

Faculté des sciences

Transfer Learning for Penalized Gaussian Graphical Models

Author: **Lucas ELVIRA**
Supervisor: **Eugen PIRCALABELU**
Reader: **Rainer VON SACHS**
Academic year 2023–2024
Master [120] en statistique, orientation biostatistiques

Acknowledgment.

First and more importantly, I would like to express my deepest gratitude to my thesis supervisor, Eugen Pircalabelu, who has been more than a support to me during this journey, his presence was always comforting and he made sure that I was always on the right path.

I am also grateful to Rainer Von Sachs who has accepted to be the reader of my thesis and Ensiyeh Nezakati who has provided me a part of their work for the thesis.

A special thanks to my family and friends who were always supportive and encouraging, they gave me this opportunity and I would not be able to finish this thesis without them. I am especially grateful to Marine Lambert for her constant presence and advice when I needed it, this work would not be achieved without her help.

Lastly, I would like to thank all those who, in a way or another, have contributed to this thesis.

Contents

1	Introduction	1
2	Notations	3
3	Graphical Models	9
3.1	Graph Theory : Introduction	9
3.2	Gaussian Graphical Models	11
3.3	Estimating a GGM	13
3.4	Penalization	21
3.4.1	Lasso-type estimator	22
3.4.2	Glasso estimator	24
3.4.3	Edge detection method for the Glasso estimator	25
4	Transfer Learning on GGM	31
4.1	The CLIME Estimator	32
4.2	Transfer Learning	33
4.3	Edge detection	38
4.4	Rates of convergence	40
4.4.1	Rates of convergence : CLIME and Trans-CLIME	42
4.4.2	Rates of convergence : Glasso	44
5	Simulations	45
5.1	Banded matrix	48
5.2	Random matrix	54
5.3	Banded matrix when $ A = 5$	59
5.4	Modified Banded matrix	62
5.5	Conclusion	65
6	Application : GTEx brain tissues data	67
6.1	Preprocessing and PCA	68
6.2	Estimation	71
6.3	Conclusion	79

7 Conclusion	81
A Appendix	87
A.1 Karush Kuhn Tucker conditions	87

Chapter 1

Introduction

In a world where statistical studies are rapidly multiplying, relying only on one main study for statistical research could not be the most effective approach anymore. What if some valuable method exists that could allow to exploit all that available information ? As known and studied from several decades now, Graphical Models are a perfect choice to investigate the relationships between different variables and their underlying structure, especially Gaussian Graphical Models (see Lauritzen, 1996; Ke Wang, 2023; Bhushan et al., 2019).

In many fields such as genetics, chemistry, medicine, it becomes common to be in a high-dimensional setting. In this situation, the use of penalization on Gaussian Graphical Models is required to keep only the important information and detect the most important interactions. Throughout the years, several methods have emerged to perform penalization strategies, such as the lasso-type estimator (Yuan & Lin, 2007), the Graphical Lasso (Friedman et al., 2008), the SCAD estimator (Fan & Li, 2001) or the CLIME estimator (Cai et al., 2011).

However, these models only rely on data from one study, the target study, which may be insufficient in high-dimensional situations. It would be more relevant to use similar auxiliary studies and exploit their information to enhance the performance of the model. This is where Transfer Learning offers a powerful strategy by leveraging in an efficient way the information given by the auxiliary studies, where one could enhance in an elegant fashion the performance of the estimators. The use of Transfer Learning for Gaussian Graphical Models could be an important contribution to the actual field in high-dimensional studies where sample sizes are often low. This thematic was introduced by Li et al. (2022) and is the main study of this work.

In this thesis, we aim to assess the performance of this Transfer Learning-based estimator on penalized Gaussian Graphical Models, using different strategies such as simulation frameworks and a real-world application.

For this purpose, the thesis is organized as follows. The notations used throughout are presented in Chapter 2. In Chapter 3, we introduce the Graphical Models and specifically the Gaussian Graphical Models theory followed by discussions on the possible penalization methods. Next, the novel Transfer Learning method is explained and the Trans-CLIME estimator is introduced in Chapter 4. In Chapter 5, several simulations are performed and discussed to explore the performance of the estimator against two competitors, namely the CLIME and Glasso estimators. Finally, in Chapter 6 an illustration of the performance of this estimation method is performed on brain tissues data, coming from the GTEx database. Chapter 7 concludes this thesis with a general discussion on the results, encountered obstacles, limitations of this work and possible extensions.

Chapter 2

Notations

Here is an exhaustive list of all the notations and abbreviations used in this Master's thesis by order of appearance :

- $\mathcal{G}$: A graph.
- $\mathcal{V}$: Set of vertices of a graph.
- $\mathcal{E}$: Edges of a graph.
- $X_i \sim Y_i$ or $[X_i Y_i] \in \mathcal{E}$: Variables X_i and Y_i are connected by an edge and called adjacent.
- $X_i \perp\!\!\!\perp Y_i$ denotes the marginal independence between variables X_i and Y_i .
- $X_i \perp\!\!\!\perp Y_i | Z_i$ denotes the conditional independence between variables X_i and Y_i given the variable Z_i .
- $u \subseteq U$: the set u is a subset of U containing at most all the elements of U . The complementary of a set is expressed by $\bar{U}$.
- X_i denotes the variable i of the random vector $X \in \mathbb{R}^p$.
- μ_X : mean vector of random vector X .
- $\Sigma = \{\sigma_{ij}\}$: Covariance matrix.
- $\Omega = \Sigma^{-1} = \{\omega_{ij}\}$: Precision matrix.
- Σ_{XY} : Covariance matrix between random vector Y and Z .
- $\mu_Y(Z)$: Conditional mean of Y given Z .

- $\Sigma_{Y|Z}$: Conditional covariance matrix of Y given Z .
- ρ : Correlation.
- $\rho_{X|Y}$: Conditional correlation.
- Let A be a square matrix, $|A|$ denotes the determinant of the matrix A and $\text{tr}(A)$ denotes the trace of matrix A .
- $\mathbf{x}_i \in \mathbb{R}^p$ is a realization of the random vector X .
- $\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$ is the empirical mean.
- $S = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^t$ is the empirical covariance matrix.
- pdf : probability density function.
- $\bar{\mathcal{G}}$ denotes the complementary graph of $\mathcal{G}$.
- $\xrightarrow{\mathcal{L}}$ denotes convergence in distribution.
- $\mathbb{K}(\cdot)$: indicator function, resulting in 1 if the statement is true and 0 otherwise.
- $\Delta^{(k)}$ is the divergence matrix for study k .
- $\inf$: denotes the infimum of a set, the highest value in the real space which is lower or equal to every values in the set (for example in the real space, the infimum of the set $U = \{1, 4, -2, 6\}$ is -2).
- $\Lambda_{\min}(\cdot)$ and $\Lambda_{\max}(\cdot)$ denotes the lowest and largest eigenvalue of a matrix respectively.
- $u \sim U(a, b)$: variable u follows an uniform distribution with parameters a and b .
- A_+ : denotes the positive-definite projection of a matrix.
- $C_1, C_2, \dots$ and $c_1, c_2, \dots$ are general constants not linked at all used in different places.

The Kronecker product between two matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$ denoted by $A \otimes B$ is defined by a block-matrix $mp \times nq$ and is computed as

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & \dots & a_{11}b_{1q} & \dots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \dots & a_{11}b_{2q} & \dots & a_{1n}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \dots & a_{11}b_{pq} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \dots & a_{m1}b_{pq} & \dots & a_{mn}b_{pq} \end{pmatrix}.$$

Let a be a p -vector ($a \in \mathbb{R}^p$) and A a $p \times q$ matrix ($A \in \mathbb{R}^{p \times q}$). Recall that the general notation for a ℓ_q vector norm is

$$|a|_q = \left(\sum_{j=1}^p |a_j|^q \right)^{1/q}.$$

For example we denote the ℓ_1 vector norm as

$$|a|_1 = \sum_{j=1}^p |a_j|,$$

and the ℓ_2 vector norm as

$$|a|_2 = \sqrt{\sum_{j=1}^p a_j^2},$$

and the ℓ_∞ norm of a vector as

$$|a|_\infty = \max_{1 \leq j \leq p} |a_j|.$$

These norms are intuitive as they compute the lengths of the vectors in different ways. For example having two vectors, $v = (1, 1)^t$ and $u = (2, 2)^t$, it is easy to see that u has higher ℓ_1 and ℓ_2 norms than v .

For matrices this comparison is more complicated as a matrix can be seen to be composed by several vectors (for each column) but the idea is to use vector norms to compute the matrix norms. The common notation for an element-wise ℓ_q matrix norm is

$$\|A\|_q = \sup_{|x|_q=1} |Ax|_q.$$

For example the *spectral norm* (or ℓ_2 matrix norm) is defined as

$$\|A\|_2 = \sup_{|x|_2=1} |Ax|_2 = \sqrt{\Lambda_{\max}(A^t A)}.$$

The ℓ_1 and ℓ_∞ matrix norms are

$$\begin{aligned} \|A\|_{L_1} &= \sup_{|x|_1=1} |Ax|_1 = \max_{1 \leq j \leq q} \sum_{i=1}^p |a_{ij}|, \\ \|A\|_{L_\infty} &= \|A^t\|_{L_1} = \max_{1 \leq i \leq p} \sum_{j=1}^q |a_{ij}|, \end{aligned}$$

which are just the maximum ℓ_1 norm of the columns and the maximum ℓ_1 norm of the rows respectively, where a_{ij} is the $(i, j)^{th}$ element of A . The idea behind these matrix norms is that by fixing the norm of a q -vector x , we use our matrix to compute how high the ℓ_q norm of the given vector Ax can be. Figure 2.1 illustrates the idea behind these norms and how we attain a higher norm with vector x_2 .

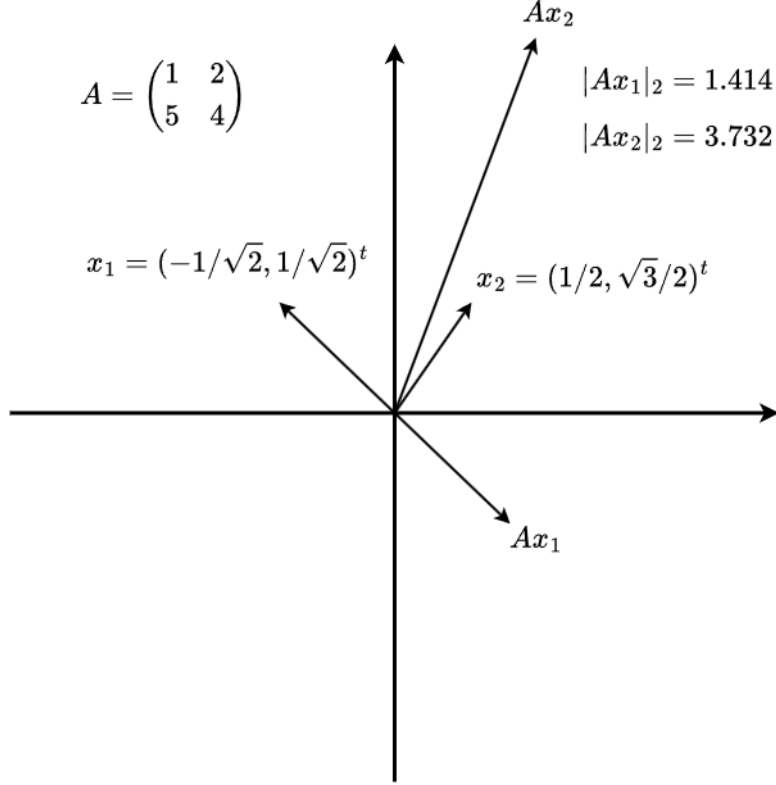


Figure 2.1: Illustration of the computation of the ℓ_2 matrix norm for two given vectors, inspired by Ford (2015, Chapter 7).

Other possible norms have been introduced to compare matrices like the element-wise ℓ_∞ norm (the maximum absolute element of A)

$$\|A\|_\infty = \max_{1 \leq i, j \leq p} |a_{ij}|,$$

the element-wise ℓ_1 norm

$$\|A\|_1 = \sum_{i=1}^p \sum_{j=1}^q |a_{ij}|,$$

and the Frobenius norm

$$\|A\|_F = \sqrt{\sum_{i=1}^p \sum_{j=1}^q a_{ij}^2}.$$

We denote the Big-O notation $f(n) = \mathcal{O}(g(n))$, for which $\exists c > 0$, $\exists n_0$ such that $\forall n \geq n_0$, we have $|f(n)| \leq cg(n)$, or asymptotically we have $\lim_{n \rightarrow \infty} \sup \frac{f(n)}{g(n)} < \infty$ (converge to a finite constant).

We denote the little-o notation $f(n) = o(g(n))$, for which $\forall c, \exists n_0$ such that $\forall n \geq n_0$, we have $|f(n)| < cg(n)$, or asymptotically we have $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$ (convergence to 0).

The main difference between the Big-O and little-o notation is that in the little-o setting, $f(n)$ is dominated asymptotically by $g(n)$ while this is not the case in the Big-O setting.

Similarly, we denote the Big-Omega notation $f(n) = \Omega(g(n))$, for which $\exists c, \exists n_0$ such that $\forall n \geq n_0$, we have $f(n) \geq cg(n)$, or asymptotically we have $\lim_{n \rightarrow \infty} \inf \frac{f(n)}{g(n)} > 0$ (converge to a finite constant larger than 0).

Eventually, we denote the Big-O in probability notation as $S_n = \mathcal{O}_p(a_n)$, where S_n is a set of random variables and a_n the corresponding set of constants. This notation says that, $\forall \epsilon > 0, \exists M, n_0 > 0$, such that

$$P(|\frac{S_n}{a_n}| > M) < \epsilon, \forall n > n_0.$$

Chapter 3

Graphical Models

Before going into the Transfer Learning method which is the main interest in this thesis, we need to understand how Graphical Models work and how Gaussian Graphical Models can be estimated.

In graphical modelling one focuses on the variables, more precisely one is looking for independencies between variables. Before presenting Graphical Models, we have to introduce the concept of a Graph.

3.1 Graph Theory : Introduction

This part mainly contains theory from Edwards (2000) and Koller and Friedman (2009).

A graph, denoted by $\mathcal{G}$, is composed of two elements, the set of vertices $\mathcal{V}$ which maps in our case the variables and the set of edges $\mathcal{E}$ which represent the links between the vertices. Mathematically we denote $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

As we said earlier, in graphical modelling we focus on independencies between variables. More precisely, we focus on **conditional** independencies, that is, some variables X and Y are independent given other variables, denoted by $X \perp\!\!\!\perp Y \mid \text{other variables}$. The edges of the graph are providing information on the conditional independence between two variables given the rest. If two nodes are not connected by an edge, then these variables are conditionally independent given the rest. This is called the *pairwise Markov property*.

These edges can be directed, meaning that the dependence is caused by one of the two variables and that there is an influence (e.g. heredity between parents and children), or they can be undirected, i.e. there isn't any direction of dependence. In this thesis we will only focus on undirected graphs.

The graphs can be represented in a diagram as in Figure 3.1. In this example we illustrate a graph with 5 vertices and 6 edges showing which variables are conditionally dependent or independent. For instance, $D \perp\!\!\!\perp A|E$ but $D \not\perp\!\!\!\perp E$.

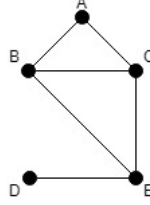


Figure 3.1: Graph with 5 vertices and 6 edges. Figure inspired from Edwards (2000).

Two vertices are said to be adjacent if they have an edge between them. In our example A and B are adjacent and this is denoted by $A \sim B$ or $[AB] \in \mathcal{E}$. If all the pairs of vertices are connected, we call the graph *complete*.

Any subset $u \subseteq \mathcal{V}$ induces a subgraph of $\mathcal{G}$ denoted as $\mathcal{G}_u = (u, \mathcal{F})$ where $\mathcal{F}$ contains the edges $\mathcal{E}$ for which both vertices are contained in u . This subset is called *complete* if the subgraph it produces is also *complete*. In our example the subset $\{A, B, C\}$ is *complete* but the subset $\{A, B, C, E\}$ is not. For three subsets u, v and w , we say that v *separates* u and w if all the edges from u to w pass through v . In our example, $v = \{E\}$ separates $u = \{A, B, C\}$ and $w = \{D\}$. If a set v separates two sets u and w we then have that $u \perp\!\!\!\perp w|v$. This is called the *Global Markov's property*.

As we will see later, there is a link between the MLE computation of a given model and the cliques of the corresponding graph. A clique of a graph is a subset u of $\mathcal{V}$ for which the subset is *complete* but any other subset v containing u is not *complete*. In other words, $u \subset v$ with u being *complete* but not v . In our graph example, the cliques of the graph are the subgraph formed by nodes $\{A, B, C\}$, $\{B, C, E\}$ and $\{D, E\}$ because any subgraph containing these ones is not *complete* anymore (for example the subset $\{A, B, C, E\}$).

Eventually, it is necessary to recall that conditional independence does not mean marginal independence between the variables (the inverse is nonetheless true). One has to interpret pairs of vertices in a given graph taking into account the other vertices present in it to not make false conclusions.

3.2 Gaussian Graphical Models

With these introductory concepts in mind, one can find a link between graphs and Normally distributed variables. Let X be a p -dimensional random vector (r.ve.) following a multivariate Normal distribution $N(\mu, \Sigma)$ where

$$\mu = (\mu_1, \dots, \mu_p)^T,$$

$$\Sigma = \begin{pmatrix} \sigma_{11} & \dots & \sigma_{1p} \\ \vdots & \ddots & \vdots \\ \sigma_{p1} & \dots & \sigma_{pp} \end{pmatrix},$$

are the mean and the covariance matrix. One then could be interested in the covariance matrix and on its zero elements denoting the variables for which the covariance (and so the correlation) are null meaning they are (marginally) independent.

When dealing with data it is more interesting to look for conditional independencies than marginal independencies, as two variables can be marginally dependent but independent given the value of other variables.

We will show that the inverse of the covariance matrix, also called precision matrix $\Omega = \Sigma^{-1}$, can be used to determine the conditional independencies between the variables. First, let's split the r.ve X into two r.ve Y and Z (i.e. $(Y^t, Z^t)^t = X$). In this case the multivariate normal can be rewritten as

$$\begin{pmatrix} Y \\ Z \end{pmatrix} \sim N \left(\begin{pmatrix} \mu_Y \\ \mu_Z \end{pmatrix}, \begin{pmatrix} \Sigma_{YY} & \Sigma_{YZ} \\ \Sigma_{ZY} & \Sigma_{ZZ} \end{pmatrix} \right),$$

with $\Sigma_{ZY} = \Sigma_{YZ}^t$, the covariance matrix between the variables in Y and Z (i.e. $\text{Cov}(Y, Z)$). Then recall that the conditional distribution of any r.ve Y given Z is also a multivariate normal denoted by $Y|Z \sim N(\mu_Y(Z), \Sigma_{Y|Z})$ where

$$\mu_Y(Z) = E[Y|Z] = \mu_Y + \Sigma_{YZ}\Sigma_{ZZ}^{-1}(Z - \mu_Z),$$

is the conditional mean of Y given Z and

$$\Sigma_{Y|Z} = V(Y|Z) = \Sigma_{YY} - \Sigma_{YZ}\Sigma_{ZZ}^{-1}\Sigma_{ZY},$$

is the conditional covariance matrix of Y given Z . We can then show that the inverse precision matrix of the r.ve Y , Ω_{YY}^{-1} , is equal to the conditional covariance matrix $\Sigma_{Y|Z}$.

From the definition of the precision matrix, we have

$$\Sigma \Omega = I_p \Leftrightarrow \begin{pmatrix} \Sigma_{YY} & \Sigma_{YZ} \\ \Sigma_{ZY} & \Sigma_{ZZ} \end{pmatrix} \begin{pmatrix} \Omega_{YY} & \Omega_{YZ} \\ \Omega_{ZY} & \Omega_{ZZ} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix},$$

from which these two equations are obtained:

$$\Sigma_{YY}\Omega_{YY} + \Sigma_{YZ}\Omega_{ZY} = I, \quad (3.2.1)$$

$$\Sigma_{ZY}\Omega_{YY} + \Sigma_{ZZ}\Omega_{ZY} = 0, \quad (3.2.2)$$

and eventually by multiplying equation (3.2.1) by Ω_{YY}^{-1} to the right and reformulating equation (3.2.2), we have

$$\Sigma_{YY}\Omega_{YY}\Omega_{YY}^{-1} + \Sigma_{YZ}\Omega_{ZY}\Omega_{YY}^{-1} = \Omega_{YY}^{-1} \Leftrightarrow \Sigma_{YY} + \Sigma_{YZ}\Omega_{ZY}\Omega_{YY}^{-1} = \Omega_{YY}^{-1},$$

$$\Sigma_{ZZ}\Omega_{ZY} = -\Sigma_{ZY}\Omega_{YY} \Leftrightarrow \Omega_{ZY} = -\Sigma_{ZZ}^{-1}\Sigma_{ZY}\Omega_{YY},$$

plugging in (3.2.2) into (3.2.1) :

$$\Omega_{YY}^{-1} = \Sigma_{YY} - \Sigma_{YZ}\Sigma_{ZZ}^{-1}\Sigma_{ZY}\Omega_{YY}\Omega_{YY}^{-1},$$

eventually we get

$$\Omega_{YY}^{-1} = \Sigma_{YY} - \Sigma_{YZ}\Sigma_{ZZ}^{-1}\Sigma_{ZY}.$$

This last result shows that only by looking at the precision matrix, do we get the information on the possible conditional independencies between variables. Indeed, if for example we are interested in variables X_2 and X_3 given the other variables, the conditional covariance matrix is

$$\Omega_{X_2X_3}^{-1} = \begin{pmatrix} \omega_{22} & \omega_{23} \\ \omega_{32} & \omega_{33} \end{pmatrix}^{-1} = \frac{1}{(\omega_{22}\omega_{33} - \omega_{23}^2)} \begin{pmatrix} \omega_{33} & -\omega_{32} \\ -\omega_{23} & \omega_{22} \end{pmatrix},$$

as $\omega_{23} = \omega_{32}$. From this formula the conditional covariance between X_2 and X_3 is $\frac{-\omega_{23}}{(\omega_{22}\omega_{33} - \omega_{23}^2)}$ and the conditional correlation is

$$\rho_{23|1,4,\dots,p} = \frac{\text{Cov}(X_2, X_3|X_1, X_4, \dots, X_p)}{\sqrt{V(X_2|X_1, X_4, \dots, X_p)}\sqrt{V(X_3|X_1, X_4, \dots, X_p)}} = \frac{-\omega_{32}}{\sqrt{\omega_{22}}\sqrt{\omega_{33}}}.$$

It is known that for multivariate normally distributed variables, we have independence (in this case conditional independence) if and only if the conditional correlation between two variables is equal to 0, i.e. when $\omega_{32} = 0$. In the general case, we have conditional independence between the variables X_k and X_l if $\omega_{kl} = 0$.

To relate a graph with its corresponding normal model, we look for null values in the precision matrix showing which edges are not present in the graph. For example given this precision matrix

$$\Omega = \begin{pmatrix} \omega_{11} & \omega_{12} & \omega_{13} & \omega_{14} \\ \omega_{21} & \omega_{22} & 0 & 0 \\ \omega_{31} & 0 & \omega_{33} & \omega_{34} \\ \omega_{41} & 0 & \omega_{43} & \omega_{44} \end{pmatrix}$$

the corresponding graph will be

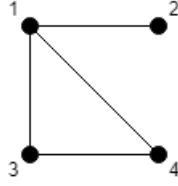


Figure 3.2: Graph corresponding to the precision matrix Ω . Figure inspired from Edwards (2000).

As depicted in the precision matrix and in the graph, the conditional independence statements are $X_2 \perp\!\!\!\perp X_3 | X_1, X_4$ and $X_2 \perp\!\!\!\perp X_4 | X_3, X_1$. What the graph tells us more is that knowing X_1 , adding X_3 or X_4 does not give us more information on X_2 as X_1 separates X_2 from the other variables.

