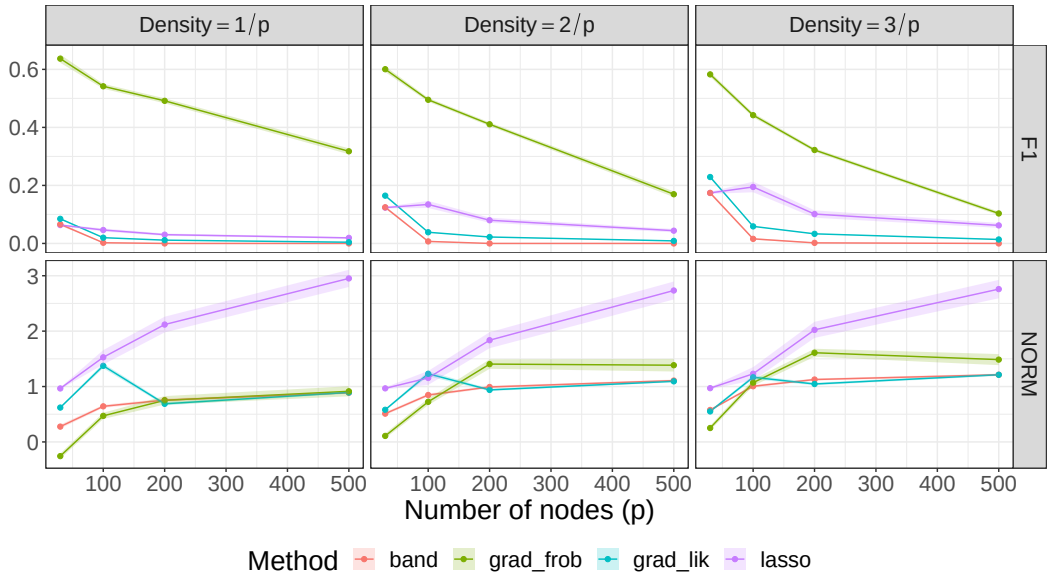



Figure 6.5: Results of the simulation experiment for an arbitrary sparsity pattern in $\mathbf{T}$. Metric NORM is in logarithmic scale. Density indicates the average proportion of lower triangular non-zero entries in the simulated Cholesky factors.

## 6.4.2 Real data

In this section two data sets from the UCI machine learning repository (Dua and Graff, 2020), where a natural order arises among the variables, have been selected. Both of them are labelled with a class variable, therefore after estimating the respective Cholesky factors with each method, classification performance is assessed. For classifying a sample $\boldsymbol{x}$ quadratic discriminant analysis (Rothman et al., 2010) is used, and $\boldsymbol{x}$ is classified in

the class value $c$ that maximizes

$$\ln \hat{f}(\boldsymbol{x}, c) = \ln \hat{f}(\boldsymbol{x}|c) + \ln \hat{f}(c),$$

where $\hat{f}(c)$ is the proportion of observations for class $c$ and $\hat{f}(\boldsymbol{x}|c)$ is expressed in terms of $\mathbf{T}$ instead of $\boldsymbol{\Sigma}$,

$$\begin{aligned} \ln \hat{f}(\boldsymbol{x}|c) &\propto \frac{1}{2} \ln \det(\widehat{\boldsymbol{\Sigma}}_c) - \frac{1}{2}(\boldsymbol{x} - \hat{\boldsymbol{\mu}}_c)^t \widehat{\boldsymbol{\Sigma}}_c^{-1} (\boldsymbol{x} - \hat{\boldsymbol{\mu}}_c) \\ &= \ln \det(\widehat{\mathbf{T}}_c) - \frac{1}{2}(\boldsymbol{x} - \hat{\boldsymbol{\mu}}_c)^t \widehat{\mathbf{T}}_c^{-t} \widehat{\mathbf{T}}_c^{-1} (\boldsymbol{x} - \hat{\boldsymbol{\mu}}_c) \\ &= \sum_{i=1}^{p} \ln t_{ii} - \frac{1}{2}(\widehat{\mathbf{T}}_c^{-1}(\boldsymbol{x} - \hat{\boldsymbol{\mu}}_c))^t \widehat{\mathbf{T}}_c^{-1} (\boldsymbol{x} - \hat{\boldsymbol{\mu}}_c), \end{aligned} \tag{6.17}$$

with $\hat{\boldsymbol{\mu}}_c$, $\widehat{\boldsymbol{\Sigma}}_c$ and $\widehat{\mathbf{T}}_c$ the respective estimates from training samples belonging to class $c$.