3.3 Estimating a GGM

To estimate a GGM and more importantly the precision matrix, we will use the Maximum Likelihood Estimation method to find a possible estimator. Recall that the density function of a multivariate normal distribution is given by

$$f_X(x) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} \exp \left(-(x - \mu)^t \Sigma^{-1} (x - \mu) / 2 \right),$$

where (to refresh) p is the number of variables and $|\cdot|$ denotes the determinant of a matrix. If one now has n i.i.d. samples $\mathbf{x}_1, \dots, \mathbf{x}_n$ each one coming from this distribution, we can create the likelihood function $L(\mu, \Sigma)$,

$$L(\mu, \Sigma) = (2\pi)^{-np/2} |\Sigma|^{-n/2} \exp \left(- \sum_{i=1}^n (\mathbf{x}_i - \mu)^t \Sigma^{-1} (\mathbf{x}_i - \mu) / 2 \right).$$

To obtain the MLE based on this model, the log-likelihood function $l(\mu, \Sigma)$ is preferred.

$$l(\mu, \Sigma) = -\frac{np}{2} \log 2\pi - \frac{n}{2} \log |\Sigma| - \frac{\sum_{i=1}^n (\mathbf{x}_i - \mu)^t \Sigma^{-1} (\mathbf{x}_i - \mu)}{2}.$$

The following demonstration is derived in Taboga (2021). To get the MLE of the parameters we need to maximize the log-likelihood by taking partial derivatives with respect to the parameters and setting them to 0. Before this, we need to state some important algebraic lemmas to understand the next derivation.

Lemma 3.1. *Let A , B and C be three matrices, we characterize the cycling permutation of the trace by*

$$\text{tr}(ABC) = \text{tr}(CAB) = \text{tr}(BCA).$$

Lemma 3.2. *Let A and B be two matrices, then the gradient of the trace of the product between these matrices with respect to A is*

$$\nabla_A \text{tr}(AB) = B^t.$$

Lemma 3.3. *The gradient of the logarithm of the determinant of A is*

$$\nabla_A \log |A| = (A^{-1})^t.$$

Lemma 3.4. *Let x be a p -dimensional r.v.e and A a $p \times p$ matrix, then*

$$\nabla_x (x^t A x) = 2Ax.$$

We start by deriving a MLE for the mean vector μ . Let

$$\begin{aligned} \nabla_\mu l(\mu, \Sigma) &= \nabla_\mu \left[-\frac{np}{2} \log 2\pi - \frac{n}{2} \log |\Sigma| - \frac{\sum_{i=1}^n (\mathbf{x}_i - \mu)^t \Sigma^{-1} (\mathbf{x}_i - \mu)}{2} \right] \\ &= -\frac{1}{2} \nabla_\mu \left[\sum_{i=1}^n (\mathbf{x}_i - \mu)^t \Sigma^{-1} (\mathbf{x}_i - \mu) \right]. \end{aligned}$$

By Lemma 3.4,

$$\nabla_\mu l(\mu, \Sigma) = -\Sigma^{-1} \sum_{i=1}^n (\mathbf{x}_i - \mu),$$

and by setting this last equation to 0 and since $\Sigma \succ 0$, we get

$$\begin{aligned} \nabla_\mu l(\mu, \Sigma) = 0 &\Leftrightarrow \sum_{i=1}^n (\mathbf{x}_i - \mu) = 0 \\ &\Leftrightarrow \sum_{i=1}^n \mathbf{x}_i - n\mu = 0 \\ &\Leftrightarrow \mu = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i = \bar{\mathbf{x}}, \end{aligned}$$

hence the MLE for μ is $\bar{X}$.

Now let us define the MLE for Σ . It is easier to take partial derivatives with respect to the precision matrix Σ^{-1} instead of Σ to obtain the MLE. We first perform a transformation by denoting $\log |\Sigma| = -\log |\Sigma^{-1}|$ since $|\Sigma^{-1}| = \frac{1}{|\Sigma|}$.

$$\begin{aligned}\nabla_{\Sigma^{-1}} l(\mu, \Sigma^{-1}) &= \nabla_{\Sigma^{-1}} \left[-\frac{np}{2} \log 2\pi + \frac{n}{2} \log |\Sigma^{-1}| - \frac{\sum_{i=1}^n (\mathbf{x}_i - \mu)^t \Sigma^{-1} (\mathbf{x}_i - \mu)}{2} \right] \\ &= \frac{n}{2} \nabla_{\Sigma^{-1}} (\log |\Sigma^{-1}|) - \frac{1}{2} \nabla_{\Sigma^{-1}} \left(\sum_{i=1}^n (\mathbf{x}_i - \mu)^t \Sigma^{-1} (\mathbf{x}_i - \mu) \right).\end{aligned}$$

By Lemma 3.1, Lemma 3.3 and by using the fact that as $x^t \Sigma x$ is a scalar and hence the trace of this product to the product itself,

$$\begin{aligned}\nabla_{\Sigma^{-1}} l(\mu, \Sigma^{-1}) &= \frac{n}{2} \Sigma^t - \frac{1}{2} \nabla_{\Sigma^{-1}} \left(\sum_{i=1}^n \text{tr} \left((\mathbf{x}_i - \mu)^t \Sigma^{-1} (\mathbf{x}_i - \mu) \right) \right) \\ &= \frac{n\Sigma}{2} - \frac{1}{2} \nabla_{\Sigma^{-1}} \left(\sum_{i=1}^n \text{tr} \left((\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^t \Sigma^{-1} \right) \right),\end{aligned}$$

and eventually as $(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^t$ is a $p \times p$ matrix and by using Lemma 3.2,

$$\begin{aligned}\nabla_{\Sigma^{-1}} l(\mu, \Sigma^{-1}) &= \frac{n\Sigma}{2} - \frac{1}{2} \nabla_{\Sigma^{-1}} \left(\text{tr} \left(\sum_{i=1}^n (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^t \Sigma^{-1} \right) \right) \\ &= \frac{n\Sigma}{2} - \frac{1}{2} \sum_{i=1}^n (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^t.\end{aligned}$$

By setting the equation to 0,

$$\begin{aligned}\nabla_{\Sigma^{-1}} l(\mu, \Sigma^{-1}) = 0 &\Leftrightarrow \frac{n\Sigma}{2} - \frac{1}{2} \sum_{i=1}^n (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^t = 0 \\ &\Leftrightarrow n\Sigma - \sum_{i=1}^n (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^t = 0 \\ &\Leftrightarrow \Sigma = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^t.\end{aligned}$$

This last term is actually the formula of the sample covariance matrix S . Koller and Friedman (2009, pp 947-948) showed an elegant manner to demonstrate that the log-likelihood for any undirected graph model is concave, thus demonstrating that $\hat{\mu} = \bar{X}$ and $\hat{\Sigma} = S$ are the MLEs for the multivariate normal distribution.

One could estimate the precision matrix as S^{-1} and get information from this matrix. In a graphical model context, this will not give us a lot of information as it is very unlikely to get a sparse matrix S^{-1} (matrix containing many zeros). The common way to construct a graphical model is to compare models for which an edge has been deleted/added using a specific metric (Deviance, AIC, BIC...) and to choose the model via a hypothesis testing procedure or via the values of the metric (e.g. the one that minimizes the AIC or BIC). This process is explained in Figure 3.3.

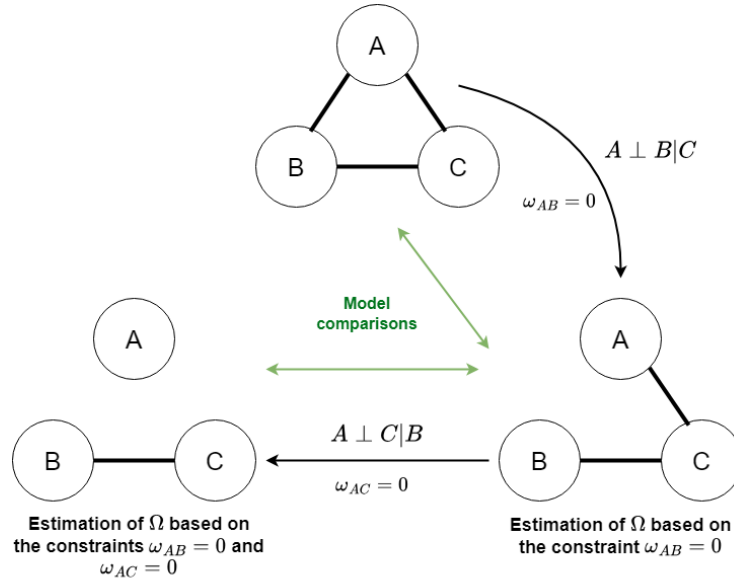


Figure 3.3: Common process to construct a graphical model

When using constraints on the construction of the precision matrix (e.g. with $\omega_{ij} = 0$ for some i, j and $i \neq j$), one has to use iterating algorithms to estimate this matrix. We will illustrate the IPS (Iterative Proportional Scaling) algorithm presented by Speed and Kiiveri (1986) in this thesis, but other algorithms are possible, for example the ADMM (see Yuan, 2012).

The basic idea of the algorithm comes from Dempster (1972) who demonstrates that a positive-definite matrix Σ which respects :

$$\begin{aligned} \Sigma &= S \text{ for the vertices } i \text{ and } j \text{ such that } [i, j] \in \mathcal{E} \\ (\Sigma^{-1})_{ij} &= \omega_{ij} = 0 \text{ for the vertices } i \text{ and } j \text{ such that } [i, j] \notin \mathcal{E}, \end{aligned} \quad (3.3.1)$$

exists and is unique. Furthermore, this matrix has maximum entropy (see Dempster (1972) for a definition) and is the maximum likelihood estimator of Σ under this model.

Let the set $\{i, j\}$ denote the vertices for which $[i, j] \in \mathcal{E}$ and $\{k, l\}$ denote the vertices for which $[k, l] \notin \mathcal{E}$. Two choices are possible for the algorithm. Fix the values σ_{ij} to s_{ij} and update the matrix Ω such that the elements ω_{kl} are driven to 0. The other possibility is to fix the elements ω_{kl} to 0 and update the covariance matrix Σ such that the elements σ_{ij} are driven to the corresponding values of S .

As the cliques of a graph represent the maximum complete subgraph that one can obtain from a graph (meaning that the given covariance-matrix of the clique will stay the same during the algorithm), one will update the matrices iteratively for each clique.

Lemma 3.5. *(see Speed & Kiiveri, 1986) Suppose Q , R and H are positive-definite matrices and denote $a \subseteq \mathcal{V}$ a subset of the components of X and the subscript a to denote the subset of the covariance matrix for X_a . Then the matrix*

$$Q^{-1} = R^{-1} + \begin{bmatrix} H_a^{-1} - R_a^{-1} & 0 \\ 0 & 0 \end{bmatrix} \quad (3.3.2)$$

is positive definite with the properties that :

- (a) $Q_{ij} = H_{ij}$ if $i, j \in a$
- (b) $(Q_{ij})^{-1} = (R_{ij})^{-1}$ if i or $j \notin a$.

These properties are corresponding with equation (3.3.1) and will be used to find the MLE of the model under constraints.

To derivate the formula of matrix Q , we will use the marginal scaling procedure proposed by Kullback (1968). We want to find a density $q(x)$ from the densities $r(x)$ and $h(x)$, with covariance matrix R and H respectively, by scaling the density $r(x)$ so that the marginals $r_a(x_a)$ and $h_a(x_a)$ agree. This can be formulated as

$$q(x) = \frac{r(x)h_a(x_a)}{r_a(x_a)}.$$

Using the multivariate normal density function with the variables X centered such that $\mu = 0$, we have

$$\begin{aligned}
q(x) &= (2\pi)^{-n/2} |R|^{-1/2} \exp(-x^t R^{-1} x / 2) \\
&\quad \times (2\pi)^{-n/2} |H_a|^{-1/2} \exp(-x_a^t H_a^{-1} x_a / 2) \\
&\quad \times (2\pi)^{n/2} |R_a|^{1/2} \exp(x_a^t R_a^{-1} x_a / 2) \\
&= (2\pi)^{-n/2} \left(\frac{|R| |H_a|}{|R_a|} \right)^{-1/2} \exp\left(-\frac{1}{2} (x^t R^{-1} x + x_a^t H_a^{-1} x_a - x_a^t R_a^{-1} x_a)\right) \\
&= (2\pi)^{-n/2} \left(\frac{|R| |H_a|}{|R_a|} \right)^{-1/2} \exp\left(-\frac{1}{2} (x^t R^{-1} x + x_a^t (H_a^{-1} - R_a^{-1}) x_a)\right).
\end{aligned}$$

As the vector x can be decomposed in two parts as $x^t = [x_a^t, x_{\bar{a}}^t]$ with $\bar{a}$ being the complementary set of a ,

$$\begin{aligned}
q(x) &= (2\pi)^{-n/2} \left(\frac{|R| |H_a|}{|R_a|} \right)^{-1/2} \exp\left(-\frac{1}{2} \left(x^t R^{-1} x + [x_a^t, x_{\bar{a}}^t]^t \begin{bmatrix} H_a^{-1} - R_a^{-1} & 0 \\ 0 & 0 \end{bmatrix} [x_a^t, x_{\bar{a}}^t] \right)\right) \\
&= (2\pi)^{-n/2} \left(\frac{|R| |H_a|}{|R_a|} \right)^{-1/2} \exp\left(-\frac{1}{2} x^t \left(R^{-1} + \begin{bmatrix} H_a^{-1} - R_a^{-1} & 0 \\ 0 & 0 \end{bmatrix} \right) x\right).
\end{aligned}$$

Denote Q^{-1} by $R^{-1} + \begin{bmatrix} H_a^{-1} - R_a^{-1} & 0 \\ 0 & 0 \end{bmatrix}$ and we get

$$q(x) = (2\pi)^{-n/2} |Q|^{-1/2} \exp\left(-x^t Q^{-1} x / 2\right),$$

where one can show that $|Q| = \frac{|R| |H_a|}{|R_a|}$.

Assuming that this density function is proper, it results that matrix Q is positive-definite. With these properties ensured for this matrix, two possible algorithms are proposed. Note that as we will update the matrix Q for each clique of the graph, these algorithms are cyclic.

First cyclic algorithm : As a set A has its complementary set denoted by $\bar{A}$, a graph $\mathcal{G}$ has its complementary graph $\bar{\mathcal{G}}$ for which the vertices are the same but the edge set $\bar{\mathcal{E}}$ has the property that $[A, B] \in \bar{\mathcal{E}}$ if and only if $[A, B] \notin \mathcal{E}$.

List next the cliques of the complementary graph $\bar{\mathcal{G}}$ as $\bar{c}_1, \bar{c}_2, \dots, \bar{c}_m$. While the estimate for Σ does not respect the equivalences (3.3.1), we iterate for each clique of the graph as follows. Let $s \in \mathbb{N}$, $1 \leq t \leq m$ and $\Sigma_0 = S$, we create a sequence by updating each time the covariance matrix as

$$\Sigma_{sm+t} = Z_t(\Sigma_{sm+t-1})$$

with $Z_t(\Sigma) = Q^{-1}$ where Q is computed as in equation (3.3.2) using $R = \Sigma_{sm+t-1}^{-1}$, $H_a = \text{diag}(((\Sigma_{sm+t-1}^{-1})_a)^{-1})^{-1}$ and $a = \bar{c}_t$. In this algorithm the matrix Σ is fixed for the elements belonging to a clique while the elements of Σ^{-1} are updated at each step.

Second cyclic algorithm : List the cliques of the graph $\mathcal{G}$ as $c_1, c_2, \dots, c_m$. While the estimate for Σ does not respect the equivalences (3.3.1), we iterate for each clique of the graph as follows. Let $\Sigma_0 = I$, the identity matrix. We create a sequence by updating each time the covariance matrix as

$$\Sigma_{sm+t} = Y_t(\Sigma_{ds+t-1})$$

with $Y_t(\Sigma) = Q$ where Q is computed as in equation (3.3.2) using $R = \Sigma_{sm+t-1}$, $H = S$ and $a = c_t$. In this second algorithm it is the matrix Σ^{-1} that is fixed for the elements belonging to a clique of the **complementary** graph $\bar{\mathcal{G}}$ while the elements of Σ are updated at each step.

Both of these algorithms converge to the unique Σ respecting (3.3.1) as shown by Speed and Kiiveri (1986) using the I -divergence of Kullback (1997). In practice the second algorithm is more often used. After having estimated the precision matrix under certain constraints, we can then compare two models using the Deviance criterion and hypothesis testing as shown by Edwards (2000).

Returning back to the log-likelihood of the model we now express it differently using the fact that

$$\begin{aligned} \sum_{i=1}^n (\mathbf{x}_i - \mu)^t \Sigma^{-1} (\mathbf{x}_i - \mu) &= \sum_{i=1}^n ((\mathbf{x}_i - \bar{\mathbf{x}}) + (\bar{\mathbf{x}} - \mu))^t \Sigma^{-1} ((\mathbf{x}_i - \bar{\mathbf{x}}) + (\bar{\mathbf{x}} - \mu)) \\ &= \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})^t \Sigma^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}) + \sum_{i=1}^n (\bar{\mathbf{x}} - \mu)^t \Sigma^{-1} (\bar{\mathbf{x}} - \mu) \\ &\quad + \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})^t \Sigma^{-1} (\bar{\mathbf{x}} - \mu) + \sum_{i=1}^n (\bar{\mathbf{x}} - \mu)^t \Sigma^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}), \end{aligned}$$

and as $\sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}}) = n\bar{\mathbf{x}} - n\bar{\mathbf{x}} = 0$,

$$\sum_{i=1}^n (\mathbf{x}_i - \mu)^t \Sigma^{-1} (\mathbf{x}_i - \mu) = \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})^t \Sigma^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}) + n(\bar{\mathbf{x}} - \mu)^t \Sigma^{-1} (\bar{\mathbf{x}} - \mu).$$

Let us look at the first term which, using Lemma 3.1 can be expressed as

$$\begin{aligned} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})^t \Sigma^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}) &= \sum_{i=1}^n \text{tr} \left((\mathbf{x}_i - \bar{\mathbf{x}})^t \Sigma^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}) \right) \\ &= \text{tr} \left(\sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}}) (\mathbf{x}_i - \bar{\mathbf{x}})^t \Sigma^{-1} \right) \\ &= \text{tr} \left(n S \Sigma^{-1} \right) = n \text{tr} (S \Sigma^{-1}). \end{aligned}$$

Thus the log-likelihood of the model is now

$$l(\mu, \Sigma) = -\frac{np}{2} \log 2\pi - \frac{n}{2} \log |\Sigma| - \frac{n}{2} \text{tr} \left(S \Sigma^{-1} \right) - \frac{n}{2} (\bar{\mathbf{x}} - \mu)^t \Sigma^{-1} (\bar{\mathbf{x}} - \mu).$$

The idea of the Deviance is to compare a model M with a "saturated" model M_s (using all the data we have, i.e. without any restrictions) in terms of likelihood. It can be computed as

$$D^2 = -2 \log \left(\frac{L(\hat{\mu}, \hat{\Sigma})}{L_s(\hat{\mu}_s, \hat{\Sigma}_s)} \right) = 2(l_s(\hat{\mu}_s, \hat{\Sigma}_s) - l(\hat{\mu}, \hat{\Sigma})),$$

where $\hat{\mu}/\hat{\mu}_s$ and $\hat{\Sigma}/\hat{\Sigma}_s$ are the MLEs under a model and the saturated model respectively. If the model is well-performing, then the log-likelihoods are close and the Deviance is close to 0.

For the saturated model, the log-likelihood evaluated at the MLE ($\hat{\mu}_s = \bar{X}$ and $\hat{\Sigma}_s = S$) is

$$\begin{aligned} l_s(\hat{\mu}_s, \hat{\Sigma}_s) &= -\frac{np}{2} \log 2\pi - \frac{n}{2} \log |S| - \frac{n}{2} \text{tr} \left(S S^{-1} \right) - \frac{n}{2} (\bar{\mathbf{x}} - \bar{\mathbf{x}})^t \hat{\Sigma}_s^{-1} (\bar{\mathbf{x}} - \bar{\mathbf{x}}) \\ &= -\frac{np}{2} \log 2\pi - \frac{n}{2} \log |S| - \frac{np}{2} \end{aligned}$$

For the model, note that the elements in the MLE $\hat{\Sigma}$ are similar to the corresponding ones in S except for the elements where $(\hat{\Sigma}^{-1})_{ij} = 0$, thus the product of the two matrices will stay similar in the diagonal (as the elements differing from S are multiplied by 0) and the trace will be similar, $\text{tr}(S\hat{\Sigma}) = \text{tr}(\hat{\Sigma}\hat{\Sigma}) = p$. The log-likelihood of the model is

$$l(\hat{\mu}, \hat{\Sigma}) = -\frac{np}{2} \log 2\pi - \frac{n}{2} \log |\hat{\Sigma}| - \frac{np}{2},$$

and the deviance becomes

$$\begin{aligned} D^2 &= 2 \left[-\frac{np}{2} \log 2\pi - \frac{n}{2} \log |S| - \frac{np}{2} + \frac{np}{2} \log 2\pi + \frac{n}{2} \log |\hat{\Sigma}| + \frac{np}{2} \right] \\ &= n \log(|\hat{\Sigma}|/|S|). \end{aligned}$$

Thus if one want to compare two models with the Deviance criterion, where $M_0 \subseteq M_1$ (e.g. compare the model AB, BC with ABC), the test statistic becomes

$$\begin{aligned} T &= D_{M_0}^2 - D_{M_1}^2 = 2 \left(\left(l_s(\hat{\mu}_s, \hat{\Sigma}_s) - l_0(\hat{\mu}_0, \hat{\Sigma}_0) \right) - \left(l_s(\hat{\mu}_s, \hat{\Sigma}_s) - l_1(\hat{\mu}_1, \hat{\Sigma}_1) \right) \right) \\ &= 2 \left(l_1(\hat{\mu}_1, \hat{\Sigma}_1) - l_0(\hat{\mu}_0, \hat{\Sigma}_0) \right) \\ &= n \log(|\hat{\Sigma}_0|/|\hat{\Sigma}_1|), \end{aligned}$$

where $\hat{\Sigma}_0$ and $\hat{\Sigma}_1$ are the MLEs of Σ under M_0 and M_1 respectively. This test statistic T , under M_0 (so under the hypothesis that M_0 is the true model), follows a χ^2 distribution where the degrees of freedom are given by the difference between the number of free parameters in M_0 and M_1 (i.e. the number of ω_{ij} put to 0 from M_1 to M_0). A possible hypothesis testing is then

$$H_0 : M_0 \text{ is the true model versus } H_1 : M_0 \text{ is not the true model,}$$

and one can use the test statistic T to reject H_0 if the pvalue of the test is lower than a given significance level α (say 0.05).

In the next part, we will focus on the high-dimensional case where $p \gg n$ and where the usual statistical tools cannot be used anymore.

3.4 Penalization

When dealing with high-dimensional data such as genomics data where $p \gg n$, the sample covariance matrix is singular and cannot be inverted, preventing us from estimating the precision matrix of the GGM. In this situation, the number of parameters $(p(p-1)/2)$ exceeds the number of observations and without selecting variables or using penalization methods, this will result in an overfit of the data. A possibility could be to use a stepwise forward variable selection, starting from an empty model and choosing the variable to add given their pvalue of a hypothesis test (or via AIC, BIC...). With this approach, one needs to select a model for each step and estimate the corresponding precision matrix but we do not take into account the multiple comparisons done using this method and we lose in terms of computational complexity as the number of variables increases.

To handle this problem, a possible solution is the use of simultaneous conservative confidence intervals at significance level $1 - \alpha$ to perform a model selection in only one step (see Drton & Perlman, 2004). Another possibility is to perform a neighborhood selection for each variable (using the graph) and to combine the obtained results as developed by Meinshausen and Bühlmann (2006). For these solutions, the model selection part and the parameters estimation part are separated.

In this thesis, we will study two lasso-type penalization methods on this graphical model framework, one developed by Yuan and Lin (2007) and the second by Friedman et al. (2008) called the Glasso method. We start with the first method.

3.4.1 Lasso-type estimator

The main idea is to apply an ℓ_1 penalty on the off-diagonal elements of the precision matrix to obtain a sparse matrix. The optimization problem is to maximize the log-likelihood (or to minimize the negative log-likelihood) with a constraint on the off-diagonal elements. Note that, as before, we perform the transformation $\log |\Sigma| = -\log |\Sigma^{-1}|$. As such the goal is to minimize

$$\frac{np}{2} \log 2\pi - \frac{n}{2} \log |\Omega| + \frac{1}{2} \sum_{i=1}^n (x_i - \mu)^t \Omega (x_i - \mu),$$

subject to

$$\sum_{i \neq j} |\omega_{ij}| \leq t,$$

over the set of positive definite matrices where t is a non-negative constant. The positive-definiteness of the matrix is crucial for invertability reasons. If $t = \infty$ then no constraint is imposed and the solution of the problem is the classical MLE of Ω , S^{-1} . If $t = 0$, all the off-diagonal elements are forced to 0 and we obtain a diagonal matrix Ω .

As the first term of the log-likelihood does not depend on Ω we can discard it and if the observations are centered, then $\hat{\mu} = 0$ and we get

$$\hat{\Omega} = \arg \min_{\Omega \succ 0} \left\{ -\log |\Omega| + \frac{1}{n} \sum_{i=1}^n x_i^t \Omega x_i \right\},$$

subject to $\sum_{i \neq j} |\omega_{ij}| \leq t$, or

$$\hat{\Omega} = \arg \min_{\Omega \succ 0} \{ -\log |\Omega| + \text{tr}(\Omega S) \},$$

subject to $\sum_{i \neq j} |\omega_{ij}| \leq t$, over the set of positive definite matrices.

One can show that the objective function and the constraint are convex. If a solution of the problem is found, this solution will be a global minimizer. We can also rewrite the problem using the Lagrangian

$$\mathcal{L}(\Omega, \lambda) = -\log |\Omega| + \text{tr}(\Omega S) + \lambda \sum_{i \neq j} |\omega_{ij}|,$$

where $\lambda > 0$ is a tuning parameter of the penalization.

To optimize this problem the positive-definiteness of Ω needs to be ensured. We will use the Maxdet problem formulation and the interior point algorithm (see Vandenberghe et al., 1998) to solve the problem. The interior point algorithm is a type of algorithm that iteratively goes through the feasible set (the region in which the constraints are respected, in our case where $\sum_{i \neq j} |\omega_{ij}| \leq t$) and finds an optimal solution in the feasible set or on its

boundary (where the sign $\leq$ becomes an equality). The Maxdet problem is of the form

$$\begin{aligned} & \min_{x \in \mathbb{R}^m} c^t x - \log |G(x)| \\ & \text{subject to } G(x) \succ 0 \\ & \quad F(x) \succeq 0, \end{aligned}$$

where $c \in \mathbb{R}^m$, $G : \mathbb{R}^m \rightarrow \mathbb{R}^{l \times l}$ and $F : \mathbb{R}^m \rightarrow \mathbb{R}^{l \times l}$ are affine functions such that

$$\begin{aligned} G(x) &= G_0 + x_1 G_1 + \cdots + x_m G_m \\ F(x) &= F_0 + x_1 F_1 + \cdots + x_m F_m, \end{aligned}$$

and each G_i and F_i are symmetric matrices. In our case, $F(x) = I$ as it is not used and the Maxdet problem for the lasso-type estimator is

$$\begin{aligned} & \min_{\Omega} \sum_{i=1}^p s_{ii} \omega_{ii} + 2 \sum_{i < j} s_{ij} \omega_{ij} - \log \left| \sum_{i=1}^p \omega_{ii} I^{(i)} + \sum_{i < j} \omega_{ij} I^{(ij)} \right| \\ & \text{subject to } \sum_{i=1}^p \omega_{ii} I^{(i)} + \sum_{i < j} \omega_{ij} I^{(ij)} \succ 0, \\ & \quad t - 2 \sum_{i < j} \omega_{ij} r_{ij} \geq 0, \\ & \quad \omega_{ij} r_{ij} \geq 0, \end{aligned} \tag{3.4.1}$$

where $I^{(i)}$ is equal to 1 for the $(i, i)^{th}$ entry and 0 elsewhere, $I^{(ij)}$ is 1 for the $(i, j)^{th}$ and $(j, i)^{th}$ entries and 0 elsewhere, r_{ij} is the sign of ω_{ij} (-1 if $\omega_{ij} < 0$ and 1 if $\omega_{ij} > 0$). This reformulation can be obtained using the fact that $|AB| = \sum_{i=1}^p a_{ii} b_{ii} + \sum_{i < j} a_{ij} b_{ij} + \sum_{i > j} a_{ij} b_{ij}$ and using the symmetry of S and Ω .

Eventually, the interior point algorithm for this problem can be decomposed in 3 steps. First let $\hat{\Omega}_0$ be any estimator but if $p < n$, S^{-1} can be used, and $r_{ij} = \text{sign} \{\hat{\omega}_{0,ij}\} \forall i \neq j$. Next solve problem (3.4.1) at iteration k with $\hat{\Omega}_{k-1}$ and obtain an update $\hat{\Omega}_k$ over the set of positive-definite matrices. Eventually if $\hat{\Omega}_k = \hat{\Omega}_{k-1}$ we stop and return $\hat{\Omega} = \hat{\Omega}_k$, else $\hat{\Omega}_{k-1} = \hat{\Omega}_k$ and $r_{ij} = -r_{ij}$ for all pairs (i, j) such that $\hat{\omega}_{ij} = 0$ and return to the second step. We then iterate this algorithm until convergence.

The estimator obtained from this algorithm can then be investigated to construct the graph structure of the model.

3.4.2 Glasso estimator

The second algorithm called graphical lasso or Glasso is based on the same optimization problem but uses a block coordinate descent approach to estimate the covariance matrix Σ instead of the precision matrix Ω . The block coordinate descent aims to find a solution to the problem by iterating over each column and corresponding row of the matrix until convergence. The idea is to construct the matrix into blocks where the upper-left block includes all rows and columns except the j^{th} row and column.

We start by setting the optimization problem.

$$\max \log |\Omega| - \text{tr}(S\Omega) - \lambda_{GL} \|\Omega\|_1, \quad (3.4.2)$$

where λ_{GL} is a tuning parameter and $\|\cdot\|_1$ denotes the element-wise ℓ_1 norm defined in Chapter 2. The main difference with the previous method is the penalization of all the elements in the matrix rather than just the off-diagonal elements.

To develop the block coordinate descent method, let us define the estimated covariance matrix W . This matrix will be constructed as a block matrix

$$W = \begin{pmatrix} W_{-j,-j} & W_j \\ W_j^t & w_{jj} \end{pmatrix},$$

where $W_{-j,-j}$ is the block matrix produced by removing row and column j and W_j denotes column j without the w_{jj} diagonal element. Note that the target column is always in the last row/column of the matrix W .

We can also construct the sample covariance matrix S in the same way

$$S = \begin{pmatrix} S_{-j,-j} & S_j \\ S_j^t & s_{jj} \end{pmatrix}.$$

The Graphical lasso algorithm is then performed in 3 steps. First, compute the matrix $W = S + \lambda I_p$. The diagonal elements of W will then remain fixed during the process. For each column $j = 1, \dots, p$, we iteratively solve

$$\hat{\beta} = \arg \min_{\beta} \left\{ \frac{1}{2} |W_{-j,-j}^{1/2} \beta - W_{-j,-j}^{-1/2} S_j|_2 + \lambda_{GL} |\beta|_1 \right\}, \quad (3.4.3)$$

to obtain a $p-1$ vector $\hat{\beta}$ for each column/row and compute $w_j = W_{-j,-j} \hat{\beta}$. The equivalence between the optimization equation 3.4.2 and the optimization equation (3.4.3), to obtain a solution for the graphical lasso, is shown in Banerjee et al. (2008) using optimization techniques.

We eventually reiterate step 2 until convergence, defined by Friedman et al. (2008) as when the average absolute change of W between two iterations of step 2 is lower than $0.001 * \frac{1}{p(p-1)} ||S^{-\text{diag}}||_1$ where $S^{-\text{diag}}$ are the off-diagonal elements of S .

From the matrix W , one can obtain the precision matrix by using the fact that

$$W\Omega = I_p \Leftrightarrow \begin{pmatrix} W_{-j,-j} & W_j \\ W_j^t & w_{jj} \end{pmatrix} \begin{pmatrix} \Omega_{-j,-j} & \Omega_j \\ \Omega_j^t & \omega_{jj} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0^t & 1 \end{pmatrix},$$

leading to the equalities

$$\begin{aligned} W_{-j,-j}\Omega_j + W_j\omega_{jj} &= 0, \\ W_j^t\Omega_j + w_{jj}\omega_{jj} &= 1, \end{aligned}$$

from which one can show that

$$\begin{aligned} \Omega_j &= -W_{-j,-j}^{-1}W_j\omega_{jj}, \\ \omega_{jj} &= 1/(w_{jj} - W_j^tW_{-j,-j}^{-1}W_j), \end{aligned}$$

and eventually by using the fact that $\hat{\beta} = W_{-j,-j}^{-1}W_j$,

$$\begin{aligned} \hat{\Omega}_{GL,j} &= -\hat{\beta}\hat{\omega}_{GL,jj}, \\ \hat{\omega}_{GL,jj} &= 1/(w_{jj} - W_j^t\hat{\beta}). \end{aligned}$$

As these estimates are only needed at the end of the algorithm, we can actually save a matrix containing all the vectors $\hat{\beta}$ for each column and use it to estimate all the elements of the estimated precision matrix.

With this second penalization method explained, we can introduce an edge detection method developed for this estimator, which will be useful in the simulation part to compare with the Transfer-Learning estimator.

3.4.3 Edge detection method for the Glasso estimator

This method was developed by Janková and van de Geer (2015). By definition, the Glasso estimator is biased. The idea is then to debias (or de-sparsify) the estimator and find a statistic to construct hypothesis tests.

An important assumption used by Janková and van de Geer (2015) is the so-called "irrepresentability condition" that we will explain after setting some notations.

Denote by $\Gamma := \Omega^{-1} \otimes \Omega^{-1}$ the Hessian of the negative log-likelihood a $p^2 \times p^2$ matrix, where $\otimes$ defines the Kronecker product between two matrices (see Chapter 2).

Indeed, recall we have defined the negative likelihood as

$$-l(\Omega) = \text{tr}(S\Omega) - \log |\Omega|,$$

and showed in Section 3.3 that its first derivative is

$$\nabla_{\Omega}(-l(\Omega)) = S - \Omega^{-1},$$

meaning that taking derivatives of the negative log-likelihood with respect to ω_{ij} gives us

$$\frac{\partial(-l(\Omega))}{\partial\omega_{ij}} = s_{ij} - (\Omega^{-1})_{ij}.$$

The Hessian for the pair of indexes $((i, j), (k, l))$ can then be obtained by taking derivatives with respect to ω_{kl} ,

$$\frac{\partial^2(-l(\Omega))}{\partial\omega_{ij}\partial\omega_{kl}} = e_i^t \Omega^{-1} e_k e_l^t \Omega^{-1} e_j = \sigma_{ik} \sigma_{lj},$$

which can be arranged in matrix form as

$$\Gamma = \Omega^{-1} \otimes \Omega^{-1}.$$

The second partial derivative of the negative log-likelihood for the pair of indexes $((i, j), (k, l))$ is then defined as $\Gamma_{(i,j)(j,l)}$.

Moreover, denote $V = \{(i, j) \in \mathcal{V} : \omega_{ij} \neq 0\}$ as the set of non-zero elements in the precision matrix Ω (including the diagonal), $\bar{V}$ the complement set of V and $|V| = v + p$ where v is the number of non-zero off-diagonal elements. We can then consider the matrix

$$\Gamma_{VV} := \left[\Omega^{-1} \otimes \Omega^{-1} \right]_{VV} \in \mathbb{R}^{(v+p) \times (v+p)},$$

where Γ_{VV} denotes the $(v + p) \times (v + p)$ matrix with rows and columns indexed by the elements belonging to V .

Eventually Janková and van de Geer (2015) have stated this irrepresentability condition :

Assumption 3.6. *There exists some $\alpha \in (0, 1]$ such that*

$$\max_{e \in \bar{V}} \|\Gamma_{eV} (\Gamma_{VV})^{-1}\|_{L_1} \leq (1 - \alpha),$$

where $\|\cdot\|_{L_1}$ is the ℓ_1 matrix norm defined in Chapter 2. This assumption limits the influence that the non-edge terms (i.e. belonging to $\bar{V}$) can have on the edge-based terms. Intuitively, there shouldn't be an edge that is not included in the graph having a high correlation with an edge belonging to this graph. We assume that for some small α this correlation between non-edge terms and edge-based terms will be small. See Ravikumar et al. (2011, section 3) for some illustrations of this assumption.

Starting from this framework, we define the Karush-Kuhn-Tucker (KKT) condition of the optimization problem (see appendix A.1 for an introduction of these conditions):

$$S - \hat{\Omega}^{-1} + \lambda \hat{R} = 0, \quad (3.4.4)$$

where the matrix $\hat{R}$ belongs to the sub-differential of the matrix norm $\|\Omega\|_1$ evaluated at the glasso estimator $\hat{\Omega}$ and is defined as

$$\hat{r}_{ij} = \begin{cases} 0 & \text{if } i = j \\ \text{sign}(\omega_{ij}) & \text{if } i \neq j \text{ and } \omega_{ij} \neq 0 \\ \in [-1, 1] & \text{if } i \neq j \text{ and } \omega_{ij} = 0. \end{cases}$$

Janková and van de Geer (2015) then proposed to "invert" the KKT condition by multiplying it by the inverse Hessian of the negative log-likelihood Γ

$$(\Gamma)^{-1} = (\Omega^{-1} \otimes \Omega^{-1})^{-1} = \Omega \otimes \Omega,$$

evaluated at the glasso estimator $\hat{\Omega}$. Using the mixed Kronecker matrix-vector product property of the Kronecker product (see Liu et al., 2023), we can perform this multiplication by multiplying $\hat{\Omega}$ from both sides.

$$\begin{aligned} \hat{\Omega} S \hat{\Omega} - \hat{\Omega} \hat{\Omega}^{-1} \hat{\Omega} + \hat{\Omega} \lambda \hat{R} \hat{\Omega} &= 0 \\ \Leftrightarrow \hat{\Omega} S \hat{\Omega} - \hat{\Omega} + \hat{\Omega} \lambda \hat{R} \hat{\Omega} &= 0. \end{aligned}$$

By defining $E = S - \Sigma$, we can express the previous equation as

$$\begin{aligned}
& \hat{\Omega}S\hat{\Omega} - \hat{\Omega}\Sigma\Omega + \hat{\Omega}\lambda\hat{R}\hat{\Omega} = 0 \\
& \Leftrightarrow \hat{\Omega}S\hat{\Omega} + \hat{\Omega}(S - \Sigma)\Omega - \hat{\Omega}S\Omega + \hat{\Omega}\lambda\hat{R}\hat{\Omega} = 0 \\
& \Leftrightarrow \hat{\Omega}S\hat{\Omega} + \hat{\Omega}E\Omega - \hat{\Omega}S\Omega + \hat{\Omega}\lambda\hat{R}\hat{\Omega} = 0 \\
& \Leftrightarrow (\hat{\Omega}S - I)\hat{\Omega} + \hat{\Omega} + \hat{\Omega}E\Omega - (\hat{\Omega}S - I)\Omega - \Omega + \hat{\Omega}\lambda\hat{R}\hat{\Omega} = 0 \\
& \Leftrightarrow \hat{\Omega} + (\hat{\Omega} - \Omega)E\Omega + \Omega E\Omega + (\hat{\Omega}S - I)(\hat{\Omega} - \Omega) - \Omega + \hat{\Omega}\lambda\hat{R}\hat{\Omega} = 0 \\
& \Rightarrow \hat{\Omega} + \hat{\Omega}\lambda\hat{R}\hat{\Omega} - \Omega = -\Omega E\Omega + \text{rem},
\end{aligned}$$

where $\text{rem} = -(\hat{\Omega} - \Omega)E\Omega - (\hat{\Omega}S - I)(\hat{\Omega} - \Omega)$. The goal of this transformation was to show that the left side is equal to $\Omega E\Omega$ which is (elementwise) asymptotically normal as shown by Janková and van de Geer (2015) in Theorem 1.

From this expression we can define the de-sparsified Glasso estimator as

$$\hat{T} := \hat{\Omega} + \hat{\Omega}\lambda\hat{R}\hat{\Omega} - \Omega.$$

By using the KKT condition (3.4.4) we have

$$\lambda\hat{R} = \hat{\Omega}^{-1} - S,$$

and the de-sparsified estimator becomes

$$\hat{T} = 2\hat{\Omega} + \hat{\Omega}S\hat{\Omega},$$

with, under regularity conditions,

$$\frac{\sqrt{n}(\hat{T}_{ij} - \omega_{ij})}{\sqrt{\sigma_{ij}^2}} \xrightarrow{\mathcal{L}} N(0, 1),$$

where $\xrightarrow{\mathcal{L}}$ denotes convergence in distribution. The asymptotic variance of $\hat{T}_{ij}$ can be estimated by $\hat{\sigma}_{ij}^2 = \hat{\omega}_{ii}\hat{\omega}_{jj} + \hat{\omega}_{ij}^2$. From this asymptotic behavior, confidence intervals with confidence level $(1 - \alpha)100\%$ can be computed as

$$IC_{1-\alpha}(\omega_{ij}) = \left[\hat{T}_{ij} - \Phi^{-1}\left(1 - \frac{\alpha}{2}\right) \times \frac{\hat{\sigma}_{ij}}{\sqrt{n}}, \quad \hat{T}_{ij} + \Phi^{-1}\left(1 - \frac{\alpha}{2}\right) \times \frac{\hat{\sigma}_{ij}}{\sqrt{n}} \right],$$

where $\Phi^{-1}(1 - \frac{\alpha}{2})$ is the quantile function of the normal distribution at probability $1 - \frac{\alpha}{2}$.

We now aim to test the hypothesis

$$H_{0,ij} : \omega_{ij} = 0 \text{ versus } H_{1,ij} : \omega_{ij} \neq 0.$$

One can compute the test statistic $\hat{z}_{ij} = \frac{\sqrt{n}\hat{\Gamma}_{ij}}{\sqrt{\hat{\sigma}_{ij}^2}}$ for $i, j = 1, \dots, p$ and adjust the pvalues obtained by a multiplicity correction as the Benjamini-Hochberg procedure for a significance level α (see Benjamini & Hochberg, 1995). Eventually, we can set to 0 all elements of the estimator for which we do not reject H_0 at level α with multiplicity correction to get an edge detection method for the Glasso estimator.

This section eventually brings us to the end of this chapter on GGMs and we now dive into the basis of the Transfer Learning method.

Chapter 4

Transfer Learning on GGM

We saw in the first chapter how to estimate a precision matrix in a high-dimensional setting. While these methods are particularly useful for biological studies, they can face high uncertainty in the estimation process due to small sample sizes. A possible and promising solution is Transfer Learning, where we share the knowledge from one or multiple source datasets to improve the performance and precision of the target data estimation. For example we could have data from a specific time period and use the corresponding model to reduce the estimation uncertainty in another time period. By using related models based on data with the **same** variables but **different** distributions, we presume that some similarities between them could exist and could help on estimating the target model.

Usually, this method is used for Machine Learning frameworks to reduce the efforts of learning a new model from scratch by adding skills and knowledge learned by other related models. To cite some useful applications of this method, one can perform it with text classification - for which topic a given text talks about - (see Dai et al., 2007), sentiment classification problems - given a review, does it contain good or bad sentiments - (see Liu et al., 2019), medical image analysis (see Morid et al., 2021) or even Wifi localization models (see Pan et al., 2007).

A problem that can occur with Transfer Learning is negative transfer, i.e. the transferred knowledge results in lowering the performance of the estimation. Large dissimilarities between the target model and the other models have been shown to be one of the reasons of negative transfer (see Rosenstein et al., 2005). In a GGM framework, we will use an aggregation step to counteract this effect in order to improve the process.

This chapter is mainly inspired by the article of Li et al. (2022) who developed the Transfer Learning methodology into the estimation of Gaussian Graphical Models. Firstly the estimator used by this article, the CLIME estimator, will be presented. Then the main Transfer Learning process will be discussed with a method to detect edges of the graph. Eventually we will introduce the rates of convergence of the different estimators.

4.1 The CLIME Estimator

Before going into the Transfer Learning methodology, let us talk about the estimator introduced by Cai et al. (2011) and used by Li et al. (2022) in the Transfer Learning framework. To fully understand the following propositions, some notations have been explained in Chapter 2.

The CLIME estimator (which stands for Constrained ℓ_1 -minimization for Inverse Matrix Estimation) is obtained not by optimizing the log-likelihood, but comes from the so-called Moment Equation

$$\Sigma\Omega - I_p = 0.$$

The idea is to put an upper bound on this equation so that the estimator of Ω tends to respect it. Firstly we compute $\{\hat{\Omega}_1\}$ a solution set of the problem

$$\begin{aligned} & \min \|\Omega\|_1 \\ & \text{subject to } \|S\Omega - I_p\|_\infty \leq \lambda_n, \Omega \in \mathbb{R}_{p \times p}, \end{aligned} \quad (4.1.1)$$

where $\lambda_n > 0$ is a tuning parameter and S the empirical covariance matrix. In this optimization problem we do not impose the symmetry of the matrix Ω , so one needs to add this symmetry on the estimator $\hat{\Omega}_1$.

Defining $\hat{\Omega}_1 = (\hat{\omega}_{ij}^1)$, the CLIME estimator $\hat{\Omega}$ of Ω is defined as $\hat{\Omega} = (\hat{\omega}_{ij})$ where

$$\hat{\omega}_{ij} = \hat{\omega}_{ji} = \hat{\omega}_{ij}^1 \mathbb{1}(|\hat{\omega}_{ij}^1| \leq |\hat{\omega}_{ji}^1|) + \hat{\omega}_{ji}^1 \mathbb{1}(|\hat{\omega}_{ij}^1| > |\hat{\omega}_{ji}^1|),$$

where $\mathbb{1}$ is the indicator function, resulting in 1 if the statement is true and 0 otherwise. This transformation actually takes the minimum between $\hat{\omega}_{ij}^1$ and $\hat{\omega}_{ji}^1$ in terms of magnitude. The estimator obtained is positive-definite with high probability as shown by Cai et al. (2011). As Ω is positive-definite and the constraint is convex (in the form $g(x) \leq 0$ with $g(x)$ a convex function), this optimization problem is convex, meaning that a minimizer of problem (4.1.1) is actually a global minimizer.

Another useful property of this optimization problem is its possible decomposition into a p -vector minimization problem. Let e_i be a p -vector containing a one at the i^{th} element and zeros elsewhere. Let $\hat{\beta}_i$ be the solution of the problem

$$\begin{aligned} & \min \|\beta\|_1 \\ & \text{subject to } |S\beta - e_i|_\infty \leq \lambda_n. \end{aligned} \quad (4.1.2)$$

Cai et al. (2011) showed that solving the p -optimization problems (4.1.2) with solutions $\hat{\beta}_i$ or solving the optimization problem (4.1.1) with solution $\hat{\Omega}_1$ is equivalent. Mathematically speaking, let $\{\hat{B}\} := \{(\hat{\beta}_1, \dots, \hat{\beta}_p)\}$ then $\{\hat{\Omega}_1\} = \{\hat{B}\}$. This means that the estimation of the precision matrix can be done column by column.

4.2 Transfer Learning

In a Transfer Learning setting, we observe samples $\mathbf{x}_i$, $i = 1, \dots, n$ from a target normal distribution $N(\mu, \Sigma)$ and samples $\mathbf{x}_i^{(k)}$ from K other auxiliary studies with normal distributions $N(\mu^{(k)}, \Sigma^{(k)})$, $i = 1, \dots, n_k$, $k = 1, \dots, K$ where n_k is the sample size of study k (we therefore have a total of $K + 1$ studies). We denote $\Omega^{(k)} = [\Sigma^{(k)}]^{-1}$ the precision matrix for the k^{th} study, $\Omega \in \mathbb{R}^{p \times p}$. As explained in the introduction of Transfer Learning, the samples share the same variables.

The main idea of the framework is to create a measure of the similarities between the target study and the auxiliary studies and transfer the knowledge from one study to the target one. The entropy (or Kullback-Leibler Divergence) will be the basis to measure this similarity (see Kullback, 1997). The Kullback-Leibler Divergence between the k^{th} study and the target one can be written as

$$D_{KL}(N_{(\Sigma^{(k)})}|N_{(\Sigma)}) = \frac{1}{2}\text{tr}(\Delta^{(k)}) - \frac{1}{2}\log(I_p + \Delta^{(k)}), \quad (4.2.1)$$

where $\Delta^{(k)} := \Omega\Sigma^{(k)} - I_p$ is defined as the divergence matrix for study k .