Finally, for evaluating classification performance the following metrics have been used:

- The F1 score, already defined in Equation (6.16) in terms of TDR and TPR, but adapted to classification instead of matrix entries.
- The true negative rate, TNR, since it is not contained in the F1 score, which is the proportion of observations that have been correctly *not* classified as class $c$.
- The accuracy, ACC, which measures the proportion of observations that have been correctly assigned a class. Observe that this last metric, unlike the other two, is not class-dependent, but instead global.

### Sonar: Mine vs. Rocks

The first real data set explored is the *Connectionist Bench (Sonar, Mines vs. Rocks)* data set, which contains numeric observations from a sonar signal bounced at both a metal cylinder (mine) and rocks. It contains 60 variables and 208 observations. Each variable corresponds to the energy within a certain frequency band, integrated over a period of time, in increasing order. Each observation represents a different beam angle for the same object under detection. Over this data set the objective is to classify a sample as rock or mine. This data set was also analysed by Rothman et al. (2010), but without the expression in terms of $\mathbf{T}$ for Equation (6.17) and only using method band for $\mathbf{T}$.

As a first exploratory step, each of the methods for learning the Cholesky factor $\mathbf{T}$ has been applied to all instances labelled as $M$ (mines), and $R$ (rocks), shown as a heatmap in Figure 6.6. The Cholesky factor for mines retrieved by grad_lik and lasso look fairly similar, whereas the one for rocks that lasso estimates is nearly diagonal. Bands can be clearly observed from heatmaps by band, and all methods impose zero values for variables near to or higher than 50, which could be motivated by the problem characteristics and hint at high sonar frequencies being nearly noiseless. The entries in the Cholesky factor estimated by grad_frob are the most extreme, since most of them are zero, and the ones which are not have the highest and lowest values among all the estimates recovered.

For the quadratic discriminant analysis leave-one-out cross-validation was used, since the sample size was sufficiently small to allow it. Table 6.1 contains the results thus obtained. Observe that lasso is the method that performs poorest overall. Conversely,

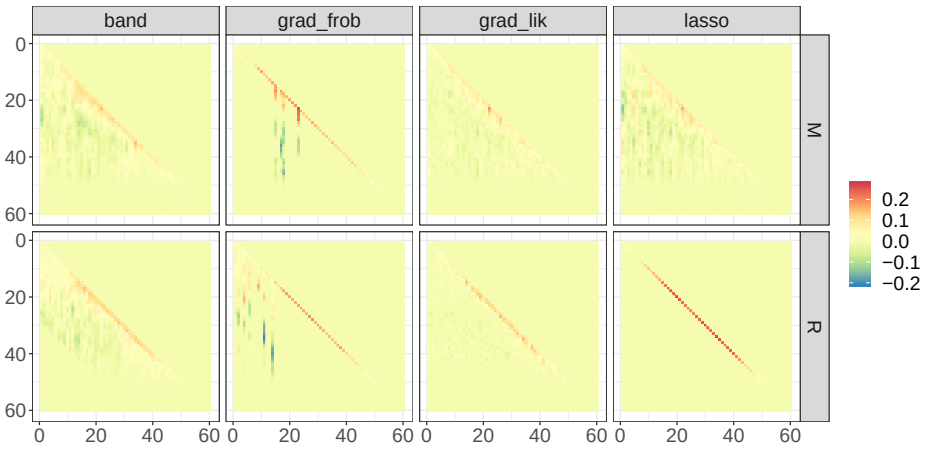



Figure 6.6: Heatmaps of the Cholesky factors of rock and mine samples. M: Mines; R: Rocks.

band is arguably the best for this problem, except for the TNR of rock samples, which is highest for grad_frob. However, observing the rest of statistics for grad_frob, it can be deduced that this method over-classifies samples as mines: it has the lowest TNR for them. On the other hand, grad_lik performs competitively for this problem, but is in general outperformed or matched by band. Since the sonar behaviour hints at a band structure for the covariance (frequency patterns being related to those close to them), and therefore for its Cholesky factor, the good performance of band could be expected.

| | band | grad_frob | grad_lik | lasso |
|---|---|---|---|---|
| TNR (M) | **0.78** | 0.08 | **0.78** | 0.62 |
| F1 (M) | **0.8** | 0.7 | 0.55 | 0.33 |
| TNR (R) | 0.79 | **0.97** | 0.45 | 0.26 |
| F1 (R) | **0.78** | 0.15 | 0.65 | 0.5 |
| ACC | **0.79** | 0.56 | 0.61 | 0.43 |

Table 6.1: Statistics for the sonar problem. M: mines; R: rocks.

### Wall-Following Robot Navigation

The other real data set used is the *Wall-Following Robot Navigation* one. Here a robot moves in a room following the wall clockwise. It contains 5456 observations and 24 variables. Each variable corresponds to the value of an ultrasound sensor, which are arranged circularly over the robot's body. Here the increasing order reflects the reference angle where the sensor is located. Since the robot is moving clockwise, here the classification task is between four possible class values: Move-Forward, Sharp-Right-Turn, Slight-Left-Turn or Slight-Right-Turn.