Let us present a Lemma useful for the proof.

Lemma 4.1. *Let A and B be two $n \times n$ matrices and $|\cdot|$ denotes the determinant of a matrix, then*

$$|AB| = |A||B|.$$

We now demonstrate the proof of the Kullback-Leibler divergence in (4.2.1).

Proof. Let $X^{(k)} \in \mathbb{R}^p$, $X^{(k)} \sim N(0, \Sigma^{(k)})$ with pdf $f(x)$ and $X \in \mathbb{R}^p$, $X \sim N(0, \Sigma)$ with pdf $g(x)$, then

$$\begin{aligned} D_{KL}(N_{\Sigma^{(k)}}|N_{\Sigma}) &= \int \log\left(\frac{f(x)}{g(x)}\right) f(x)dx = E\left[\log\left(\frac{f(X)}{g(X)}\right)\right] \\ &= E\left[\log\left(\frac{(2\pi)^p |\Sigma|^{1/2}}{(2\pi)^p |\Sigma^{(k)}|^{1/2}} \exp\left(-\frac{1}{2}X^t[\Sigma^{(k)}]^{-1}X + \frac{1}{2}X^t\Sigma^{-1}X\right)\right)\right] \\ &= E\left[\frac{1}{2}\log\left(\frac{|\Sigma|}{|\Sigma^{(k)}|}\right) - \frac{1}{2}X^t[\Sigma^{(k)}]^{-1}X + \frac{1}{2}X^t\Sigma^{-1}X\right]. \end{aligned}$$

As $X^t \Sigma^{-1} X$ is a scalar, we use Lemma 3.1 and the fact that the trace of a scalar is the scalar itself to get

$$\begin{aligned} D_{KL}(N_{\Sigma^{(k)}}|N_{\Sigma}) &= \frac{1}{2} E \left[\log \left(\frac{|\Sigma|}{|\Sigma^{(k)}|} \right) - \text{tr}(X^t [\Sigma^{(k)}]^{-1} X) + \text{tr}(X^t \Sigma^{-1} X) \right] \\ &= \frac{1}{2} \left(\log \left(\frac{|\Sigma|}{|\Sigma^{(k)}|} \right) - E[\text{tr}(X X^t [\Sigma^{(k)}]^{-1})] + E[\text{tr}(X X^t \Sigma^{-1})] \right) \\ &= \frac{1}{2} \left(\log \left(\frac{|\Sigma|}{|\Sigma^{(k)}|} \right) - \text{tr}(E[X X^t] [\Sigma^{(k)}]^{-1}) + \text{tr}(E[X X^t] \Sigma^{-1}) \right). \end{aligned}$$

Under the model $N(0, \Sigma^{(k)})$, the expectation of $X X^t$ is the covariance matrix $\Sigma^{(k)}$.

$$\begin{aligned} D_{KL}(N_{\Sigma^{(k)}}|N_{\Sigma}) &= \frac{1}{2} \left(\log \left(\frac{|\Sigma|}{|\Sigma^{(k)}|} \right) - \text{tr}(\Sigma^{(k)} [\Sigma^{(k)}]^{-1}) + \text{tr}(\Sigma^{(k)} \Sigma^{-1}) \right) \\ &= \frac{1}{2} \left(\log \left(\frac{|\Sigma|}{|\Sigma^{(k)}|} \right) - \text{tr}(I_p) + \text{tr}(\Sigma^{(k)} \Omega) \right). \end{aligned}$$

Define

$$\Delta^{(k)} := \Omega \Sigma^{(k)} - I_p. \quad (4.2.2)$$

Using Lemma 4.1, we have $\log \left(\frac{|\Sigma|}{|\Sigma^{(k)}|} \right) = -\log \left(\frac{|\Sigma^{(k)}|}{|\Sigma|} \right) = -\log(|\Sigma^{(k)}| |\Omega|) = -\log |\Sigma^{(k)} \Omega|$. By equation (4.2.2) we get $\Omega \Sigma^{(k)} = \Delta^{(k)} + I_p$ and we have

$$\begin{aligned} D_{KL}(N_{\Sigma^{(k)}}|N_{\Sigma}) &= \frac{1}{2} \left(-\log(|\Sigma^{(k)} \Omega|) + \text{tr}(\Sigma^{(k)} \Omega - I_p) \right) \\ &= \frac{1}{2} \text{tr}(\Delta^{(k)}) - \frac{1}{2} \log |\Delta^{(k)} + I_p|. \end{aligned}$$

□

From the Kullback-Leibler divergence, all we need to estimate is the matrix $\Delta^{(k)}$ measuring the similarity between the target study (Σ) and the k^{th} study ($\Sigma^{(k)}$). The closer $\Delta^{(k)}$ is to $0_{p \times p}$, the larger the similarity is (since $\Omega \Sigma^{(k)}$ will be close to I_p). Note that as Σ and $\Sigma^{(k)}$ are different, the product $\Omega \Sigma^{(k)}$ will not result in a symmetric matrix, therefore neither will $\Delta^{(k)}$. Using this idea, Li et al. (2022) proposed a metric to measure the difference between Ω and $\Omega^{(k)}$:

$$D_q(\Omega^{(k)}, \Omega) = \max_{1 \leq j \leq p} \|\Delta_{j,\cdot}^{(k)}\|_q + \max_{1 \leq j \leq p} \|\Delta_{\cdot,j}^{(k)}\|_q, \quad (4.2.3)$$

where $A_{j,\cdot}$ denotes the j^{th} row of A and $A_{\cdot,j}$ the j^{th} column of A , $q \in [0, 1]$. We compute the maximum ℓ_q norm for the columns and rows of $\Delta^{(k)}$ since it is not symmetric. Note that this metric measures a "relative distance" as $D_q(c\Omega^{(k)}, c\Omega) = D_q(\Omega^{(k)}, \Omega)$ for any constant $c > 0$. It can be proved by the fact that

$$\Delta^{(k)} = (c\Omega)(c\Omega^{(k)})^{-1} - I_p = \Omega(\Omega^{(k)})^{-1} - I_p.$$

Another interesting property of this metric is that as $q \in [0, 1]$, we can get an upper bound for $D_{KL}(N_{\Sigma^{(k)}}|N_{\Sigma})$ when $q = 1$.

Coming back to the transfer learning algorithm, the main idea is to estimate $\Delta^{(k)}$. As with the CLIME estimator, it can be useful to look at the Moment Equation to search for an estimation method.

$$\Sigma\Omega - I_p = 0.$$

Using equation (4.2.2), we can express the Moment Equation in a Transfer Learning context as

$$I_p = \Sigma^{(k)}\Omega^{(k)} = \Sigma^{(k)}\Omega - (\Delta^{(k)})^t. \quad (4.2.4)$$

From equation (4.2.4), $\Delta^{(k)}$ seems to be the bias when the estimation of Ω is based on the k^{th} study. As the sample size for each study is different, it is useful to compute a weighted covariance and divergence matrix.

$$\Sigma^K = \sum_{k=1}^K \alpha_k \Sigma^{(k)} \text{ and } \Delta^K = \sum_{k=1}^K \alpha_k \Delta^{(k)},$$

where $\alpha_k = n_k/N$, $N = \sum_{k=1}^K n_k$. From the Moment Equation (4.2.4) we get

$$\Sigma^K\Omega - (\Delta^K)^t - I_p = 0, \quad (4.2.5)$$

which will be useful for the trans-CLIME algorithm. As said in the introduction, an inconvenience of the Transfer Learning is the negative transfer, which can occur in our case if $\Omega^{(k)}$ is far from Ω (or if $D_q(\Omega^{(k)}, \Omega)$ is very large). Li et al. (2022) proposed to use an aggregation method as a solution.

Aggregation methods focus on finding an estimator by combining several possible estimators in a suitable way. For example for a regression model $Y_i = f(X_i) + \epsilon_i$, ϵ_i being the error term, 4 suitable estimators $\hat{f}_1, \hat{f}_2, \hat{f}_3$ and $\hat{f}_4$ are proposed. We would like to find a function $\tilde{f}$, called an *aggregate*, by combining the previous 4 possible estimators. This method is mainly used in regression models (see Bunea et al., 2007; Lecué & Rigollet, 2014).

Three common ways of aggregations are model selection aggregation, convex aggregation and linear aggregation. In a model selection aggregation setting, we aim to find the optimal estimator from the possibilities in terms of a certain metric, in a convex aggregation we want to find the optimal **convex** combination of the possible estimators and in the linear aggregation the objective is to find the optimal **linear** combination of the possible estimators. In our case, Li et al. (2022) proposed to use a model selection aggregation.

To perform the aggregation method, we will split the data of the target study into a training set and a test set. Let $\mathcal{C}$ be a random subset of the sample such that $|\mathcal{C}| = cn$ with $c \in (0, 1)$ and $|\cdot|$ denotes the cardinality of the set, the number of elements contained in it (not to be confused with the determinant of a matrix or the norm of a vector). We denote $\bar{\mathcal{C}}$ as the complementary set of $\mathcal{C}$ with $\tilde{n} = |\bar{\mathcal{C}}|$. We now define several statistics :

$$\begin{aligned}\bar{X} &= \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} X_i \text{ and } \tilde{X} = \frac{1}{|\bar{\mathcal{C}}|} \sum_{i \in \bar{\mathcal{C}}} X_i \\ \hat{\Sigma} &= \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} X_i X_i^t - \bar{X} \bar{X}^t \text{ and } \tilde{\Sigma} = \frac{1}{|\bar{\mathcal{C}}|} \sum_{i \in \bar{\mathcal{C}}} X_i X_i^t - \tilde{X} \tilde{X}^t \\ \bar{X}^{\mathcal{K}} &= \frac{1}{N} \sum_{k=1}^K \sum_{i=1}^{n_k} X_i^{(k)} \text{ and } \hat{\Sigma}^{\mathcal{K}} = \frac{1}{N} \sum_{k=1}^K \sum_{i=1}^{n_k} X_i^{(k)} [X_i^{(k)}]^t - \bar{X}^{\mathcal{K}} [\bar{X}^{\mathcal{K}}]^t.\end{aligned}$$

The Transfer Learning algorithm is then decomposed in 4 steps. We begin by estimating Ω using the CLIME-estimator (or any other estimator). From Proposition 5.2 by Cai et al. (2016), a modification has been made to the CLIME algorithm to force the sample covariance matrix $\hat{\Sigma}$ to be positive-definite by adding a diagonal matrix $|\mathcal{C}|^{-1}I_p$. The optimization problem is defined by

$$\begin{aligned}\hat{\Omega}_{CL} &= \arg \min_{\Omega \in \mathbb{R}^{p \times p}} \|\Omega\|_1 \\ \text{subject to } &\|(\hat{\Sigma} + |\mathcal{C}|^{-1}I_p)\Omega - I_p\|_{\infty} \leq \lambda_{CL},\end{aligned}\tag{4.2.6}$$

where $\lambda_{CL} > 0$ is a tuning parameter.

We then compute an estimation of $\Delta^{\mathcal{K}}$ based on the Moment Equation (4.2.5). The optimization problem is

$$\begin{aligned}\hat{\Delta}^{\mathcal{K}} &= \arg \min_{\Delta \in \mathbb{R}^{p \times p}} \|\Delta\|_1 \\ \text{subject to } &\|\Delta - [(\hat{\Omega}_{CL})^t \hat{\Sigma}^{\mathcal{K}} - I_p]\|_{\infty} \leq \lambda_{\Delta},\end{aligned}\tag{4.2.7}$$

where $\lambda_{\Delta} > 0$ is a tuning parameter. As explained in Section 4.1, we use the transpose of the CLIME estimator because as computed in problem (4.2.6) it is not a symmetric matrix. From the Moment Equation, a possible initial estimate for the algorithm could be given by $(\hat{\Omega}_{CL})^t \hat{\Sigma}^{\mathcal{K}} - I_p$.

Using the previous estimate, we now focus on obtaining the Transfer Learning estimator of Ω . The optimization problem is again derived from equation (4.2.5).

$$\begin{aligned}\hat{\Omega}_{TL} &= \arg \min_{\Omega \in \mathbb{R}^{p \times p}} \|\Omega\|_1 \\ \text{subject to } &\|(\hat{\Sigma}^{\mathcal{K}} - N^{-1}I_p)\Omega - (\hat{\Delta}^{\mathcal{K}} + I_p)\|_{\infty} \leq \lambda_{\Omega},\end{aligned}\tag{4.2.8}$$

where $\lambda_{\Omega} > 0$ is a tuning parameter. The optimization problem is different from the Moment Equation to ensure the positive-definiteness of $\hat{\Sigma}^{\mathcal{K}}$ by adding a diagonal matrix I_p/N to it.

As we said earlier, if the similarity is weak between an auxiliary study and the target study, $\hat{\Omega}_{TL}$ may not be a good estimator (at least worse than the single-task estimator). We will use a Model Selection aggregation to produce a final estimator $\hat{\Omega}$. To perform the

aggregation we want to select the optimal estimator between the CLIME estimator $\hat{\Omega}_{CL}$ and the Transfer Learning estimator $\hat{\Omega}_{TL}$. Instead of doing it for the whole estimator, we will choose for each column the best column estimator for Ω . Readjusting the Moment Equation vector-wise, we get

$$\Sigma\Omega_j - e_j = 0,$$

where e_j was defined in Section 4.1. Now Li et al. (2022) proposed to compute a weight vector $\hat{v}_j$ as

$$\tilde{\Sigma}(\hat{\Omega}_{CL,j}, \hat{\Omega}_{TL,j})v_j - e_j \approx 0. \quad (4.2.9)$$

As we are in a Model Selection aggregation setting, this vector can be either $(1, 0)^t$ or $(0, 1)^t$, selecting either $\hat{\Omega}_{CL,j}$ or $\hat{\Omega}_{TL,j}$. Note also that we use the sample covariance matrix from the test set to evaluate both estimators on different samples to preserve the independence between them. Multiplying equation (4.2.9) by $((\hat{\Omega}_{CL,j}, \hat{\Omega}_{TL,j})v_j)^t$ to the left, we get

$$v_j^t(\hat{\Omega}_{CL,j}, \hat{\Omega}_{TL,j})^t \tilde{\Sigma}(\hat{\Omega}_{CL,j}, \hat{\Omega}_{TL,j})v_j - v_j^t(\hat{\Omega}_{CL,j}, \hat{\Omega}_{TL,j})^t e_j \approx 0. \quad (4.2.10)$$

By the definition of e_j and defining the matrix $\widehat{W}(j)$ as

$$\widehat{W}(j) = (\hat{\Omega}_{CL,j}, \hat{\Omega}_{TL,j})^t \tilde{\Sigma}(\hat{\Omega}_{CL,j}, \hat{\Omega}_{TL,j}) = \begin{pmatrix} \hat{\Omega}_{CL,j}^t \tilde{\Sigma} \hat{\Omega}_{CL,j} & \hat{\Omega}_{CL,j}^t \tilde{\Sigma} \hat{\Omega}_{TL,j} \\ \hat{\Omega}_{TL,j}^t \tilde{\Sigma} \hat{\Omega}_{CL,j} & \hat{\Omega}_{TL,j}^t \tilde{\Sigma} \hat{\Omega}_{TL,j} \end{pmatrix},$$

equation (4.2.9) becomes

$$v_j^t \widehat{W}(j) v_j - v_j^t (\hat{\omega}_{CL,jj}, \hat{\omega}_{TL,jj})^t \approx 0.$$

In the paper, they decided to multiply the second term by 2 without further explanation. One can suppose that this decision was made to take account of the approximation. The final step of the Transfer Learning algorithm is then to compute

$$\hat{v}_j = \arg \min_{v \in \{(1,0)^t, (0,1)^t\}} v_j^t \widehat{W}(j) v_j - 2v_j^t (\hat{\omega}_{CL,jj}, \hat{\omega}_{TL,jj})^t,$$

and the final estimator of the algorithm is

$$\hat{\Omega}_j = (\hat{\Omega}_{CL,j}, \hat{\Omega}_{TL,j}) \hat{v}_j.$$

If we want to ensure that $\hat{\Omega}$ is symmetric, we can use $(\hat{\Omega} + \hat{\Omega}^t)/2$ in the optimization step. Now that the final estimator of the precision matrix has been obtained, it can be useful to use an edge detection method to get a more precise graph of the model.

4.3 Edge detection

By penalizing the Transfer Learning algorithm, we force the precision matrix estimated to respect the Moment Equation with a certain boundary without trying to determine the edges of the graph. Using small tuning parameters in the algorithm results in a sparse matrix where it is difficult to detect edges for the small weights and we may find many false edges by using large tuning parameters.

The solution is to perform simultaneous hypothesis tests of the form

$$H_{0,ij} : \omega_{ij} = 0 \text{ vs. } H_{1,ij} : \omega_{ij} \neq 0,$$

for $1 \leq i < j \leq p$ ($p(p-1)/2$ different tests). If $H_{0,ij}$ is rejected, we then conclude that there is an edge between the variables X_i and X_j . As well-known, the problem with multiple testing is the insurance of an adequate significance level α . The goal of the edge detection method with false discovery rate (FDR) control is to adjust the hypothesis testing for a significance level α and to control the FDR .

Denoting FP as the number of false positives, i.e. the number of wrongly rejected hypotheses, and P the number of rejected hypotheses, the false discovery proportion (FDP) and false discovery rate are computed as

$$FDP = \frac{FP}{P} \text{ and } FDR = E[FDP].$$

To perform a hypothesis test, one needs a test statistic with which one will be able to reject or not the null hypothesis. To this purpose, Li et al. (2022) conceived a powerful debiasing method that will then be used to develop a test statistic. Starting from a quadratic form for ω_{ij} , one can show that

$$\omega_{ij} = \Omega_i^t \Sigma \Omega_j = \Omega_i^t E[S] \Omega_j.$$

From this equation, they proposed a debiasing method for ω_{ij} and for Ω . Using any possible estimator $\hat{\Omega}^{(est)}$, compute

$$\hat{\omega}_{ij}^{(db)} = \hat{\omega}_{ij}^{(est)} + \hat{\omega}_{ji}^{(est)} - (\hat{\Omega}_j^{(est)})^t S \hat{\Omega}_i^{(est)} \quad (4.3.1)$$

$$\hat{\Omega}^{(db)} = \hat{\Omega}^{(est)} + (\hat{\Omega}^{(est)})^t - (\hat{\Omega}^{(est)})^t S \hat{\Omega}^{(est)}. \quad (4.3.2)$$

Note that we distinguish between $\hat{\omega}_{ij}^{(est)}$ and $\hat{\omega}_{ji}^{(est)}$ for non-symmetric matrices. We could use this debiasing method for example for the lasso-type estimator developed in Section 3.4 or of course for the Transfer Learning estimator. For the latter one we can use $\tilde{\Sigma}$ instead of S due to the low dependence with $\hat{\Omega}$ thanks to the aggregation step.

$$\hat{\omega}_{ij}^{(db)} = \hat{\omega}_{TL,ij} + \hat{\omega}_{TL,ji} - (\hat{\Omega}_{TL,j})^t \tilde{\Sigma} \hat{\Omega}_{TL,i}.$$

From this debiasing method, Li et al. (2022) eventually demonstrates in Theorem 3.1 that the Transfer Learning debiased estimator, under certain regularity conditions, follows asymptotically a normal distribution

$$\frac{\sqrt{\tilde{n}}(\hat{\omega}_{ij}^{(db)} - \omega_{ij})}{\sqrt{\sigma_{ij}^2}} \xrightarrow{\mathcal{L}} N(0, 1),$$

with $\sigma_{ij}^2 = \omega_{ii}\omega_{jj} + \omega_{ij}^2$ the asymptotic variance of $\hat{\omega}_{ij}^{(db)}$, and $\tilde{n}$ defined in Section 4.2.

By introducing an estimator for this quantity, $\hat{\sigma}_{ij}^2 = \hat{\omega}_{ii}\hat{\omega}_{jj} + \hat{\omega}_{ij}\hat{\omega}_{ji}$, and under certain regularity conditions, we have

$$\frac{\sqrt{\tilde{n}}(\hat{\omega}_{ij}^{(db)} - \omega_{ij})}{\sqrt{\hat{\sigma}_{ij}^2}} \xrightarrow{\mathcal{L}} N(0, 1).$$

Note that this is the same de-sparsified process found by Janková and van de Geer (2015) in Subsection 3.4.3 for the Glasso edge detection except that we use only the test set sample size. Li et al. (2022) assume furthermore that this procedure can be used for every possible estimators of Ω .

From this asymptotic behavior, confidence intervals with confidence level $(1 - \alpha)100\%$ can be computed as

$$IC_{1-\alpha}(\omega_{ij}) = \left[\hat{\omega}_{ij}^{(db)} - \Phi^{-1}\left(1 - \frac{\alpha}{2}\right) \times \frac{\hat{\sigma}_{ij}}{\sqrt{\tilde{n}}} \quad , \quad \hat{\omega}_{ij}^{(db)} + \Phi^{-1}\left(1 - \frac{\alpha}{2}\right) \times \frac{\hat{\sigma}_{ij}}{\sqrt{\tilde{n}}} \right].$$

Using this asymptotic normal distribution, one can then define a test statistic. Indeed, under the null hypothesis $H_0 : \omega_{ij} = 0$,

$$\hat{z}_{ij} = \frac{\sqrt{\tilde{n}} \hat{\omega}_{ij}^{(db)}}{\sqrt{\hat{\sigma}_{ij}^2}} \xrightarrow{\mathcal{L}} N(0, 1),$$

for $1 \leq i < j \leq p$.

We now develop a method for edge detection with *FDR* control. Define $\mathcal{H}_0 := \{(i, j) : \omega_{ij} = 0, 1 \leq i < j \leq p\}$ the set of true null hypotheses. We can reformulate the *FDP* and *FDR* as

$$FDP(t) = \frac{\sum_{(i,j) \in \mathcal{H}_0} \mathbb{1}(|\hat{z}_{ij}| \geq t)}{\max\{\sum_{1 \leq i < j \leq p} \mathbb{1}(|\hat{z}_{ij}| \geq t), 1\}},$$

and

$$FDR(t) = E[FDP(t)]$$

where $t \geq 0$ is a threshold for which we reject hypothesis $H_{0,ij}$ if $|\hat{z}_{ij}| \geq t$.

The goal then is to find an optimal value for t , for which we have many correctly rejected hypotheses and where we control the FDR at level $\alpha \in (0, 1)$. For this purpose, Liu (2013) defined

$$\hat{t}_0 = \inf \left\{ t \in [0, 2\sqrt{\log p}] : \frac{\sum_{(i,j) \in \mathcal{H}_0} \mathbb{1}(|\hat{z}_{ij}| \geq t)}{\max\{\sum_{1 \leq i < j \leq p} \mathbb{1}(|\hat{z}_{ij}| \geq t), 1\}} \leq \alpha \right\},$$

as an ideal value of t where $\inf$ denotes the infimum of a set, but unfortunately we do not know $\mathcal{H}_0$. An upper bound $2\sqrt{\log p}$ was defined for t because Liu (2013) showed in Section 6 that $P(\max_{(i,j) \in \mathcal{H}_0} |\hat{z}_{ij}| \geq 2\sqrt{\log p}) \rightarrow 0$ so we can focus on values of $t \in [0, 2\sqrt{\log p}]$.

To estimate $\sum_{(i,j) \in \mathcal{H}_0} \mathbb{1}(|\hat{z}_{ij}| \geq t)$, Liu (2013) proposed to use $2(1 - \Phi(t))|\mathcal{H}_0|$, where $\Phi(t)$ is the cumulative function of a standard normal distribution. Since we assume Ω to be sparse (many true zeros in the precision matrix), $|\mathcal{H}_0|$ can be estimated by $p(p-1)/2$ (which is the number of parameters to test) as we do not know in practice the true number of zero elements.

From this framework, Li et al. (2022) developed a similar method. Denote $m := p(p-1)/2$, we compute

$$\hat{t} = \inf \left\{ t \in [0, \sqrt{2 \log m - 2 \log \log m}] : \frac{2m(1 - \Phi(t))}{\max\{\sum_{1 \leq i < j \leq p} \mathbb{1}(|\hat{z}_{ij}| \geq t), 1\}} \leq \alpha \right\}, \quad (4.3.3)$$

and eventually reject the hypotheses for which $|\hat{z}_{ij}| \geq t$. If t from (4.3.3) does not exist, then set $\hat{t} = \sqrt{2 \log m}$. They decided to use $\sqrt{2 \log m - 2 \log \log m}$ instead of $2\sqrt{\log p}$ as an upper bound for t because it allows to relax a necessary condition used to ensure the asymptotic control of the FDP and FDR (condition (12) in Liu (2013)). Li et al. (2022) then showed in Theorem 4.1 that this method can asymptotically control the FDP and FDR under certain regularity conditions.

We now focus on the rates of convergence of the three investigated estimators, the Glasso, CLIME and Trans-CLIME estimator.

4.4 Rates of convergence

For the three estimators, regularization parameters were needed to penalize the estimation of the different matrices. Without any prior information on an optimal λ , any λ can be used but the convergence of the estimators are not ensured. For this purpose, the rates of convergence, which quantifies how fast the estimation error of an estimator relative to the target study decreases when n is increasing, are set under certain regularity conditions such as an optimal value for λ . The results obtained can be then used to ensure the convergence of the estimators in practice.

In this section we aim to introduce the theory behind rates of convergence and compare the rates of convergence between the different estimators under the Frobenius norm and the spectral norm.

To demonstrate these rates, we need to impose some conditions on the underlying distribution and more importantly on the tail probability of the distribution of the random variable X_i . This is what we call concentration inequalities, defined by Vershynin (2018, Chapter 2), where we want to quantify how X_i deviates around its mean μ ,

$$P(|X_i - \mu| > t) \leq \text{something small.}$$

An example is the well known Tchebyshev's inequality where one can show that, for a random variable with finite variance $\sigma^2 < \infty$ and mean μ ,

$$P(|X_i - \mu| \geq k\sigma) \leq \frac{1}{k^2},$$

for any $k > 0$.

Unfortunately this bound is not optimal and other conditions may results in better rates. For example the class of sub-Gaussian distributions is often used due to its practical properties.

As an important result, for any random variable X belonging to the sub-Gaussian distribution, all the following properties are equivalent (meaning that there is only an absolute constant factor differing between each term K_i , i.e. there exists some constant $C > 0$ such that $K_j \leq CK_i$, for any $i, j = 1, \dots, 4$) :

1. $P(|X_i| \geq t) \leq 2 \exp(-t^2/K_1^2)$, $\forall t \geq 0$ and some constant $K_1 > 0$.
2. $(E[|X_i|^p])^{1/p} \leq K_2 \sqrt{p}$, $\forall p \geq 1$ and some constant $K_2 > 0$.
3. $E[\exp(\kappa^2 X_i^2)] \leq \exp(\kappa^2 K_3^2)$, $\forall \kappa$ such that $|\kappa| \leq 1/K_3$ and some constant $K_3 > 0$.
4. $E[\exp(X_i^2/K_4^2)] \leq 2$ for some constant $K_4 > 0$.

These equivalences are powerful as one only needs to prove one of these properties to prove that X_i belongs to the class of sub-Gaussian distributions and assuming that the random variable belongs to this class lets us use these properties to prove the needed rates of convergence.

We will assume in the following that the variables are sub-Gaussian and introduce the different rates of convergence for each estimator under either the Frobenius norm or the spectral norm.

4.4.1 Rates of convergence : CLIME and Trans-CLIME

To investigate the rates of convergence of the CLIME and Trans-CLIME estimators, assume that $\frac{1}{C} \leq \Lambda_{\min}(\Sigma) \leq \Lambda_{\max}(\Sigma) \leq C$ and $\frac{1}{C} \leq \Lambda_{\min}(\Sigma^{(k)}) \leq \Lambda_{\max}(\Sigma^{(k)}) \leq C$ for some constant $C > 0$, where $\Lambda_{\min}(\cdot)$ and $\Lambda_{\max}(\cdot)$ denote the lowest and largest eigenvalues of Σ .

These assumptions, under the normality of the r.v.e. X and $X^{(k)}$, allow these r.v.e. to follow a sub-Gaussian distribution (indeed any random variable following a Gaussian distribution follows also a sub-Gaussian distribution) and specify the positive-definiteness of the covariance matrix.

Moreover, let us define the parameter space of the investigated matrices :

$$\mathbb{G}(d, h) = \left\{ (\Omega, \Omega^{(1)}, \dots, \Omega^{(K)}) : \max_{1 \leq j \leq p} \sum_{i=1}^p \mathbb{1}(\omega_{ij} \neq 0) < d, \max_{1 \leq k \leq K} D_1(\Omega^{(k)}, \Omega) \leq h \right\},$$

where d and h are by definition positive. This parameter space implies that the maximum column (or row) sparsity of Ω is below a certain threshold d and the divergence between the target study and the auxiliary studies is at most below or equal to h . Note that for the CLIME estimator, the parameter space is in this case $\mathbb{G}(d, \infty)$ as no auxiliary studies are used.

We now define the rates of convergence for the CLIME estimator. First assume that the conditions detailed above are respected, that $p \geq c_1 n$, for some constant $c_1 > 0$, and $d^2 \log p = o(n)$, where the Little-o notation $o(\cdot)$ is defined in Chapter 2. For the estimation we compute the CLIME estimator with

$$\lambda_{CL} = c_2 \sqrt{\frac{\log p}{n}},$$

for some large enough constant $c_2 > 0$. Then, for any true model belonging to the parameter space $\mathbb{G}(d, \infty)$, one can show that

$$E \left[\max \left\{ \frac{1}{p} \|\hat{\Omega}_{CL} - \Omega\|_F^2, |\hat{\Omega}_{CL,j} - \Omega_j|_2^2 \right\} \right] \leq \frac{Cd \log p}{n},$$

for some constant $C > 0$ and any fixed $1 \leq j \leq p$.

In other words, the convergence rate for the CLIME estimator is

$$E \left[\max \left\{ \frac{1}{p} \|\hat{\Omega}_{CL} - \Omega\|_F^2, |\hat{\Omega}_{CL,j} - \Omega_j|_2^2 \right\} \right] = \mathcal{O}\left(\frac{d \log p}{n}\right),$$

where the Big-O notation $\mathcal{O}(\cdot)$ is defined in Chapter 2. This quantity tells us that the maximum between the Frobenius norm and the column-wise ℓ_2 -norm does not exceed $d \log p / n$ under a proper choice of the tuning parameter λ_{CL} .

We now define the convergence rate for the Trans-CLIME estimator. Under the same conditions, we define

$$\delta_{h,\Omega} := \min\{(\|\Omega\|_{L_1} + h)h\sqrt{\log p/n}, h^2\},$$

where the term h comes from the definition of the parameter space $\mathbb{G}(s, h)$. Also let the Trans-CLIME estimator be computed with

$$\lambda_\Delta = c_3(\|\Omega\|_{L_1} + h)\sqrt{\log p/n},$$

and

$$\lambda_\Omega = c_3\sqrt{\log p/N},$$

where c_1 is a large enough constant, and we remind the reader that n is the sample size of the target study and N is the combined sample size over all the auxiliary studies.

If $p \geq c_4 N$ for some constant $c_4 > 0$ and $\frac{d \log p}{N} + \min\{\delta_{h,\Omega}, \frac{d \log p}{n}\} = o(1)$ (meaning that this term converges to 0), then for any true model belonging to the parameter space $\mathbb{G}(s, h)$, one can show that

$$E \left[\max \left\{ \frac{1}{p} \|\hat{\Omega}_{TL} - \Omega\|_F^2, |\hat{\Omega}_{TL,j} - \Omega_j|^2 \right\} \right] \leq C \left(\frac{d \log p}{n + N} + \min \left\{ \delta_{h,\Omega}, \frac{d \log p}{n} \right\} + \frac{1}{n} \right),$$

for some constant $C > 0$ and any fixed $1 \leq j \leq p$.

We now have an upper bound for the Frobenius Norm and the column-wise ℓ_2 -norm under a proper choice of tuning parameters λ_Δ and λ_Ω . The term $d \log p/(n + N)$ is due to the fact that all $K + 1$ matrices have the same row-sparsity bound and are computed with $n + N$ independent samples.

The term $\delta_{h,\Omega}$ comes from the fact that we want to estimate the Δ^κ matrix in row-wise ℓ_2 -norm. Fortunately, this term tends to 0 as $n \rightarrow \infty$. It isn't determined by N because the divergence matrix is based on the target sample. When the similarities between the auxiliary studies and the target one are high, i.e. h is small, $\delta_{h,\Omega}$ can be very small and so too the rate of convergence.

The last term $1/n$ is the cost for the aggregation step, which turns out to be small as the sample size grows. Eventually, the term $\min\{\delta_{h,\Omega}, d \log p/n\}$ comes also from the aggregation step. Indeed, $d \log p/n$ is the convergence rate for the CLIME estimator and $\delta_{h,\Omega}$ comes from the Trans-CLIME estimator, so by using an aggregation step we will choose the smallest one.

From the previous convergence rate, we observe that we obtain a faster rate of convergence for the Trans-CLIME estimator when $N \gg n$ and $\delta_{h,\Omega} \ll d\sqrt{\frac{\log p}{n}}$.

4.4.2 Rates of convergence : Glasso

We now develop the rate of convergence for the Glasso estimator. Under the condition that the rescaled random variables $X_j/\sqrt{\sigma_{jj}}$ are sub-Gaussian and that the sample size respects $n = \Omega(d^2 \log p)$, where $\Omega(\cdot)$ is the Big-Omega notation defined in Chapter 2, Ravikumar et al. (2011) then proved that

$$\begin{aligned} \|\hat{\Omega}_{GL} - \Omega\|_F^2 &= \mathcal{O}_p\left(\frac{(v+p) \log p}{n}\right), \\ \|\hat{\Omega}_{GL} - \Omega\|_2^2 &= \mathcal{O}_p\left(\frac{\min\{(v+p), d^2\} \log p}{n}\right), \end{aligned}$$

where $\mathcal{O}_p$ is the Big-O in probability notation defined in Chapter 2, and v is the number of non-zero off-diagonal elements defined in Section 3.4.3. By comparison with the rates given for the CLIME and Trans-CLIME estimator, as $(v+p) > d$, the rate of convergence of the Glasso is slower than the rate of the CLIME and Trans-CLIME (as we showed that the later can have faster rate of convergence than the CLIME estimator).

From the previous subsection, a range of optimal values for the tuning parameters of the Trans-CLIME (and CLIME) estimation are

$$\begin{aligned} \lambda_{CL} &= c_2 \sqrt{\frac{\log p}{n}}, \\ \lambda_{\Delta} &= \|\hat{\Omega}_{CL}\|_{L_1} \sqrt{\frac{\log p}{n}}, \\ \lambda_{\Omega} &= c_3 \sqrt{\frac{\log p}{N}}. \end{aligned}$$

For the Glasso estimator, we will use the tuning parameter recommended in Janková and van de Geer (2015)

$$\lambda_{GL} = c_4 \sqrt{\frac{\log p}{n}},$$

for some positive and large enough constants c_2 , c_3 and c_4 .

This last section concludes our chapter on Transfer Learning. In the next chapter, we will simulate data from a Multivariate Normal Distribution with different parameters to compare the Transfer Learning method with the CLIME estimator and the Glasso estimator.

Chapter 5

Simulations

In this chapter we aim to measure the performance of the Trans-CLIME estimator against two competitors, the CLIME and Glasso estimators, which will be estimated only on the target study while the Trans-CLIME estimator will also use auxiliary studies.

To compare the different estimators, three types of graph structured Ω matrices will be simulated :

- A banded matrix with bandwidth 8. For $1 \leq i, j \leq p$, $\omega_{ij} = 2 \times 0.6^{|i-j|} \mathbb{1}(|i-j| \leq 7)$.
- A hub matrix with $p/10$ rows with sparsity 11 and the rest of the rows with sparsity 2. Let $\omega_{ij} = 0.5$ for $i = 10(k-1) + 1$ and $i+1 \leq j \leq i+9$ with $1 \leq k \leq p/10$, $\omega_{ij} = 1$ for the diagonal elements and $\omega_{ij} = 0$ elsewhere. We then add a diagonal matrix to ensure that the minimum eigenvalue of the matrix is at least 0.05 for positive-definiteness.
- A random matrix. Let $\omega_{1,ij} = \mathbb{1}(i=j) + u_{ij}$, where $u_{ij} \sim U(0, 0.8)$. We then threshold the matrix $(\Omega_1 + \Omega_1^t)/2$ so that the first 20 largest values in each row and column are kept, the rest being set to 0. We further compute $\Omega = \Omega_1 + \max\{(0.1 - |\Lambda_{\min}(\Omega_1)|), 0\} I_p$ to ensure the positive-definiteness of the matrix.

Note that the banded and random matrices come from Li et al. (2022). A plot of the different graphs with $p = 50$ is illustrated in Figure 5.1. We will perform the simulations in two parts. First we will use the banded and random matrices as in Li et al. (2022) to compare the results of the simulations with theirs. Then we will perform additional simulations with the hub matrix and a modified banded matrix to understand how the procedures behave with modifications of the data generating process. In each case, the simulations are iterated 100 times.

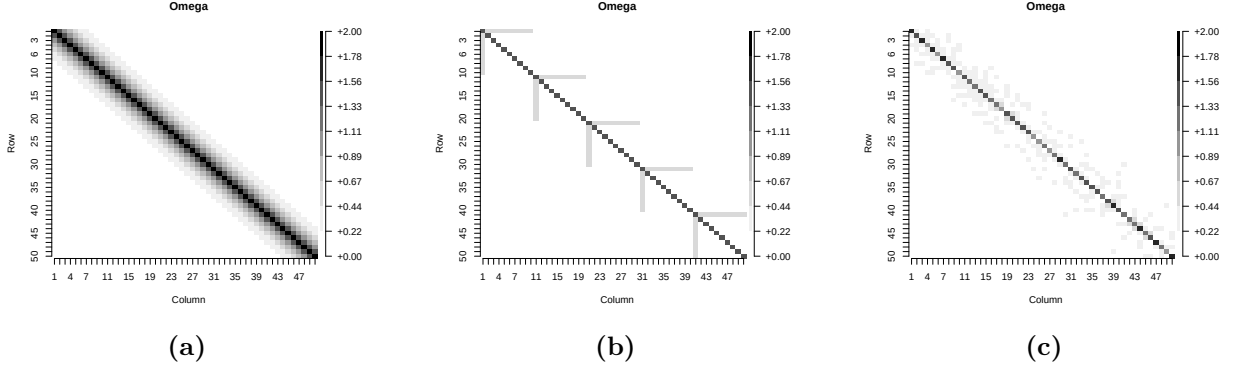


Figure 5.1: Three different types of graph structure with $p = 50$. (a) Banded, (b) Hub and (c) Random.

In each of these situations, different parameters can be modified. The target study sample size n , the number of auxiliary studies K and their sample size $n_k, k = 1, \dots, K$ and the number of parameters p . We denote by $A \subseteq \{1, \dots, K\}$ the set of informative auxiliary studies (i.e. with high similarity with the target study, or low $D_q(\Omega^{(k)}, \Omega)$) which can also be modulated.

The divergence matrix $\Delta^{(k)}$ will be generated as follows. If $k \in A$, $\Delta^{(k)} = \{\delta_{ij}^{(k)}\}$ with $\delta_{ij}^{(k)} = 0$ with probability 0.9 and $\delta_{ij}^{(k)} \sim U(-20/p, 20/p)$ with probability 0.1. If $k \notin A$, $\delta_{ij}^{(k)} = 0.5 \cdot \mathbb{1}(i = j) + \zeta_{ij}$ where ζ_{ij} is equal to 0.4 with probability 0.3 and 0 with probability 0.7. We then compute $\Sigma^{(k)}$ using equation (4.2.2), symmetrize the matrix and ensure its positive-definiteness by adding the diagonal matrix $(0.1 - |\Lambda_{\min}(\Sigma^{(k)})|)I_p$ to $\Sigma^{(k)}$.

For each estimator, tuning parameters are needed. The values $\lambda_{CL} = 2c_1\sqrt{\log p/n}$ will be used for the CLIME estimation. For the Trans-CLIME, $\lambda_{\Delta} = 2\|\hat{\Omega}_{CL}\|_1\sqrt{\log p/n}$ and $\lambda_{\Omega} = 2c_1\sqrt{\log p/n}$ will be used for the estimation of the divergence matrix and the precision matrix respectively. The same positive constant c_1 will be used for the CLIME and Trans-CLIME estimation. For the Glasso, we will use $\lambda_{GL} = c_2\sqrt{\log p/n}$ where c_2 is a positive constant. The values c_1 and c_2 are chosen from different possible values depending on the situation to minimize the loss function defined in equation (5.0.1) using a 5-fold cross-validation procedure.

During the estimation of the Trans-CLIME estimator, an aggregation step is performed. For this purpose, the sample is split and $[4n/5]$ samples are used for the estimation of the precision matrix and $[1n/5]$ samples are used to perform the aggregation in a two-fold approach.

To evaluate the performance of an estimator, we use the negative log-likelihood based on test samples x_i^{test} independent of the study samples, generated from $X^{\text{test}} \sim N(0, \Sigma)$ for $n_{\text{test}} = 100$. Explicitly, we compute

$$L = \frac{1}{p} \left[\frac{1}{2n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \text{tr}(x_i^{\text{test}} (x_i^{\text{test}})^t \hat{\Omega}_+) - \frac{1}{2} \log |\hat{\Omega}_+| \right], \quad (5.0.1)$$

where $\hat{\Omega}_+$ is the positive definite projection of any estimator previously symmetrized. A square matrix M is said to be a positive definite projection if $M^2 = M$ and M is positive definite. This step was done using an algorithm already made by Li et al. (2022).

The Frobenius norm, spectral norm, Matthews Correlation Coefficient (*MCC*), average coverage (AvgCov) of the confidence intervals (*CI*) are computed for every estimator. The *MCC* and AvgCov are used to assess the performance of the different estimators for predicting the true zeros/non-zeros of the precision matrix. The *MCC* is computed as

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}},$$

where *TP* is the number of true positives - true non-zeros correctly predicted, *TN* the true zeros correctly predicted, *FP* the number of false positives and *FN* the number of false negatives. The distinction between these four values is shown in Table 5.0.1.

Table 5.0.1: Summary of the possible results

		Estimation	
		$\hat{\omega}_{ij} \neq 0$	$\hat{\omega}_{ij} = 0$
Truth	$\omega_{ij} \neq 0$	<i>TP</i>	<i>FN</i>
	$\omega_{ij} = 0$	<i>FP</i>	<i>TN</i>

The *MCC* coefficient is known to be useful to measure the performance of a classification task (in this case whether the edge is absent or not between two nodes).

To measure the average coverage of the *CI* for each estimator, at each iteration we evaluated if the true precision matrix element ω_{ij} belonged to its corresponding estimated $CI_{1-\alpha}(\omega_{ij})$, resulting in an indicator matrix with zeros and ones. We then averaged these matrices over the 100 iterations and obtained the average coverage for two specific sets, the active set $A_{ct} := \{(i, j) : \omega_{ij} \neq 0\}$ denoting the set of true non-zero elements (diagonal elements included) and the non-active set $\bar{A}_{ct} := \{(i, j) : \omega_{ij} = 0\}$ denoting the set of true zero elements. A confidence level of $1 - \alpha = 0.9$ was used for the FDR and *CI* computations.

5.1 Banded matrix

We perform multiple simulations where every parameter is set to its default value except for one specific parameter whose value changes progressively. The default settings are $p = 100$, $n = 150$, $n_k = 250$ for $k = 1, \dots, K$, $K = 5$ and $|A| = 2$. The range of values investigated for each parameter are

- $p \in \{30, 50, 75, 100, 150, 200\}$.
- $n \in \{50, 150, 250, 500, 1000, 10000\}$.
- $n_k \in \{50, 150, 250, 500, 1000\}$ or similarly $N \in \{250, 750, 1250, 2500, 5000\}$.
- $|A| \in \{2, 3, 5, 7, 9, 11\}$ with K fixed at 11 in this case.
- $K \in \{2, 3, 5, 7, 9, 11\}$ with $|A|$ fixed at 2.

For each estimator, results are averaged over the 100 iterations. As the CLIME and Glasso estimators do not depend on the parameters n_k , K and $|A|$, we have averaged the results over all the different parameter values and only a straight line is shown in all the corresponding graphs. The 10 possible values for the constant of the tuning parameters are $c_1, c_2 \in \{0.3, 0.4, 0.5, \dots, 1, 1.1, 1.2\}$ except when p is varying, where we accept 15 possible values $c_1, c_2 \in \{0.1, 0.2, 0.3, \dots, 1.3, 1.4, 1.5\}$. This change is due to the high dependency of the constant on p , mostly when p is very low or high. Only 10 (and 15) values were used due to computational time constraint. We acknowledge that more precise results would have been obtained if more values were considered.

The Frobenius and Spectral norms as well as the MCC with n (in $\log_{10}$ transformation) and N (total auxiliary sample size) varying are displayed in Figure 5.2.

As expected, the three estimators perform better when the sample size of the target study increases but the trans-CLIME estimator performs as good as the CLIME estimator for each metric. This can be explained by the use of 3 non-informative auxiliary studies out of 5 to estimate the precision matrix. The Glasso estimator performs best when n is low (50 samples) but the performance decreases against the CLIME and Trans-CLIME as n increases.

The Trans-CLIME estimator seems to not be impacted by the increasing number of auxiliary sample sizes N for the Frobenius norm and the MCC , and performs poorly for the Spectral norm. As the number of non-informative studies is higher than informative studies, the estimator seems to be more influenced by the non-informative studies than the informative ones.

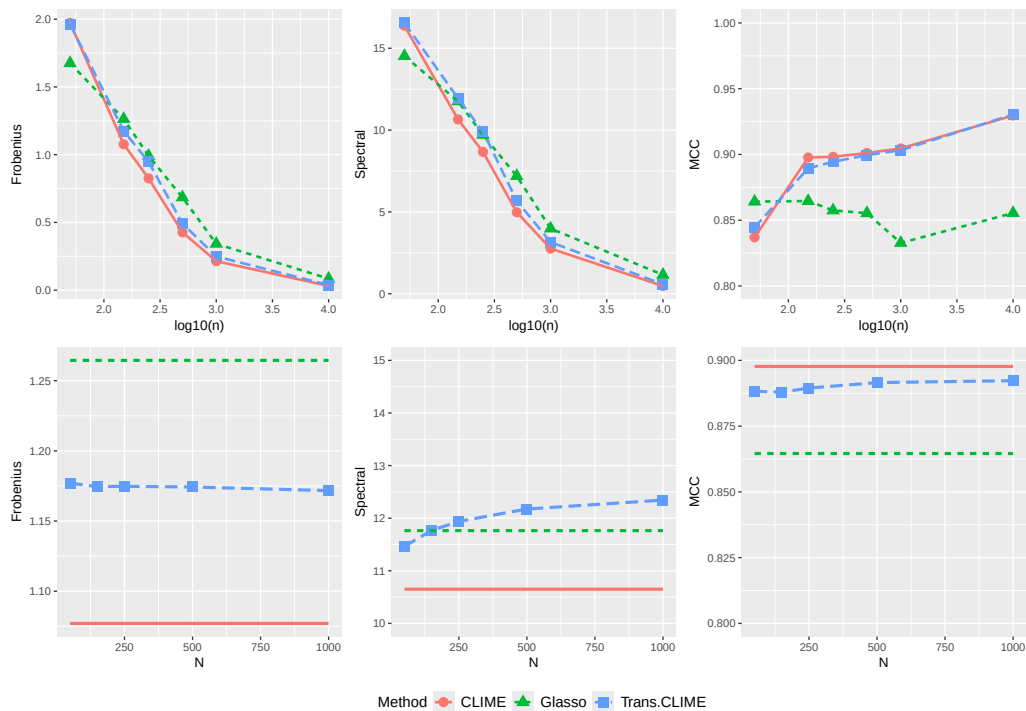


Figure 5.2: Frobenius norm, Spectral norm and MCC for different values of n (in $\log_{10}$ transformation) and N for the banded matrix.

Average coverage of the CI is displayed in Tables 5.1.1 and 5.1.2. As expected, nominal levels of average coverage are obtained with n increasing, but the Glasso estimator is slower than the CLIME and Trans-CLIME estimators to attain nominal levels. Trans-CLIME has same performance than CLIME for the non-active set but has lower performance for the active set. Coverage are always lower in the active set, meaning that the estimators perform well on estimating the true zeros but do not recover well the non-zero element information.