As in the previous problem, the Cholesky factors are estimated for each of the movements, depicted in Figure 6.7. Notice that grad_frob outputs a similar matrix (except for the Slight-Left-Turn movement) than the other three methods, which means that the

extreme behaviour observed in the sonar experiment was problem-related. By contrast, the Cholesky factor for Slight-Left-Turn is nearly diagonal. The other matrices are rather similar among the methods, with band notably choosing in general a high banding parameter $k$ (few to no bands). Here a similar structure as in the sonar problem can be observed: for all the movements except Slight-Left-Turn it can be appreciated that most entries close to the diagonal are positive, whereas distant ones are frequently negative. Regarding Slight-Left-Turn, these matrices are the sparsest and have near zero values on the diagonal. Since the robot is moving clockwise, this movement is related to obstacles, therefore it could hint that sensor readings are correctly identifying them.

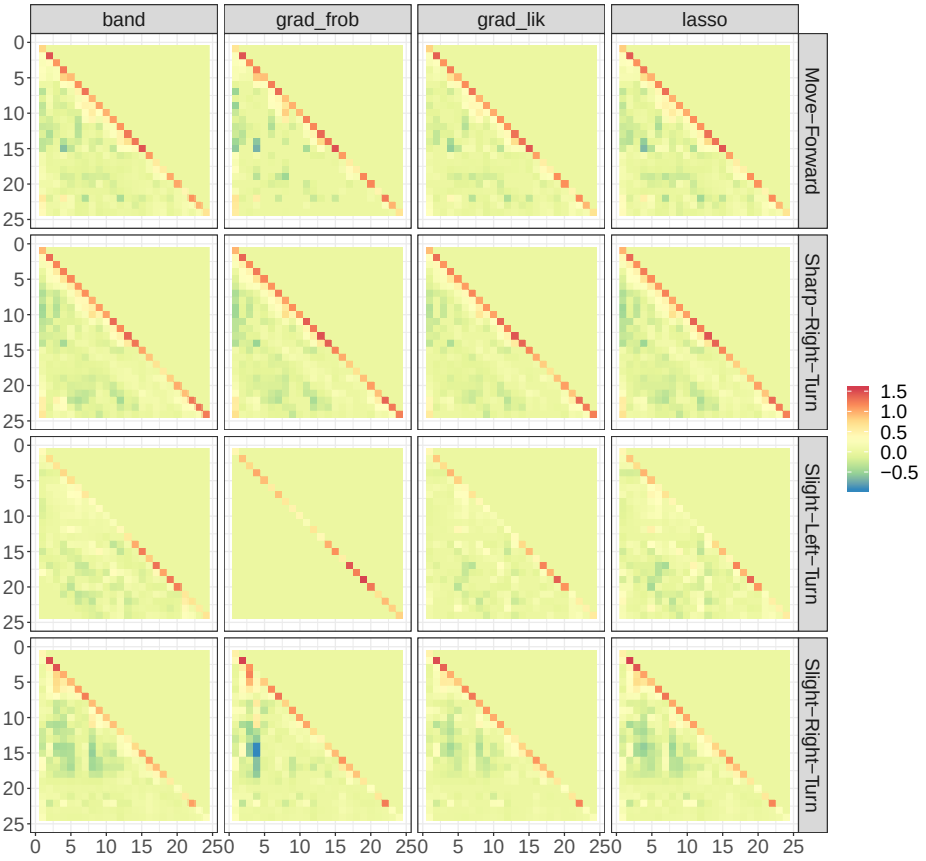



Figure 6.7: Heatmaps of the Cholesky factor of the wall-following robot navigation samples.

In this problem there is a larger sample size, and therefore data has been split into train and test, with half of the samples on each set. The classification results are shown in Table 6.2. Observe that all methods perform arguably good, in fact they achieve nearly identical accuracy. It is noticeable how competitive are lasso and grad_lik, which performed much worse in the sonar problem. Also notice that arguably the best results are obtained for the Slight-Left-Turn movement, which confirms the previous intuition

over heatmaps about sensors correctly identifying obstacles. The worst performance over all methods is for the Slight-Right-Turn movement, but is not noteworthy when compared with the rest (except for Slight-Left-Turn).

| | band | grad_frob | grad_lik | lasso |
|---|---|---|---|---|
| TNR (MF) | **0.87** | 0.86 | 0.85 | 0.84 |
| F1 (MF) | **0.64** | **0.64** | 0.61 | 0.62 |
| TNR (SHR) | **0.89** | 0.88 | 0.87 | 0.85 |
| F1 (SHR) | 0.72 | 0.72 | **0.73** | **0.73** |
| TNR (SLL) | 0.96 | 0.95 | **0.98** | **0.98** |
| F1 (SLL) | 0.67 | 0.65 | **0.72** | 0.7 |
| TNR (SLR) | 0.82 | 0.83 | 0.81 | **0.84** |
| F1 (SLR) | 0.56 | **0.57** | 0.55 | **0.57** |
| ACC | **0.66** | **0.66** | 0.65 | **0.66** |

Table 6.2: Statistics for the robot problem. MF: Move-Forward; SHR: Sharp-Right-Turn; SLL: Slight-Left-Turn; SLR: Slight-Right-Turn.

### 6.4.3 Discussion of the results

Several conclusions can be drawn from both the simulated and real experiments. Firstly, Figure 6.8 depicts the execution time for each method, where it can be seen that grad_lik is the slowest one and band is the fastest.

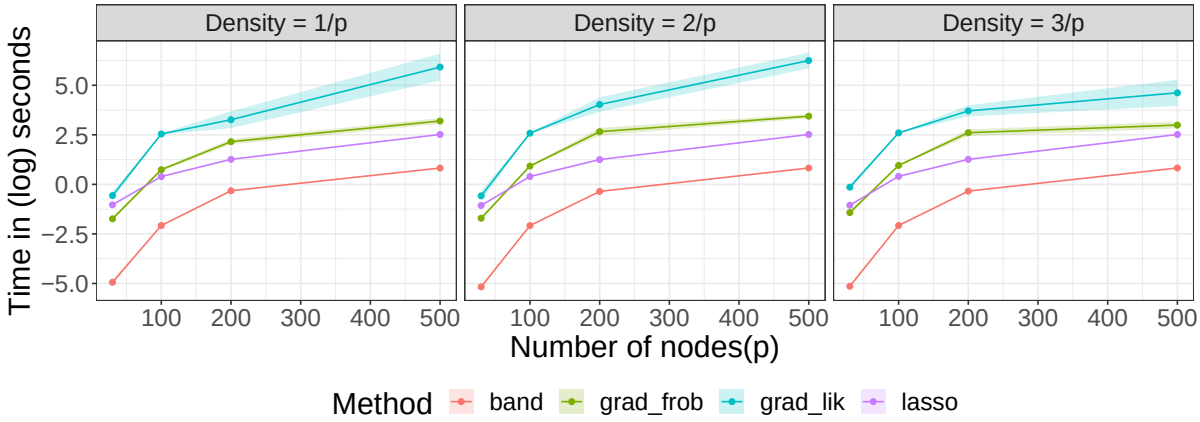



Figure 6.8: Logarithm of the execution time (in seconds) for each of the methods under evaluation. Density indicates the average proportion of lower triangular non-zero entries in the simulated $\mathbf{T}$ Cholesky factors.

Whenever there is a clear dependence between variables that are close in the ordering, such as in the sonar example, the band method could be preferred, because it is the one that more naturally approximates the structure induced in the Cholesky factor (as happened in the sonar example).

The new proposed method grad_frob has shown to be competitive both in execution time as well as in recovery results: when interested in model selection, that is, how

accurately zeros in the Cholesky factor are estimated, it yields the best results. Conversely, grad_lik has shown to be the most robust: in simulations it achieved reasonable performance even when the true covariance matrix was dense, and it also performed competitively in the sonar example, which was mostly suited for band as we discussed.

Finally, lasso has achieved overall poor results, except for the wall-following robot navigation data set. Specially, in simulations it failed to correctly recover the zero pattern in the Cholesky factor and was the numerically farthest away from the true matrix. Despite this, it is the second fastest of the four methods, so when model selection or robustness are not a concern it is a good alternative to band, since it provides more flexibility over the zero pattern in the Cholesky factor.

# Chapter 7

# Conclusions and future research

This thesis has contributed to the classical yet actively researched topic of Gaussian graphical models. The detailed list of publications by the author directly related to this thesis is:

- I. Córdoba, E. C. Garrido-Merchán, D. Hernández-Lobato, C. Bielza, and P. Larrañaga. Bayesian optimization of the PC algorithm for learning Gaussian Bayesian networks. In *Advances in Artificial Intelligence*, volume 11160 of *Lecture Notes in Artificial Intelligence*, pages 44–54. Springer, 2018a

- I. Córdoba, G. Varando, C. Bielza, and P. Larrañaga. A fast Metropolis-Hastings method for generating random correlation matrices. In *Intelligent Data Engineering and Automated Learning*, volume 11314 of *Lecture Notes in Computer Science*, pages 117–124. Springer, 2018b