	$n =$	50	150	250	500	1000	10000
Trans-CLIME	$\text{AvgCov}_{A_{ct}}$	0.30	0.28	0.26	0.51	0.71	0.85
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.91	0.94	0.94	0.93	0.92	0.91
CLIME	$\text{AvgCov}_{A_{ct}}$	0.30	0.32	0.34	0.59	0.77	0.87
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.90	0.94	0.94	0.93	0.92	0.91
Glasso	$\text{AvgCov}_{A_{ct}}$	0.47	0.34	0.34	0.36	0.63	0.72
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.98	0.96	0.95	0.94	0.92	0.91

Table 5.1.1: Average Coverage for the different estimators with varying n for the banded matrix.

	$N =$	250	750	1250	2500	5000
Trans-CLIME	$\text{AvgCov}_{A_{ct}}$	0.29	0.29	0.28	0.28	0.28
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.93	0.93	0.94	0.94	0.94
CLIME	$\text{AvgCov}_{A_{ct}}$	0.32				
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.94				
Glasso	$\text{AvgCov}_{A_{ct}}$	0.34				
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.96				

Table 5.1.2: Average Coverage for the different estimators with varying N for the banded matrix.

The Frobenius and Spectral norms as well as the MCC with K and $|A|$ varying are displayed in Figure 5.3. A huge improvement can be seen when $K = 2$ and $|A| = 11$, that is, when all auxiliary studies are informative, and the performance decreases as the number of non-informative studies increases (as K increases or $|A|$ decreases). This also shows that the Trans-CLIME estimator does not perform so well if one of the auxiliary study is not informative anymore which can be a problem in practice.

CLIME has the best performance in terms of MCC and Trans-CLIME has better results when K is high or $|A| = 9$, which can be counter-intuitive when looking at the Frobenius and Spectral results. A look at Figure 5.4 where TP and TN are displayed with K and $|A|$ varying lets us understand these results. As the number of informative studies increases, the proportion of TN slowly increases but the proportion of TP decreases more importantly at the same time, causing the MCC to decrease overall. This means that the Trans-CLIME has better performance for recovering the true zero elements but not so for the true non-zero elements, where CLIME has better performance.

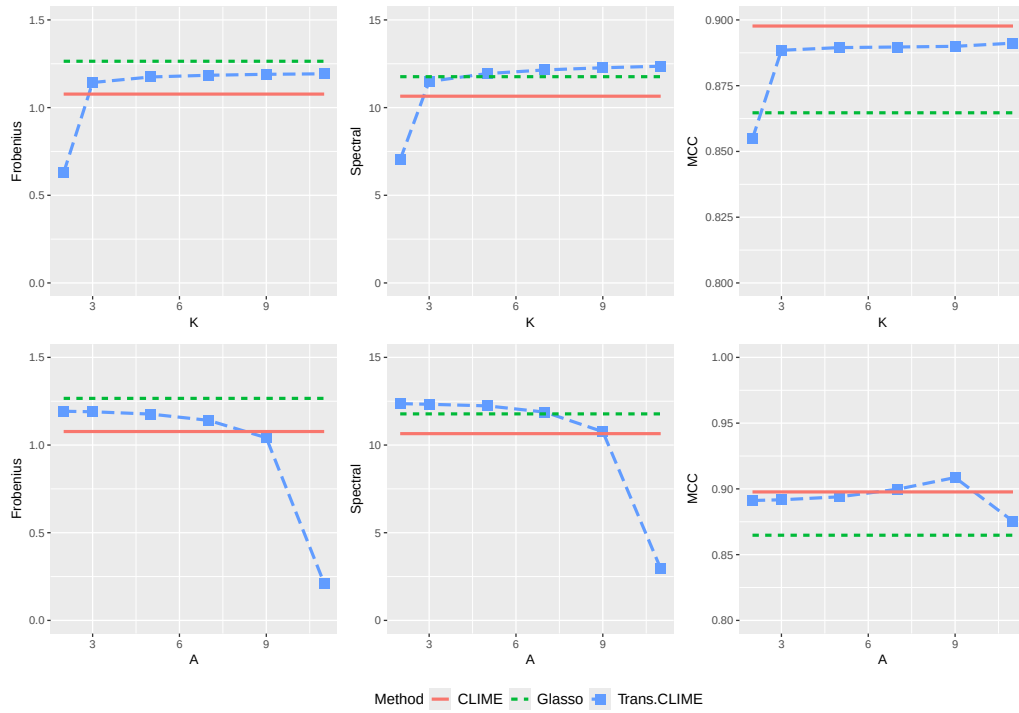


Figure 5.3: Frobenius norm, Spectral norm and MCC for different values of K and $|A|$ for the banded matrix.

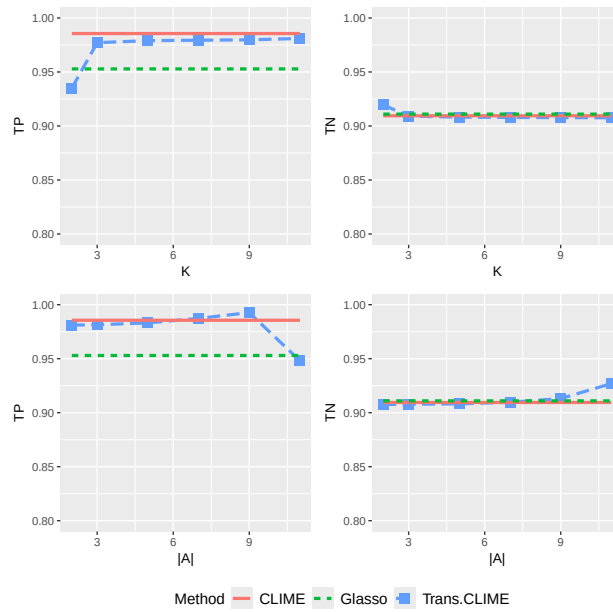


Figure 5.4: TP and TN for different values of K and $|A|$ for the banded matrix.

Average coverage of the CI is displayed in Tables 5.1.3 and 5.1.4. Again, better results are obtained when all auxiliary studies are informative. As before, nominal levels for the active set are not obtained but as seen with the sample size increasing, if only informative studies are used, nominal levels are obtained faster. Better results are obtained for the Trans-CLIME when $|A| = 11$ (with $K = 11$) than when $K = 2$ (with $|A| = 2$) meaning that from a set of informative studies, adding more informative studies will result in better performance.

	$K =$	2	3	5	7	9	11
Trans-CLIME	$\text{AvgCov}_{A_{ct}}$	0.64	0.29	0.28	0.28	0.28	0.28
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.91	0.94	0.94	0.94	0.94	0.94
CLIME	$\text{AvgCov}_{A_{ct}}$	0.32					
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.94					
Glasso	$\text{AvgCov}_{A_{ct}}$	0.34					
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.96					

Table 5.1.3: Average Coverage for the different estimators with varying K for the banded matrix.

	$ A =$	2	3	5	7	9	11
Trans-CLIME	$\text{AvgCov}_{A_{ct}}$	0.28	0.27	0.27	0.28	0.30	0.89
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.94	0.94	0.94	0.95	0.96	0.92
CLIME	$\text{AvgCov}_{A_{ct}}$	0.32					
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.94					
Glasso	$\text{AvgCov}_{A_{ct}}$	0.34					
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.96					

Table 5.1.4: Average Coverage for the different estimators with varying $|A|$ for the banded matrix.

Finally the Frobenius and Spectral norms as well as the MCC for different values of p are displayed in Figure 5.5. As in the case with varying n and N , the Trans-CLIME estimator does not perform better than the CLIME estimator but is better than the Glasso estimator in terms of MCC and Frobenius norm. While Trans-CLIME and CLIME diverge in the same magnitude, the Glasso estimator diverges more as p increases. This can be seen particularly for the Spectral norm where the Trans-CLIME estimator performs better than the Glasso estimator after a certain threshold p .

Average coverage of the estimators for different values of p is illustrated in Table 5.1.5. In the active set, CLIME and Glasso outperform uniformly the Trans-CLIME estimator. On the other hand, the Trans-CLIME estimator performs better on the non-active set as p increases. From these results, we conclude that the Trans-CLIME estimator could be a good choice when p is large to estimate the non-zero elements.

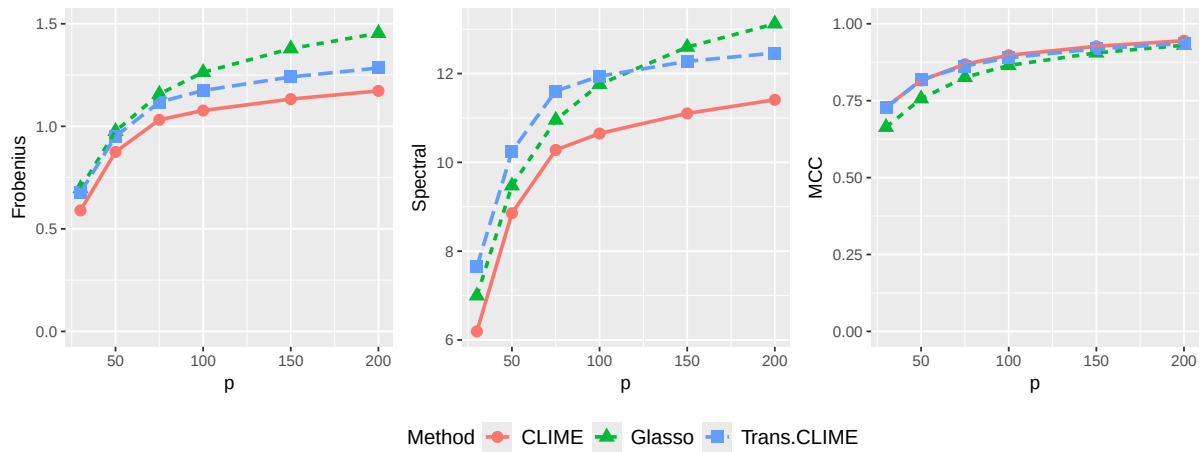


Figure 5.5: Frobenius norm, Spectral norm and MCC for different values of p for the banded matrix.

	$p =$	30	50	75	100	150	200
Trans-CLIME	$\text{AvgCov}_{A_{ct}}$	0.56	0.37	0.29	0.28	0.27	0.27
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.94	0.94	0.94	0.94	0.93	0.93
CLIME	$\text{AvgCov}_{A_{ct}}$	0.69	0.46	0.34	0.32	0.30	0.29
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.93	0.94	0.94	0.94	0.95	0.95
Glasso	$\text{AvgCov}_{A_{ct}}$	0.67	0.48	0.38	0.34	0.30	0.29
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.93	0.95	0.95	0.96	0.97	0.98

Table 5.1.5: Average Coverage for the different estimators with varying p for the banded matrix.

From these simulations, it is clear that the Trans-CLIME estimator does not perform so well when the sample size is low and a low number of auxiliary studies are informative. Furthermore, the Glasso estimator is uniformly worse in average coverage for the non-active set.

We now study the same simulations performed on the random matrix defined earlier.

5.2 Random matrix

We perform the same simulations as in the banded matrix setting with the same range of values for each parameter.

As previously, the results are averaged over 100 iterations for each estimator. Even though we have averaged the results for the CLIME and Glasso for n_k , K and $|A|$ in the previous simulations, we will not do the same here as the true precision matrix was modified for each setting (that is, we used the same matrix for all the iterations with $n = 50$, but not the same than with $n = 150$). It is not possible anymore to assume that the results would be the same for every configurations. We assume that better comparisons would have been obtained if the same random matrix was used for every simulations.

The 10 possible values for the constant of the tuning parameters are $c_1, c_2 \in \{0.1, 0.2, 0.3, \dots, 0.8, 0.9, 1\}$ except when p is varying, where we accept 15 possible values $c_1, c_2 \in \{0.1, 0.2, 0.3, \dots, 1.3, 1.4, 1.5\}$. This change is due to the high dependency of the constant when p is high.

The Frobenius and Spectral norms as well as the MCC with n (in $\log_{10}$ transformation) and N varying are displayed in Figure 5.6. As before, the results from the Trans-CLIME estimator are not better than the CLIME estimator when n is varying for the same reasons. For $n = 50$, the Trans-CLIME and CLIME estimators have high difficulties to recover the true zero and non-zero elements, resulting in a low MCC or even worse a negative MCC , where the Glasso estimator performs much better.

When N is varying, the Frobenius norm does not change in comparison with CLIME and Glasso, which is confusing as the Spectral norm is increasing slightly with N . A possible explanation is the low number of values considered for the tuning parameters. Indeed, the constant chosen will have high impact on the performance of the estimators. Considering more values could enhance the performance and discard the possible confusions about the tendencies.

In order to confirm this, we performed another set of simulations with this time 20 possible values for the constants $c_1, c_2 \in \{0.15, 0.2, 0.25, \dots, 1, 1.05, 1.1\}$. The results are shown in Figure 5.7. Unfortunately, the same tendency appears in these simulations but is more pronounced for the spectral norm. Another possibility could be that the performance of the Trans-CLIME is more influenced by non-informative studies than by informative ones, resulting in worse results when more non-informative samples are available.

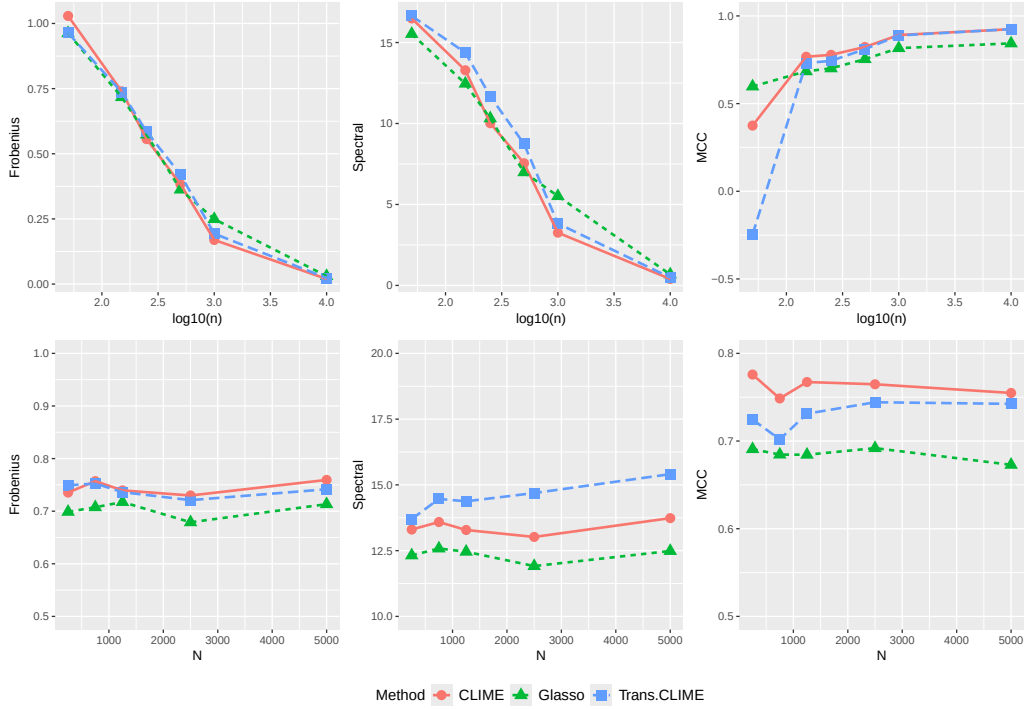


Figure 5.6: Frobenius norm, Spectral norm and MCC for different values of n (in $\log_{10}$ transformation) and N for the random matrix setting.

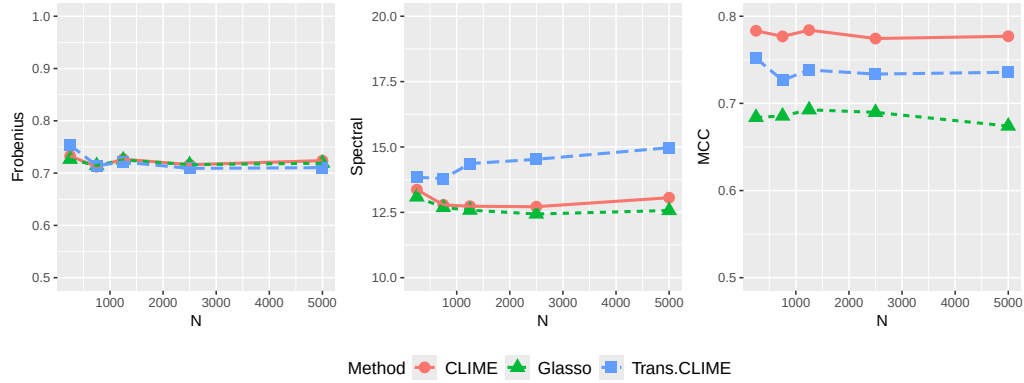


Figure 5.7: Frobenius norm, Spectral norm and MCC for different values of N for the random matrix when using 20 possible values for c_1 and c_2 .

The average coverage of the CI with n and N varying are displayed in Tables 5.2.1 and 5.2.2. As before, the estimators attain nominal levels of coverage asymptotically and better results are obtained for the non-active set. Trans-CLIME performances are, except for $n = 50$, always below the CLIME estimator for the active set but performs better for the non-active set. The Glasso estimator converges more quickly than in the banded matrix setting. When N is varying, the Trans-CLIME estimator does not perform better but actually worse, with the average coverage for the active set decreasing with N and increasing for the non-active set but its performances are still better than the CLIME and Glasso estimator for the non-active set.

	$n =$	50	150	250	500	1000	10000
Trans-CLIME	$\text{AvgCov}_{A_{ct}}$	0.74	0.72	0.76	0.75	0.85	0.89
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.92	0.90	0.90	0.90	0.90	0.90
CLIME	$\text{AvgCov}_{A_{ct}}$	0.73	0.74	0.78	0.77	0.86	0.89
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.87	0.92	0.92	0.91	0.90	0.90
Glasso	$\text{AvgCov}_{A_{ct}}$	0.87	0.79	0.77	0.79	0.79	0.88
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.97	0.94	0.92	0.91	0.90	0.90

Table 5.2.1: Average Coverage for the different estimators with varying n for the random matrix.

	$N =$	250	750	1250	2500	5000
Trans-CLIME	$\text{AvgCov}_{A_{ct}}$	0.74	0.72	0.72	0.69	0.68
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.89	0.89	0.90	0.91	0.91
CLIME	$\text{AvgCov}_{A_{ct}}$	0.74	0.72	0.74	0.72	0.71
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.92	0.92	0.92	0.92	0.92
Glasso	$\text{AvgCov}_{A_{ct}}$	0.80	0.78	0.79	0.79	0.78
	$\text{AvgCov}_{\overline{A_{ct}}}$	c 0.94	0.94	0.94	0.94	0.94

Table 5.2.2: Average Coverage for the different estimators with varying N for the random matrix.

The Frobenius and Spectral norms as well as the MCC with K and $|A|$ varying are displayed in Figure 5.8. As before, a huge improvement can be seen when $K = 2$ and $|A| = 11$. In terms of Spectral norm, the Trans-CLIME estimator performs better than the CLIME only when all auxiliary studies are informative, while the performance is better for the Frobenius norm even with some non-informative auxiliary studies included.

The Glasso estimator performs better than the other two (except when all auxiliary studies are informative) but not for the MCC . It seems that the Glasso does not recover well the information on the true zero and non-zero elements in general, resulting in lower MCC and average coverage.

The average coverage of the CI with varying K and $|A|$ is displayed in Tables 5.1.3 and 5.1.4. As K increases, the average coverage decreases for the active set and increases for the non-active set and nominal levels are obtained even if some auxiliary studies are non-informative. Best results are obtained when $K = 2$ and $|A| = 11$ (except for the non-active set when $|A| = 2$). Glasso estimator gives the worse results for the non-active set but is better in general for the active set than the Trans-CLIME and CLIME estimators.

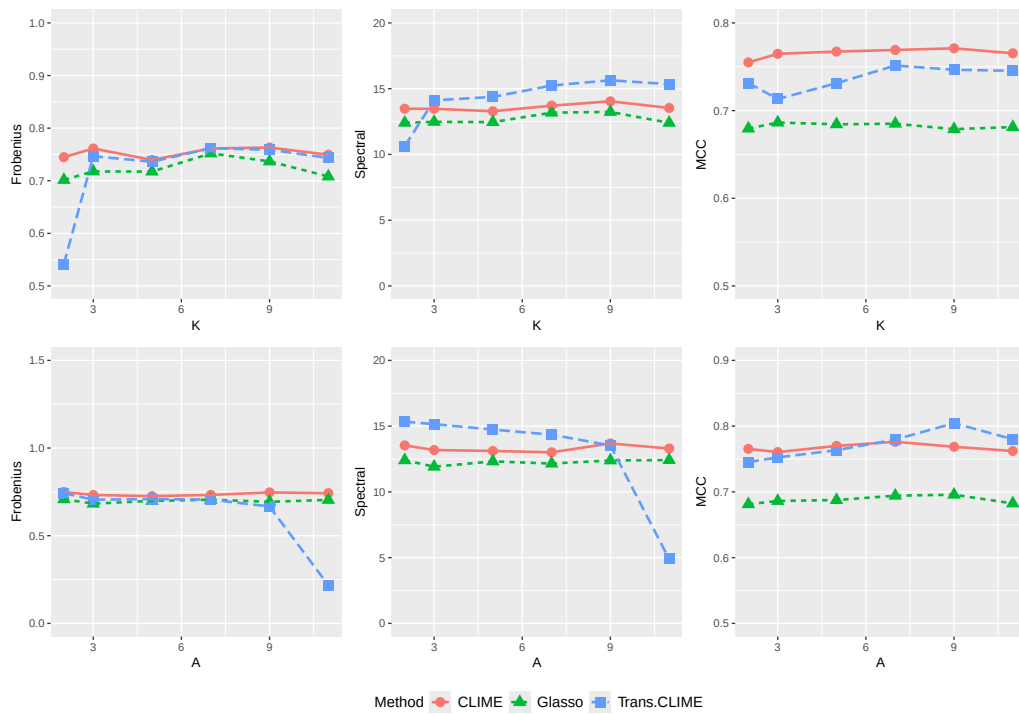


Figure 5.8: Frobenius norm, Spectral norm and MCC for different values of K and $|A|$ for the random matrix.

		$K =$	2	3	5	7	9	11
Trans-CLIME	$\text{AvgCov}_{A_{ct}}$		0.80	0.72	0.72	0.74	0.71	0.70
	$\text{AvgCov}_{\overline{A_{ct}}}$		0.89	0.90	0.90	0.90	0.91	0.91
CLIME	$\text{AvgCov}_{A_{ct}}$		0.73	0.73	0.74	0.76	0.73	0.73
	$\text{AvgCov}_{\overline{A_{ct}}}$		0.92	0.92	0.92	0.92	0.92	0.92
Glasso	$\text{AvgCov}_{A_{ct}}$		0.79	0.79	0.79	0.80	0.79	0.79
	$\text{AvgCov}_{\overline{A_{ct}}}$		0.94	0.94	0.94	0.94	0.94	0.94

Table 5.2.3: Average Coverage for the different estimators with varying K for the random matrix.

		$ A =$	2	3	5	7	9	11
Trans-CLIME	$\text{AvgCov}_{A_{ct}}$		0.90	0.69	0.73	0.72	0.70	0.89
	$\text{AvgCov}_{\overline{A_{ct}}}$		0.91	0.91	0.91	0.92	0.94	0.91
CLIME	$\text{AvgCov}_{A_{ct}}$		0.73	0.72	0.75	0.76	0.72	0.70
	$\text{AvgCov}_{\overline{A_{ct}}}$		0.92	0.92	0.92	0.92	0.92	0.92
Glasso	$\text{AvgCov}_{A_{ct}}$		0.79	0.79	0.81	0.80	0.79	0.77
	$\text{AvgCov}_{\overline{A_{ct}}}$		0.94	0.94	0.94	0.94	0.94	0.94

Table 5.2.4: Average Coverage for the different estimators with varying $|A|$ for the random matrix.

Finally, the Frobenius and Spectral norms as well as the MCC for different values of p are displayed in Figure 5.5. As in the banded matrix settings, the Trans-CLIME and CLIME estimators have the same tendency with the Trans-CLIME having always weaker results. Glasso is better than the Trans-CLIME and CLIME for the Spectral norm (except for $p = 50$ where Glasso and CLIME give similar results) and for the Frobenius norm when $p \leq 150$. Glasso still has difficulties to recover the true zero and non-zero elements as seen with the MCC .

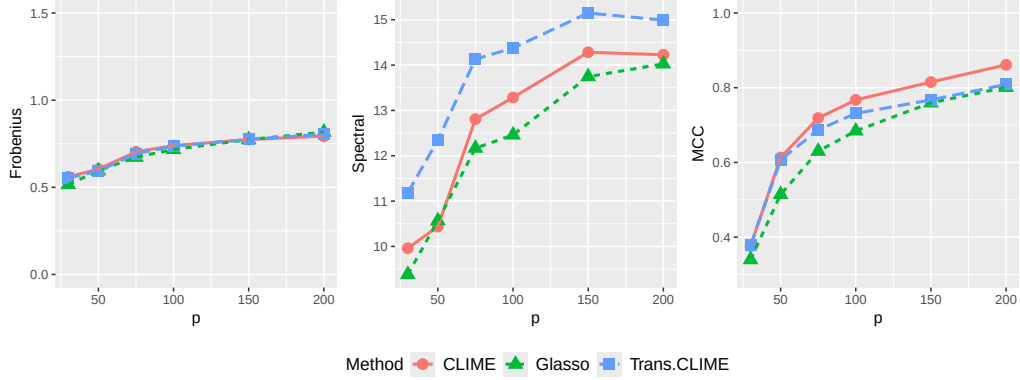


Figure 5.9: Frobenius norm, Spectral norm and MCC for different values of p for the random matrix.

The average coverage of the estimators for different values of p are illustrated in Table 5.1.5. On the active set, the Glasso estimator outperforms uniformly the Trans-CLIME estimator, which is not true for the CLIME estimator as $p \geq 150$. On the other hand, the Trans-CLIME estimator performs better on the non-active set as p increases. As seen with the MCC , the Glasso estimator does not perform well on the non-active set.

	$p =$	30	50	75	100	150	200
Trans-CLIME	$\text{AvgCov}_{A_{ct}}$	0.75	0.77	0.70	0.72	0.7	0.72
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.86	0.90	0.90	0.90	0.90	0.90
CLIME	$\text{AvgCov}_{A_{ct}}$	0.80	0.82	0.72	0.74	0.70	0.72
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.87	0.91	0.92	0.92	0.93	0.93
Glasso	$\text{AvgCov}_{A_{ct}}$	0.82	0.82	0.78	0.79	0.74	0.75
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.87	0.91	0.93	0.94	0.95	0.96

Table 5.2.5: Average Coverage for the different estimators with varying p for the random matrix.

From these simulations, it is clear that the Trans-CLIME estimator does not perform so well when the sample size is low and if a low number of auxiliary studies are informative. Glasso does not seem a good choice to provide confidence intervals for the true zero elements.

In the next simulations, we will use only informative studies (that is, $|A| = 5$ for $K = 5$) to see how much better the Trans-CLIME performs with regards to the other two estimators in the banded matrix settings.

5.3 Banded matrix when $|A| = 5$

When varying the number of auxiliary studies or the number of informative studies, huge differences occur when all auxiliary studies are informative. In these simulations, we aim to explore the performance of the Trans-CLIME estimator against the Glasso and the CLIME estimator when only informative studies are used. Default settings are $K = 5$ and $|A| = 5$ and n , n_k and p are varying.

In Figure 5.10, the Frobenius and Spectral norms as well as the MCC for the three estimators when n and N increase are displayed. As expected, Trans-CLIME outperforms the Glasso and CLIME estimators in terms of Frobenius and Spectral norms but not for the MCC where the CLIME estimator is always better. When n is very large, all estimators converge to the same value except for the MCC where the Glasso estimator is better for small sample size and then decreases. The Trans-CLIME estimator outperforms uniformly its competitors for any value of N . Furthermore, increasing the sample size of the auxiliary studies results in better performances, while this was not the case for the value of $|A|$ below K . These gains in performance are not linked to the MCC where Trans-CLIME does not perform better than CLIME.

In Tables 5.3.1 and 5.3.2 we present the average coverage of the estimators when $|A| = 5$ and n or N is varying. The average coverage for the active set is already close to 0.9 when $n = 150$ for the Trans-CLIME, compared to the other two estimators. For the non-active set, Trans-CLIME estimator stay close to 0.9 when n and N are varying. Except for some results, Trans-CLIME outperforms its competitors.

	$n =$	50	150	250	500	1000	10000
Trans-CLIME	AvgCov $_{A_{ct}}$	0.35	0.83	0.80	0.79	0.79	0.85
	AvgCov $_{\overline{A_{ct}}}$	0.91	0.92	0.92	0.91	0.91	0.91
CLIME	AvgCov $_{A_{ct}}$	0.31	0.32	0.34	0.58	0.77	0.87
	AvgCov $_{\overline{A_{ct}}}$	0.90	0.94	0.94	0.93	0.92	0.91
Glasso	AvgCov $_{A_{ct}}$	0.47	0.34	0.34	0.37	0.64	0.72
	AvgCov $_{\overline{A_{ct}}}$	0.98	0.96	0.95	0.94	0.92	0.91

Table 5.3.1: Average Coverage for the different estimators with varying n for the banded matrix with $|A| = 5$.

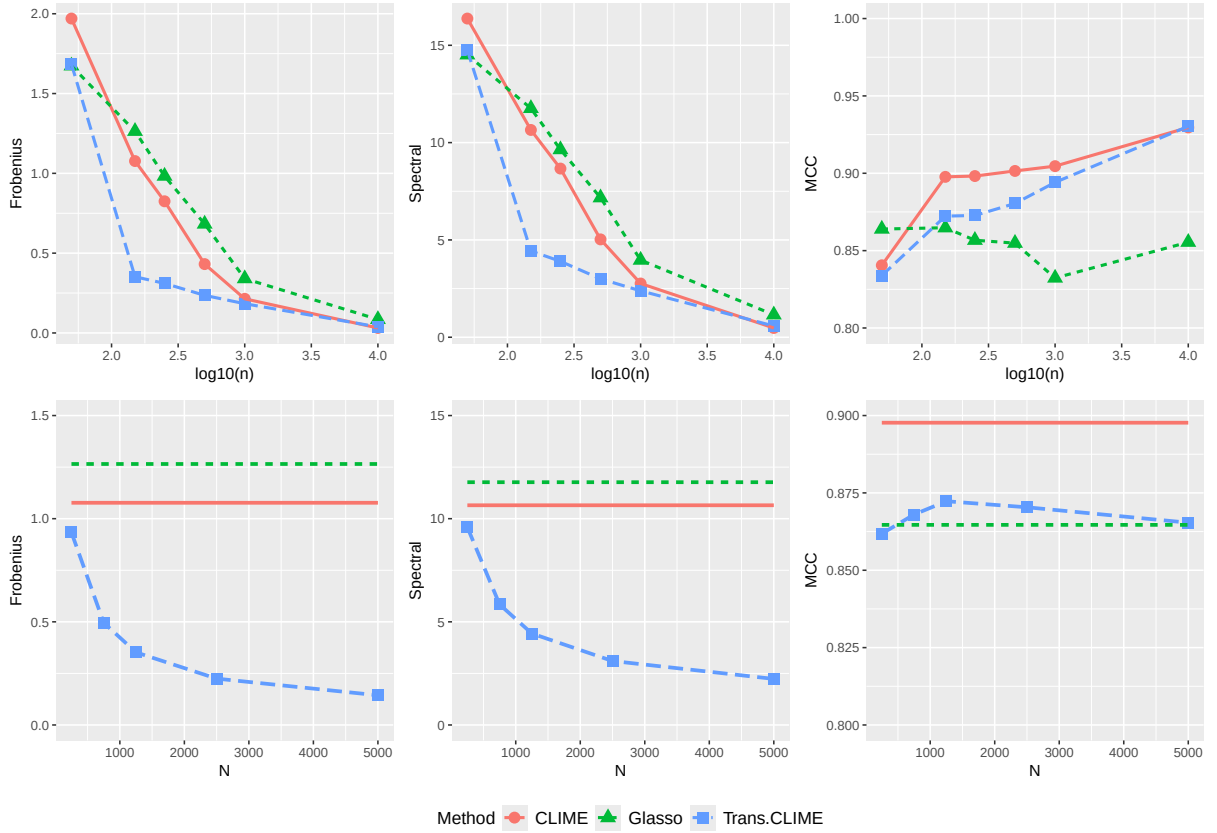


Figure 5.10: Frobenius norm, Spectral norm and MCC for different values of n and N for the banded matrix when $|A| = 5$.

		$N =$	250	750	1250	2500	5000
Trans-CLIME	$\text{AvgCov}_{A_{ct}}$		0.39	0.74	0.83	0.88	0.89
	$\text{AvgCov}_{\overline{A_{ct}}}$		0.92	0.92	0.92	0.92	0.91
CLIME	$\text{AvgCov}_{A_{ct}}$		0.32				
	$\text{AvgCov}_{\overline{A_{ct}}}$		0.94				
Glasso	$\text{AvgCov}_{A_{ct}}$		0.34				
	$\text{AvgCov}_{\overline{A_{ct}}}$		0.96				

Table 5.3.2: Average Coverage for the different estimators with varying N for the banded matrix with $|A| = 5$.

Simulation results when p is varying are displayed in Figure 5.11. Except for $p = 30$, Trans-CLIME performs better than the CLIME and Glasso in terms of Frobenius and Spectral norms, particularly for $p \geq 100$, where the results increase but at a lower rate. Even though the Trans-CLIME estimator is slightly better for the MCC at lower p , CLIME has better results for higher values of p .

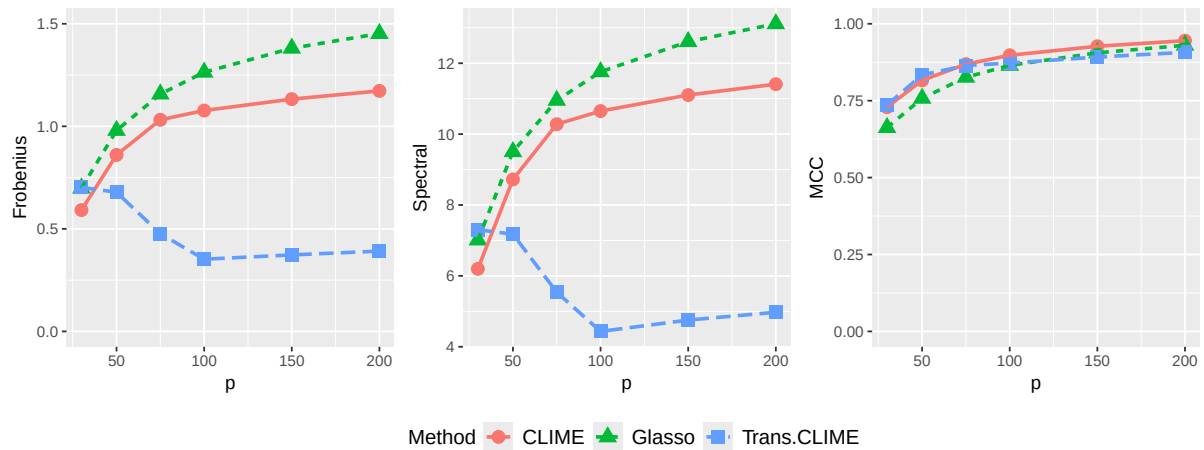


Figure 5.11: Frobenius norm, Spectral norm and MCC for different values of p for the banded matrix when $|A| = 5$.

Finally, the average coverage for the estimators when p is varying is shown in Table 5.3.3. When using only informative studies, the tendency of the Trans-CLIME is the opposite of CLIME and Glasso. While the performance of the CLIME and Glasso estimators decrease with p , the performance of Trans-CLIME is actually increasing, meaning that the information given by the auxiliary studies is more valuable as p increases.

	$p =$	30	50	75	100	150	200
Trans-CLIME	$\text{AvgCov}_{A_{ct}}$	0.54	0.54	0.74	0.83	0.82	0.81
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.95	0.95	0.93	0.92	0.92	0.92
CLIME	$\text{AvgCov}_{A_{ct}}$	0.69	0.47	0.34	0.32	0.30	0.29
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.93	0.94	0.94	0.94	0.95	0.95
Glasso	$\text{AvgCov}_{A_{ct}}$	0.68	0.48	0.38	0.34	0.31	0.29
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.93	0.95	0.95	0.96	0.97	0.98

Table 5.3.3: Average Coverage for the different estimators with varying p for the banded matrix when $|A| = 5$.

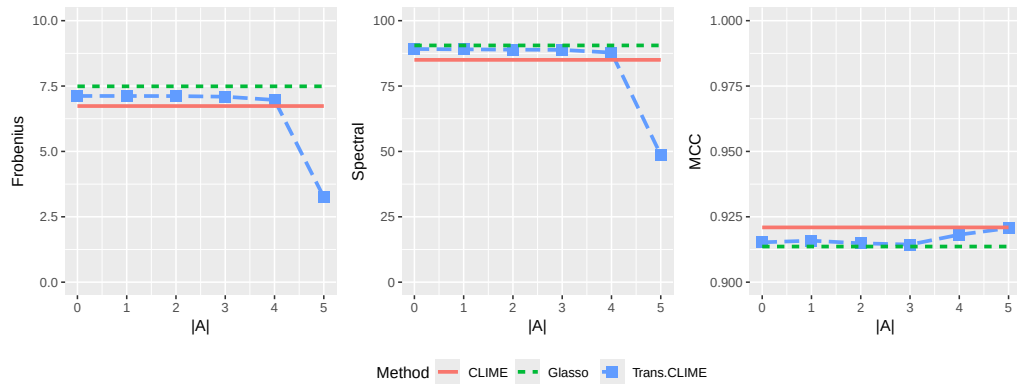
5.4 Modified Banded matrix

In their paper, Li et al. (2022) have demonstrated the performance of the Trans-CLIME estimator in some specific settings, when $n = 150$, $n_k = 300$, $K = 5$ and $p = 200$, with good results for the Trans-CLIME estimator already when $|A| = 3$. As the settings in our simulations are not the same, a strict comparison can not be done directly, but the performance of the Trans-CLIME against the CLIME and Glasso estimators does not yield such positive results. To compare them, simulations with the same settings (except that we used $n_k = 250$ instead of 300) and $|A|$ varying were performed using different matrices as true precision matrix.

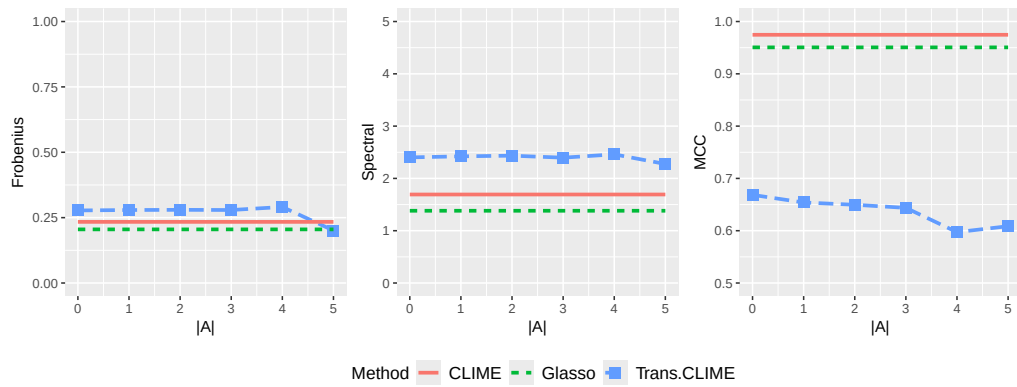
The first matrix investigated was again a banded matrix but this time with bandwidth 10. Each element was computed as $\omega_{ij} = 2 \times 0.7^{|i-j|} \mathbb{1}(|i-j| \leq 9)$, for $1 \leq i, j \leq p$. The second one was a hub matrix as described in the introduction of this chapter. Values considered for the constant of the tuning parameters are the same than in the banded matrix setting.

The Frobenius and Spectral norms as well as the *MCC* for different values of $|A|$ are illustrated in Figure 5.12 for the modified banded and hub matrix. The Trans-CLIME estimator only outperforms its competitors when $|A| = 5$, that is, when all auxiliary studies are informative, and only for the modified banded matrix, where the estimator was already performing better when $|A|$ was above three in the simulations of Li et al. (2022). In the case of the *MCC*, CLIME outperforms uniformly Trans-CLIME (and Glasso) and has similar results only when $|A| = 5$.

For the hub matrix, only the Frobenius norm of the Trans-CLIME for $|A| = 5$ shows better results, as the Spectral norm and *MCC* performances are always lower than the two competitors. The performance of the Trans-CLIME estimator in terms of *MCC* is very low compared to its competitors. Looking at Figure 5.13 where results for the *TP* and *TN* proportions are displayed, the weak performance of the Trans-CLIME to recover information from the true non-zero elements is clear, resulting in low *MCC* values.



(a)



(b)

Figure 5.12: Frobenius norm, Spectral norm and MCC for different values of $|A|$ for (a) the modified banded matrix and (b) the hub matrix.

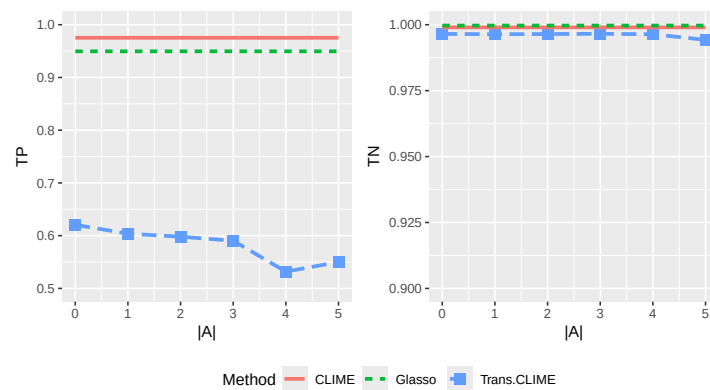


Figure 5.13: Proportion of TP and TN for different values of $|A|$ for the hub matrix.

The average coverage for the active and non-active set for the modified banded and hub matrix are presented in Tables 5.4.1 and 5.4.2. For the active set, CLIME has better results if not all auxiliary studies are informative, for which case the Trans-CLIME estimator performs better. Trans-CLIME outperforms CLIME for the hub matrix but its performance does not largely increase when $|A| = 5$, meaning that the estimator has difficulties recovering information from the true non-zero elements, which has already been highlighted with the TP results. In these settings, Glasso outperforms its two competitors for the hub matrix.

Glasso performs poorly for the modified banded matrix in the non-active set (and active set), while Trans-CLIME has in general better results than CLIME and Glasso even when only one auxiliary study is informative out of five. Results seem to be more unstable in the hub matrix settings, which may be due to the low number of possible values for the tuning parameters.

	$ A =$	0	1	2	3	4	5
Trans-CLIME	$\text{AvgCov}_{A_{ct}}$	0.21	0.21	0.21	0.21	0.20	0.42
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.94	0.89	0.89	0.90	0.92	0.95
CLIME	$\text{AvgCov}_{A_{ct}}$	0.22					
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.88					
Glasso	$\text{AvgCov}_{A_{ct}}$	0.20					
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.98					

Table 5.4.1: Average Coverage of the different estimators with varying $|A|$ for the modified banded matrix.

	$ A =$	0	1	2	3	4	5
Trans-CLIME	$\text{AvgCov}_{A_{ct}}$	0.53	0.46	0.46	0.46	0.44	0.49
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.90	0.87	0.87	0.87	0.86	0.89
CLIME	$\text{AvgCov}_{A_{ct}}$	0.18					
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.66					
Glasso	$\text{AvgCov}_{A_{ct}}$	0.85					
	$\text{AvgCov}_{\overline{A_{ct}}}$	0.95					

Table 5.4.2: Average Coverage of the different estimators with varying $|A|$ for the hub matrix.

5.5 Conclusion

From these simulations, some results can be highlighted. Firstly, to have the best performance in practice, one needs to consider as many values as possible for the tuning parameters. The Trans-CLIME estimator outperforms its competitors only when almost all auxiliary studies are informative. This would suggest in practice to use only studies for which there is evidence that there is high similarity with the target study.

Performances of the estimators depend highly on the underlying true precision matrix used as the general result can differ greatly between the hub, banded and modified banded matrix.

As expected, optimal results have been obtained with large sample sizes and Trans-CLIME and CLIME converge at a faster rate than Glasso. Depending on the matrix, the Trans-CLIME estimator performs better to detect the true zeros than true non-zeros, leading to poor MCC results. However, when using only informative studies the Trans-CLIME estimator is more resilient to decrease in performance when p is increasing, proving its usefulness when p is large (assuming a high proportion of informative studies).

We now dive into the last chapter concluding this master thesis, an application of the Trans-CLIME on brain tissues data, coming from the database GTEx.

Chapter 6

Application : GTEx brain tissues data

The Adult Genotype-Tissue Expression (GTEx) project gathered gene expressions from several tissues of the human body. In this thesis, we will evaluate the Trans-CLIME, CLIME and Glasso estimators in recovering the graph underlying some brain tissues. These specific tissues were selected due to the large number of different parts of the brain analyzed, resulting in 13 available brain tissues data ($K = 12$). These tissues and their sample size are summarized in Table 6.0.1.

Expressions from 56200 genes are available for each tissue. As the estimation of a precision matrix with this size is cumbersome (and interpretations would be difficult), we will select only genes that are known to be associated with peripheral and central neuronal systems. For this purpose, information from the GSEA Molecular Signatures Database will be used. Genes associated with neuron differentiation in the Central Neural System and Peripheral Neural System will be selected. Here are the two websites used to retrieve the data : https://www.gsea-msigdb.org/gsea/msigdb/human/geneset/GOBP_CENTRAL_NERVOUS_SYSTEM_NEURON_DIFFERENTIATION and https://www.gsea-msigdb.org/gsea/msigdb/human/geneset/GOBP_PERIPHERAL_NERVOUS_SYSTEM_NEURON_DIFFERENTIATION. The final dataset contains 187 genes. The gene expressions are already transformed in TPM (Transcripts per Kilobase Million) in order to normalize the samples between them.

Brain tissue	Abbreviation	Sample size
Brain amygdala	Amygdala	152
Brain cerebellum	Cerebellum	241
Brain cerebellar Hemisphere	C. Hemisphere	215
Brain cortex	Cortex	255
Brain frontal cortex	F. Cortex	209
Brain hippocampus	Hippocampus	197
Brain hypothalamus	Hypothalamus	202
Brain putamen basal ganglia	P.B. Ganglia	205
Brain spinal cord cervical	S.C. Cervical	159
Brain substantia nigra	S.N.	139
Brain anterior cingulate cortex	A.C. Cortex	176
Brain caudate basal ganglia	C.B. Ganglia	246
Brain nucleus accumbens basal ganglia	N.A.B. Ganglia	246

Table 6.0.1: Tissues name, abbreviation, and their corresponding sample size

6.1 Preprocessing and PCA

Before starting to compute the different estimators, we need to treat the data in order to obtain interpretable results. We first decided to look at the distribution of the different genes to ensure the needed normality with a QQplot. It is well known that log transformed gene expressions can follow a distribution closes to the normal distribution. As some genes contain zeros, we will use the transformation $\log(X + 1)$ to have 0 as a minimum. The QQplots of the 12 first genes are reported in Figure 6.1 colored by brain tissue.

Except gene LHX8 which contains a large number of zeros in several brain tissues, the other genes seem to roughly follow a normal distribution. Note that for some QQplots, gene expressions from brain tissue *C. Hemisphere* deviates from the rest, leading to possible dissimilarities with the other brain tissues.