- I. Córdoba, G. Varando, C. Bielza, and P. Larrañaga. A partial orthogonalization method for simulating covariance and concentration graph matrices. In *International Conference on Probabilistic Graphical Models*, volume 72 of *Proceedings of Machine Learning Research*, pages 61–72. PMLR, 2018c

- I. Córdoba, C. Bielza, and P. Larrañaga. A review of Gaussian Markov models for conditional independence. *Journal of Statistical Planning and Inference*, 206: 127–144, 2020a

- I. Córdoba, C. Bielza, P. Larrañaga, and G. Varando. Sparse Cholesky covariance parametrization for recovering latent structure in ordered data. *IEEE Access*, 8: 154614–154624, 2020b

- I. Córdoba, G. Varando, C. Bielza, and P. Larrañaga. On generating random Gaussian graphical models. *International Journal of Approximate Reasoning*, 125:240–250, 2020c

The relationship between each exposed chapter and the contributed papers outlined above is as follows:

- Chapter 2 is mainly based on the journal article Córdoba et al. (2020a); most of the content about chordal graphs is new, whereas Bayesian estimation and multiple hypothesis testing have not been incorporated to the thesis because they are not related to it.

- Chapter 3, covering Bayesian optimization of the PC algorithm, is essentially the conference article Córdoba et al. (2018a), with the modification that more emphasis has been put on explaining the PC algorithm's parameters, and less emphasis on Bayesian optimization, which is merely a tool for the thesis objectives.

- Gaussian graphical model simulation is covered in both Chapters 4 and 5. In particular, Chapter 4 is based on the conference article Córdoba et al. (2018b), with Section 4.4.2 and the proof of Proposition 4.2.1 being novel material first appearing in this thesis. Such chapter serves as an introduction to the sampling methodology that is later applied to Gaussian graphical models in Chapter 5, largely based on both the conference article Córdoba et al. (2018c) and the journal article Córdoba et al. (2020c). Proposition 5.1.1 and Algorithm 6 are novel content, and serve to further illustrate sampling problems in acyclic directed models, which were not covered in the aforementioned papers but are nevertheless relevant to the thesis content. Some figures also differ because of such unification, for example, Figures 5.10 and 5.11 merge the respective versions of Córdoba et al. (2018c) and Córdoba et al. (2020c), for better clarity and to avoid duplication.

- A relatively different topic, but also related to Gaussian graphical models and the concepts discussed throughout the thesis, is explored in Chapter 6, where the sparse covariance Cholesky parametrization is explored. This last contribution chapter contains most of what is described in the journal article Córdoba et al. (2020b).

# Future research

### Model selection with the PC algorithm: parameter tuning

Other objective measures that do not rely on knowing the true graph structure, such as Gaussian Bayesian network scores, could be explored. On the methodological side, a comparison could be carried out with alternative parameter optimization methods such as genetic algorithms. Finally, the Bayesian optimization method used is able to handle multi-objective scenarios, several constraints and noise, thereby the proposed methodology could be extended to such scenarios in Gaussian graphical models.

### Uniform Metropolis sampling of correlation matrices

Variants of the proposed Markov chain algorithm could be explored, such as the independent Metropolis or adaptive schemes. The theoretical convergence analysis of such variants is also relevant, as well as the extension of empirical convergence monitoring to other relevant quantities apart from the acceptance ratio. It would be also interesting to perform a thorough theoretical and empirical study of the similarities and differences between the proposed method and others in the literature for uniform sampling of correlation matrices.

### On Gaussian graphical model simulation

Since very promising results have been obtained when using the proposed simulation methods in a real validation scenario, it would be very interesting to explore how other

performance measures, and other model selection methods, are also affected. From the computational point of view, exploring alternatives to the modified Gram-Schmidt orthogonalization or taking into account special structures in the graph topology could reduce the complexity of partial orthogonalization. Finally, the main theoretical direction for future research would to investigate how to sample uniformly from the space $\mathcal{E}_{\mathcal{G}}^{p}$ for a non-chordal graph $\mathcal{G}$, or, conversely, for an acyclic digraph with no v-structures.

**Sparse Cholesky covariance parametrization**

As future research, the most direct and interesting derivative work would be to further analyse, both theoretically and empirically, the Gaussian graphical model extension to unordered variables of the sparse covariance Cholesky factorization model. Also in this direction, its relationship with the already established covariance graph model (Equation (5.1)), could yield interesting results. Regarding the proposed proximal gradient estimation method, other alternatives could be explored to solve the optimization problems that arise from both the likelihood and Frobenious loss, as well as try other losses related to matrices and the multivariate Gaussian distribution.

# Bibliography


T. W. Anderson. *An Introduction to Multivariate Statistical Analysis.* John Wiley & Sons, 3rd edition, 2003.

S. A. Andersson, D. Madigan, and M. D. Perlman. A characterization of Markov equivalence classes for acyclic digraphs. *Annals of Statistics*, 25(2):505–541, 04 1997.

B. Aragam and Q. Zhou. Concave penalized estimation of sparse Gaussian Bayesian networks. *Journal of Machine Learning Research*, 16:2273–2328, 2015.

E. Arvaniti and M. Claassen. Markov network structure learning via ensemble-of-forests models. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 42–51. AUAI Press, 2014.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.

O. Barndorff-Nielsen. *Information and Exponential Families in Statistical Theory.* John Wiley & Sons, 1978.

S. Basu, X. Li, and G. Michailidis. Low rank and structured modeling of high-dimensional vector autoregressions. *IEEE Transactions on Signal Processing*, 67(5):1207–1222, 2019.

A. E. Beaton. The use of special matrix operators in statistical calculus. *ETS Research Bulletin Series*, 1964(2):i–222, 1964.

J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 36(2):192–236, 1974.

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

P. Bühlmann and S. van de Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications.* Springer, 2011.

T. Cai, W. Liu, and X. Luo. A constrained $l_1$ minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011.

V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky. Latent variable graphical model selection via convex optimization. *Annals of Statistics*, 40(4):1935–1967, 08 2012.

D. Colombo and M. H. Maathuis. Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15:3921–3962, 2014.

D. R. Cox and N. Wermuth. Linear dependencies represented by chain graphs. *Statistical Science*, 8(3):204–218, 1993.

I. Córdoba, E. C. Garrido-Merchán, D. Hernández-Lobato, C. Bielza, and P. Larrañaga. Bayesian optimization of the PC algorithm for learning Gaussian Bayesian networks. In *Advances in Artificial Intelligence*, volume 11160 of *Lecture Notes in Artificial Intelligence*, pages 44–54. Springer, 2018a.

I. Córdoba, G. Varando, C. Bielza, and P. Larrañaga. A fast Metropolis-Hastings method for generating random correlation matrices. In *Intelligent Data Engineering and Automated Learning*, volume 11314 of *Lecture Notes in Computer Science*, pages 117–124. Springer, 2018b.

I. Córdoba, G. Varando, C. Bielza, and P. Larrañaga. A partial orthogonalization method for simulating covariance and concentration graph matrices. In *International Conference on Probabilistic Graphical Models*, volume 72 of *Proceedings of Machine Learning Research*, pages 61–72. PMLR, 2018c.

I. Córdoba, C. Bielza, and P. Larrañaga. A review of Gaussian Markov models for conditional independence. *Journal of Statistical Planning and Inference*, 206:127–144, 2020a.

I. Córdoba, C. Bielza, P. Larrañaga, and G. Varando. Sparse Cholesky covariance parametrization for recovering latent structure in ordered data. *IEEE Access*, 8:154614–154624, 2020b.

I. Córdoba, G. Varando, C. Bielza, and P. Larrañaga. On generating random Gaussian graphical models. *International Journal of Approximate Reasoning*, 125:240–250, 2020c.

J. N. Darroch, S. L. Lauritzen, and T. P. Speed. Markov fields and log-linear interaction models for contingency tables. *Annals of Statistics*, 8(3):522–539, 1980.

A. d'Aspremont, O. Banerjee, and L. El Ghaoui. First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 30(1):56–66, 2008.

A. P. Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 41(1):1–31, 1979.

A. P. Dempster. *Elements of Continuous Multivariate Analysis*. Adisson-Wesley, 1969.

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

P. Diaconis, S. Holmes, and M. Shahshahani. *Sampling from a manifold*, volume 10 of *Collections*, pages 102–125. Institute of Mathematical Statistics, 2013.

M. Drton. Algebraic problems in structural equation modeling. In *The 50th Anniversary of Gröbner Bases*, pages 35–86, Tokyo, Japan, 2018. Mathematical Society of Japan.

M. Drton and T. S. Richardson. Graphical methods for efficient likelihood inference in Gaussian covariance models. *Journal of Machine Learning Research*, 9:893–914, 2008.

D. Dua and C. Graff. UCI machine learning repository, 2020. URL `http://archive.ics.uci.edu/ml`.

M. L. Eaton. *Multivariate Statistics: A Vector Space Approach*. John Wiley & Sons, 1983.

D. Edwards. *Introduction to Graphical Modelling*. Springer, 2000.

P. Erdös and A. Rényi. On random graphs, I. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

P. S. Eriksen. Tests in covariance selection models. *Scandinavian Journal of Statistics*, 23(3):275–284, 1996.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

M. Frydenberg. The chain graph Markov property. *Scandinavian Journal of Statistics*, 17(4):333–353, 1990.

E. C. Garrido-Merchán and D. Hernández-Lobato. Predictive entropy search for multi-objective Bayesian optimization with constraints. *Neurocomputing*, 361:50–68, 2019.

E. C. Garrido-Merchán and D. Hernández-Lobato. Dealing with categorical and integer-valued variables in Bayesian optimization with Gaussian processes. *Neurocomputing*, 380:20–35, 2020.

R. Goudie and S. Mukherjee. A Gibbs sampler for learning DAGs. *Journal of Machine Learning Research*, 17:1032–1070, 2016.

G. R. Grimmett. A theorem about random fields. *Bulletin of the London Mathematical Society*, 5(1):81–84, 1973.

J. Hausser and K. Strimmer. Entropy inference and the James-Stein estimator, with application to nonlinear gene association networks. *Journal of Machine Learning Research*, 10:1469–1484, 2009.

J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems*, pages 918–926, 2014.

J. Honorio, D. Samaras, I. Rish, and G. Cecchi. Variable selection for Gaussian graphical models. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 538–546. PMLR, 2012.

R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2012.

V. Isham. An introduction to spatial point processes and Markov random fields. *International Statistical Review*, 49(1):21–43, 1981.

S. T. Jensen. Covariance hypotheses which are linear in both the covariance and the inverse covariance. *Annals of Statistics*, 16(1):302–322, 1988.

H. Joe. Generating random correlation matrices based on partial correlations. *Journal of Multivariate Analysis*, 97(10):2177–2189, 2006.

M. Kalisch and P. Bühlmann. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research*, 8:613–636, 2007.

M. Kalisch and P. Bühlmann. Robustification of the PC-algorithm for directed acyclic graphs. *Journal of Computational and Graphical Statistics*, 17(4):773–789, 2008.

M. Kalisch, M. Mächler, D. Colombo, M. H. Maathuis, and P. Bühlmann. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11):1–26, 2012.

G. Kauermann. On a dualization of graphical Gaussian models. *Scandinavian Journal of Statistics*, 23(1):105–116, 1996.

R. Kindermann and J. L. Snell. *Markov Random Fields and their Applications*. American Mathematical Society, 1980.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

N. Krämer, J. Schäfer, and A.-L. Boulesteix. Regularized estimation of large-scale gene association networks using graphical Gaussian models. *BMC Bioinformatics*, 10(1): 384, 2009.

C. Lam and J. Fan. Sparsistency and rates of convergence in large covariance matrix estimators. *Annals of Statistics*, 37(6B):4254–4278, 2009.

M. Laurent and S. Poljak. On the facial structure of the set of correlation matrices. *SIAM Journal on Matrix Analysis and Applications*, 17(3):530–547, 1996.

S. Lauritzen and K. Sadeghi. Unifying Markov properties for graphical models. *Annals of Statistics*, 46(5):2251–2278, 2018.

S. L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

S. L. Lauritzen, A. P. Dawid, B. N. Larsen, and H.-G. Leimer. Independence properties of directed Markov fields. *Networks*, 20(5):491–505, 1990.

D. Lewandowski, D. Kurowicka, and H. Joe. Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis*, 100(9): 1989–2001, 2009.

Y. Lin, S. Zhu, D. Lee, and B. Taskar. Learning sparse Markov network structure via ensemble-of-trees models. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 360–367. PMLR, 2009.

M. Maathuis, M. Drton, S. Lauritzen, and M. Wainwright, editors. *Handbook of Graphical Models*. CRC Press, 2018.

E. Makalic and D. F. Schmidt. An efficient algorithm for sampling from $sin^k(x)$ for generating random correlation matrices. *Communications in Statistics - Simulation and Computation*, In press, 2020.

K. Mardia and P. Jupp. *Directional Statistics*. Wiley, 1999.

C. Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, pages 403–410. Morgan Kaufmann, 1995.

N. Meinshausen. A note on the lasso for Gaussian graphical model selection. *Statistics & Probability Letters*, 78(7):880–884, 2008.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34(3):1436–1462, 2006.

N. Meinshausen and B. Yu. Lasso-type recovery of sparse representations for high-dimensional data. *Annals of Statistics*, 37(1):246–270, 2009.

J. Moussouris. Gibbs and Markov random systems with constraints. *Journal of Statistical Physics*, 10(1):11–33, 1974.

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

J. Pearl and A. Paz. Graphoids: A graph-based logic for reasoning about relevance relations. In *Advances in Artificial Intelligence*, volume 2, pages 357–363. Elsevier, 1987.

K. Petersen and M. Pedersen. *The Matrix Cookbook*. Technical University of Denmark, 2008.

M. Pourahmadi. Joint mean-covariance models with applications to longitudinal data: Unconstrained parameterisation. *Biometrika*, 86(3):677–690, 1999.

M. Pourahmadi and X. Wang. Distribution of random correlation matrices: Hyperspherical parameterization of the Cholesky factor. *Statistics & Probability Letters*, 106:5–12, 2015.

T. M. Pukkila and C. R. Rao. Pattern recognition based on scale invariant discriminant functions. *Information Sciences*, 45(3):379–389, 1988.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, 2020. URL `https://www.R-project.org/`.

P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing $l_1$-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

J. M. Robins, R. Scheines, P. Spirtes, and L. Wasserman. Uniform consistency in causal inference. *Biometrika*, 90(3):491–515, 2003.

A. J. Rothman, P. J. Bickel, E. Levina, and J. Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

A. J. Rothman, E. Levina, and J. Zhu. A new approach to Cholesky-based covariance regularization in high dimensions. *Biometrika*, 97(3):539–550, 2010.

A. Roverato. Cholesky decomposition of a hyper inverse Wishart matrix. *Biometrika*, 87 (1):99–112, 2000.

M. Rudelson and R. Vershynin. Smallest singular value of a random rectangular matrix. *Communications on Pure and Applied Mathematics*, 62(12):1707–1739, 2009.

J. Schäfer and K. Strimmer. An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764, 2005a.

J. Schäfer and K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(1):Article 32, 2005b.

M. Scutari. Learning Bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35(3):1–22, 2010.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2016.

A. Shojaie and G. Michailidis. Penalized likelihood methods for estimation of sparse high-dimensional directed acyclic graphs. *Biometrika*, 97(3):519–538, 2010.

J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012.

P. Spirtes and C. Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1):62–72, 1991.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. The MIT Press, 2000.

I. Stojkovic, V. Jelisavcic, V. Milutinovic, and Z. Obradovic. Fast sparse Gaussian Markov random fields learning based on Cholesky factorization. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 2758–2764, 2017.

M. Studený. Conditional independence and basic Markov properties. In *Handbook of Graphical Models*, pages 1–38. CRC Press, 2018.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 58(1):267–288, 1996.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

C. Uhler. Gaussian graphical models. In *Handbook of Graphical Models*, pages 235–256. CRC Press, 2018.

C. Uhler, G. Raskutti, P. Bühlmann, and B. Yu. Geometry of the faithfulness assumption in causal inference. *Annals of Statistics*, 41(2):436–463, 2013.

S. van de Geer and P. Bühlmann. On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3:1360–1392, 2009.

S. van de Geer and P. Bühlmann. $\ell_0$-penalized maximum likelihood for sparse directed acyclic graphs. *Annals of Statistics*, 41(2):536–567, 2013.

G. Varando and N. R. Hansen. Graphical continuous Lyapunov models. In *Proceedings of the 36th conference on Uncertainty in Artificial Intelligence*, volume 124 of *Proceedings of Machine Learning Research*, pages 989–998. PMLR, 2020.

T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, pages 255–270. AUAI Press, 1991.

N. Wermuth. Analogies between multiplicative models in contingency tables and covariance selection. *Biometrics*, 32(1):95–108, 1976a.

N. Wermuth. Model search among multiplicative models. *Biometrics*, 32(2):253–263, 1976b.

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

N. Wermuth and K. Sadeghi. Sequences of regressions and their independences. *TEST*, 21(2):215–252, 2012.

N. Wermuth, D. Cox, and G. M. Marchetti. Covariance chains. *Bernoulli*, 12(5):841–862, 2006.

S. S. Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *Annals of Mathematical Statistics*, 9(1):60–62, 1938.

S. Wright. The method of path coefficients. *Annals of Mathematical Statistics*, 5(3): 161–215, 1934.

D. Yatsenko, K. Josić, A. S. Ecker, E. Froudarakis, R. J. Cotton, and A. S. Tolias. Improved estimation and interpretation of correlations in neural circuits. *PLOS Computational Biology*, 11(3):1–28, 2015.

G. Yu and J. Bien. Learning local dependence in ordered data. *Journal of Machine Learning Research*, 18:1–60, 2017.

M. Yuan and Y. Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 2007a.

M. Yuan and Y. Lin. On the non-negative garrotte estimator. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):143–161, 2007b.

G. U. Yule. On the theory of correlation for any number of variables, treated by a new system of notation. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 79(529):182–193, 1907.

J. Zhang and P. Spirtes. Strong faithfulness and uniform consistency in causal inference. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, pages 632–639. Morgan Kaufmann, 2003.

P. Zhao and B. Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

X. Zheng, B. Aragam, P. Ravikumar, and E. P. Xing. DAGs with NO TEARS: Continuous optimization for structure learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 9492–9503. Curran Associates Inc., 2018.

M. Zorzi and R. Sepulchre. AR identification of latent-variable graphical models. *IEEE Transactions on Automatic Control*, 61(9):2327–2340, 2016.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.