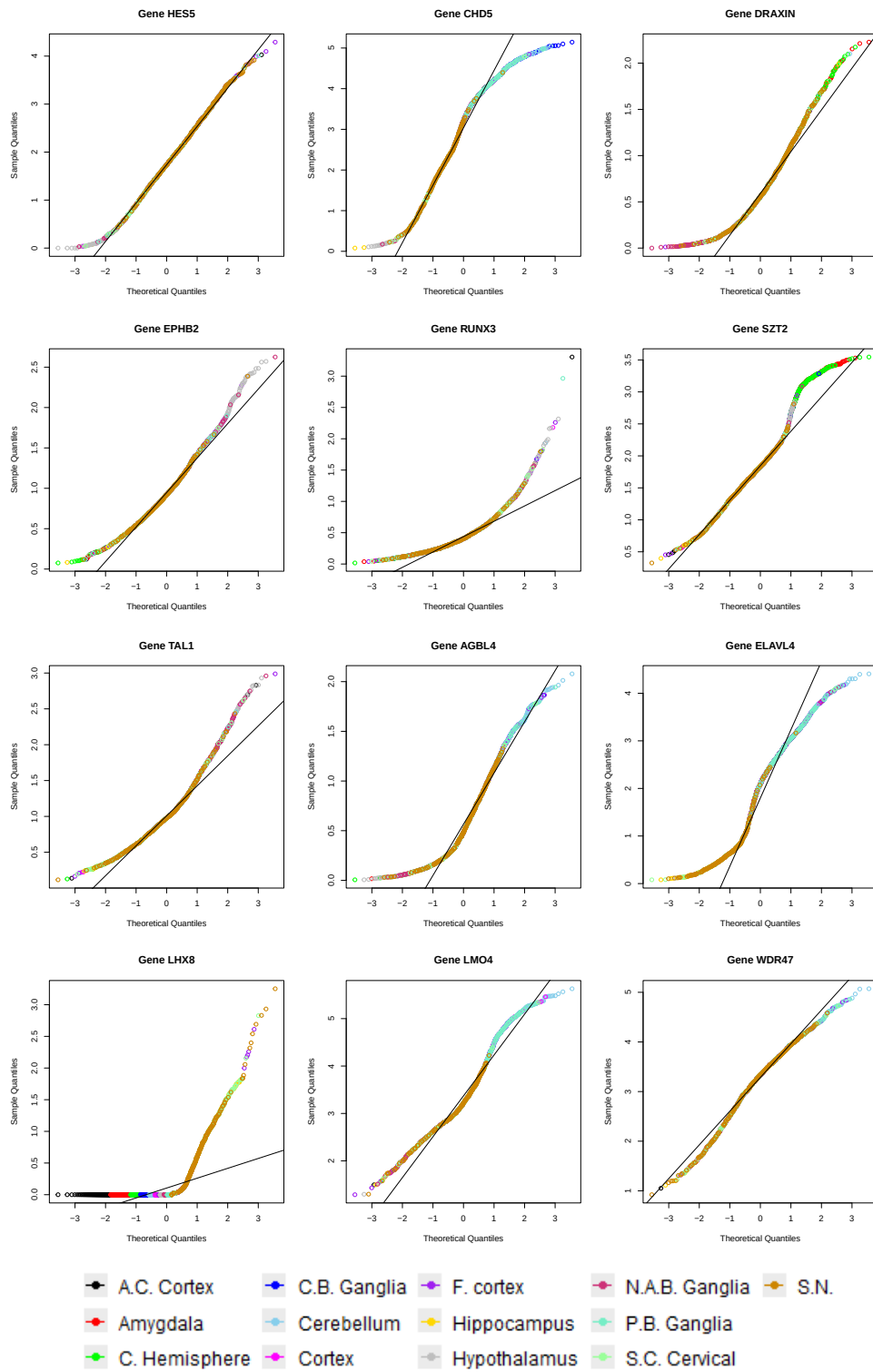
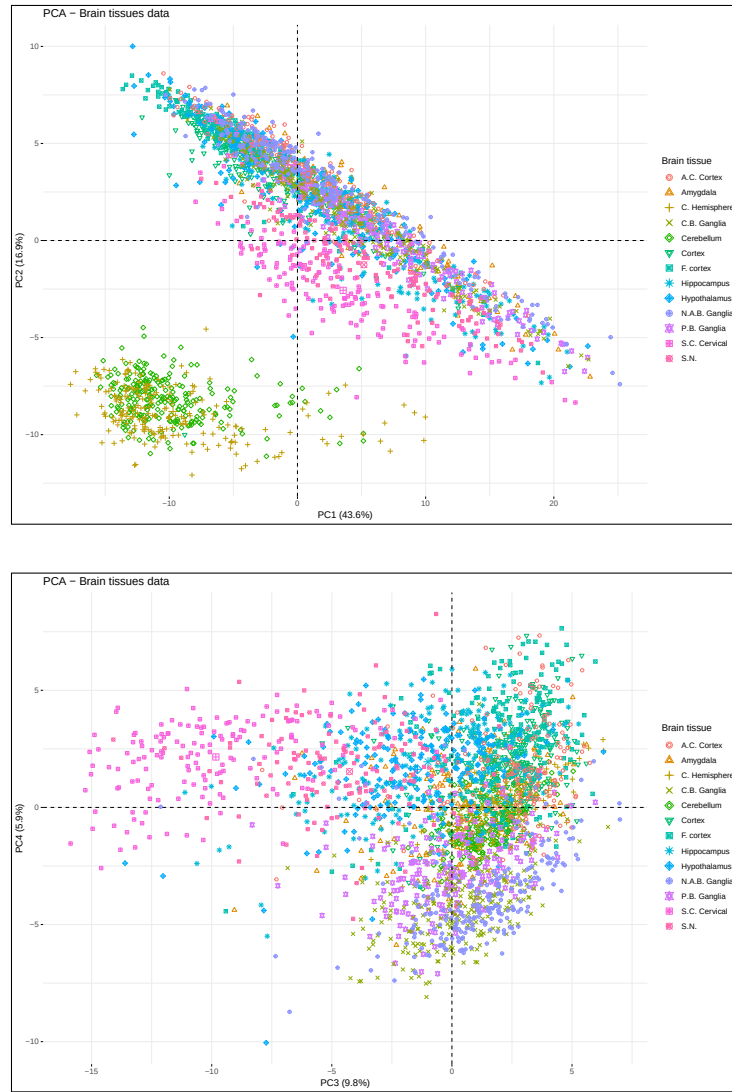


Figure 6.1: QQplots of the first 12 genes for the samples from all brain tissues. Genes are log-transformed and colored by brain tissue.

From this analysis, 147 genes are kept and the rest discarded due to marginal non-normality. A PCA is then performed on the log transformed data ($\log(X + 1)$) to evaluate the similarities between brain tissues. Score results for the first 6 PCs, colored by brain tissue, are displayed in Figure 6.2. From the first 2 PCs, it is clear that *C. Hemisphere* and *Cerebellum* tissues do not have large similarities with the other brains. From the simulation results, this could mean that better results could be obtained by discarding these tissues from the Trans-CLIME estimation. Other tissues such as *S.C. Cervical* and *Hypothalamus* slightly deviate from the other tissues in the PC3 and PC4 scores. As we expect that some dissimilarities between tissues exist, we assume that the performance of the Trans-CLIME will not be largely impacted by them and we will not take care of these ones.



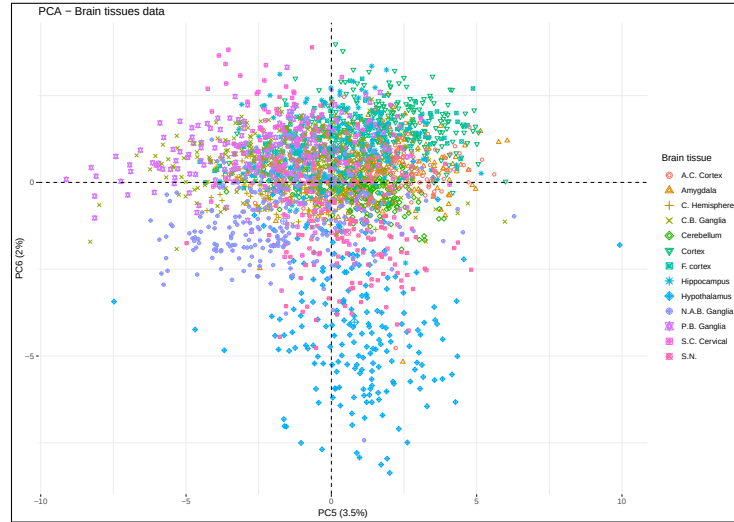


Figure 6.2: PCA scores for the first 6 PCs using $\log(X + 1)$ transformation on gene expressions, colored and shaped by brain tissue.

6.2 Estimation

To estimate the precision matrix of a specific target tissue, the samples will be split into a train and test set with 80% of the samples in the train set. The train set will be used to select c_1 and c_2 for the tuning parameters (with the same methodology as in Chapter 5) and to compute the different estimators. The test set will be used to compare the performance of the estimators by computing the loss function defined in equation (5.0.1). As the sample size is small and to ensure the robustness of the estimation, this process will be performed five times with different train/test sets. For the Trans-CLIME estimation, the train set will be split in five folds, four of them will be used to estimate the precision matrix and the last fold will be used for the aggregation step, this process being performed for each fold. Performance of the edge detection will be assessed for each estimator with $1 - \alpha = 0.9$ for the FDR control. This methodology will be used to estimate the graph structure of each brain tissue, where the target study will be a specific brain tissue and the rest used as auxiliary studies for the Trans-CLIME estimation.

Loss results are displayed in boxplots for the different estimators and brain tissues in Figure 6.3. From these results, the Glasso estimator outperforms largely the two other estimators in each brain tissue with a loss always between -1.5 and -1. The Trans-CLIME estimator has better performance than the CLIME estimator for some brain tissues but in general they have similar performance (in comparison to the Glasso estimator).

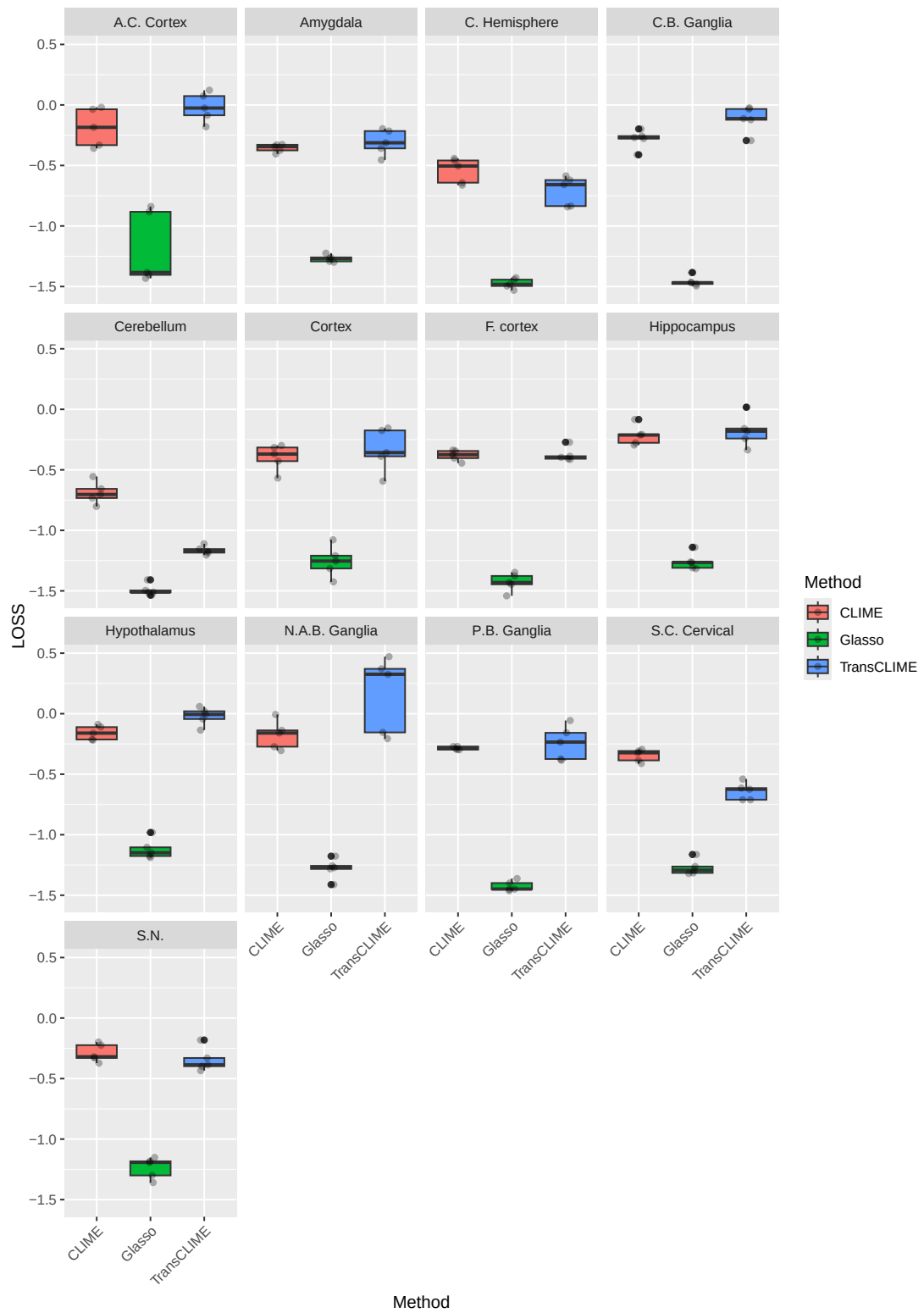


Figure 6.3: Boxplots of the loss results for the different estimators and brain tissues.

We focus on the difference of performance between the CLIME and Trans-CLIME estimator by subtracting the Trans-CLIME results from the CLIME ones, shown in Figure 6.4. Points above 0 mean better performance for the Trans-CLIME estimator. The Trans-CLIME estimator outperforms CLIME for only 4 studies out of 13 and the worst result is obtained for the N.A.B. Ganglia tissue.

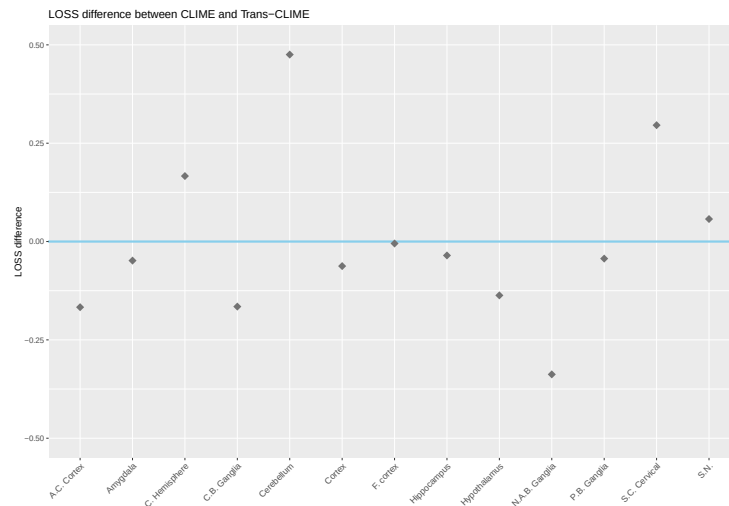


Figure 6.4: Loss differences between CLIME and Trans-CLIME for the 13 different brain tissues. Higher than 0 means Trans-CLIME has better performance.

The proportion of edges detected by FDR was also computed within each brain tissue for the different estimators (or by multiplicity testing for the Glasso). The results are displayed in Figure 6.5. Some jitter was added to identify the overlapping points and the same was applied to every plot with multiple points. Only the Glasso estimator has relevant results and obtained a sparse matrix, while the Trans-CLIME and CLIME estimators are, except for the Cerebellum tissue, highly dense, leading to graph structure not very interesting.

In Figure 6.6, the proportion of non-zero elements for the different estimators before applying an FDR control as well as the proportion of zero elements in the edges selected after the FDR control are displayed. The CLIME estimator initially contains few non-zero elements but the FDR control do not show the same results. From the Glasso estimator, only a few non-zero elements are considered as edges by the FDR control.

On the right picture, a huge part of the edges detected by the FDR control are zeros in the initial estimator for the CLIME and Trans-CLIME estimators. It is essential to recall that the FDR control can be seen as an additional estimator using the de-sparsified estimators, but this could mean that the estimation of the CLIME and Trans-CLIME estimators were not optimal to start with and led to many zeros while a lot of them were actually relevant.

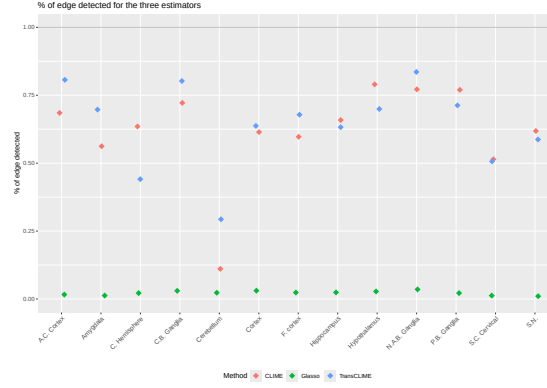


Figure 6.5: Proportion of edges detected for the different estimators for each brain tissue. Lower values denote a sparser matrix. Inspired from Li et al. (2022).

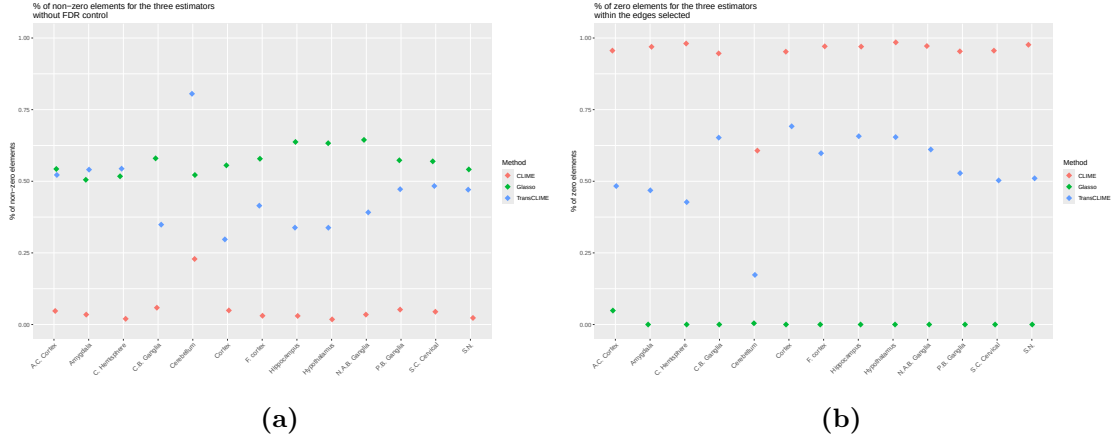


Figure 6.6: (a) Proportion of non-zero elements in the different estimators for each brain tissue. Lower values denote a sparser matrix. (b) Proportion of zero elements within the edges selected with FDR control for the different estimators for each brain tissue.

As seen in the PCA results, two brain tissues are very dissimilar from the rest, the *C. Hemisphere* and *Cerebellum* tissues. We now perform the same analysis discarding those two tissues to assess the performance of the Trans-CLIME estimator.

The loss results obtained for the three estimators are displayed in Figure 6.7. Unfortunately, not taking account of the two most dissimilar brain tissues did not result in better performance of the Trans-CLIME. By comparing these results with the previous ones, the loss obtained are more or less equivalent between the two estimations, with Glasso still being the clear winner.

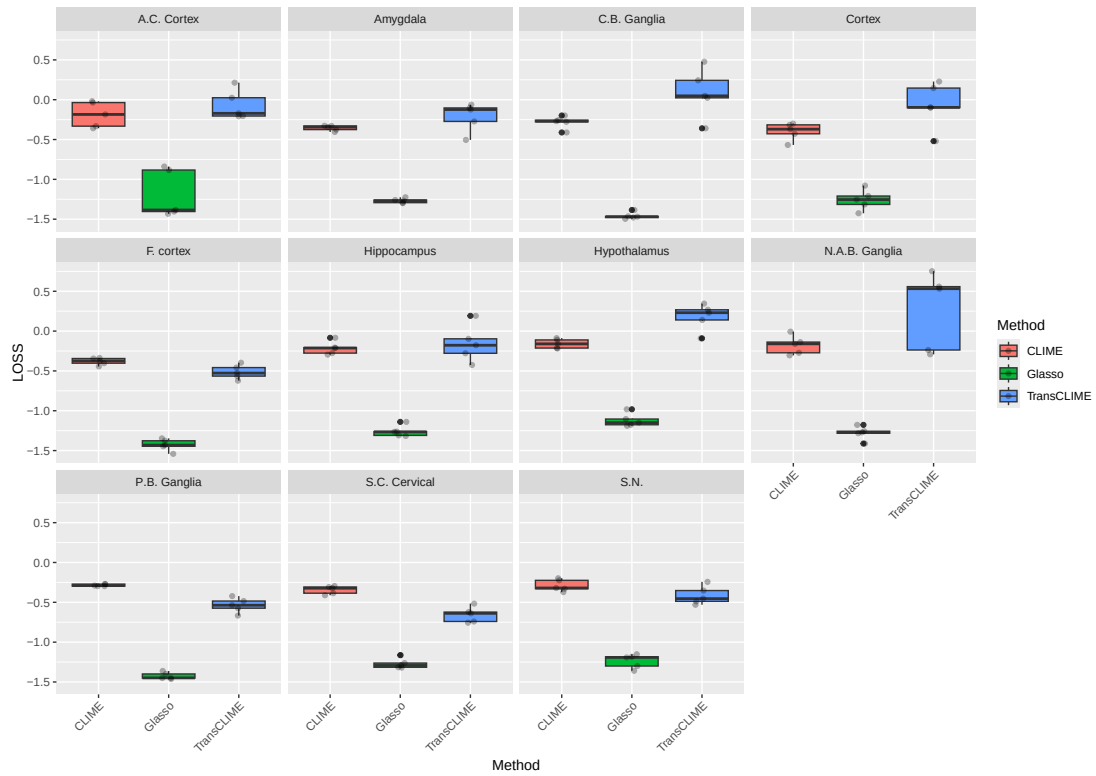


Figure 6.7: Boxplots of the loss results for the different estimators and brain tissues.

We focus only on the difference between the Trans-CLIME and CLIME loss results which are depicted in Figure 6.8. Performance of the Trans-CLIME did not change using the most dissimilar auxiliary brain tissues, with only 4 brain tissues out of 11 with better results for the Trans-CLIME. Again, the results for the N.A.B Ganglia tissue are the lowest for the Trans-CLIME against the CLIME estimator.

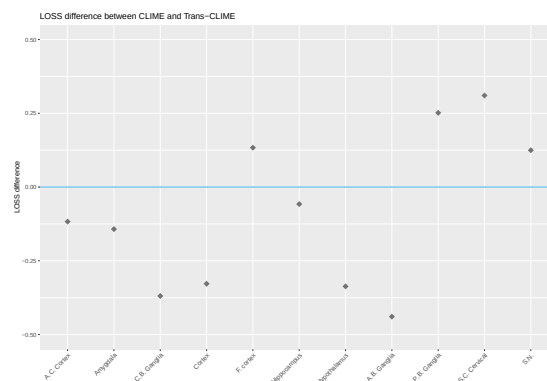


Figure 6.8: Loss differences between CLIME and Trans-CLIME for the 11 different brain tissues. Higher than 0 means Trans-CLIME has better performance. Inspired from Li et al. (2022).

Eventually the proportion of edges detected by the three estimators is shown in Figure 6.9. The same general results are obtained with the Glasso estimator giving very low percentage of significant edges while the CLIME and Trans-CLIME gives very high values with the Trans-CLIME detecting more edges than the CLIME estimator in general.

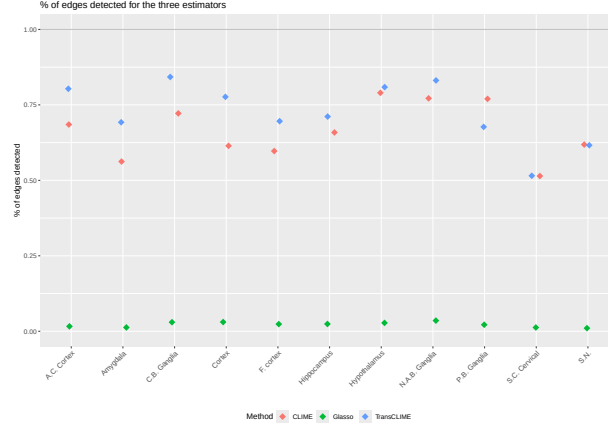


Figure 6.9: Proportion of edges detected for the different estimators and for each brain tissue. Lower values mean sparser matrix. Inspired from Li et al. (2022).

Comparing these results with the ones obtained by Li et al. (2022) raises an important issue. In their paper they showed the superior performance of the Trans-CLIME against the CLIME estimator, while this was not the case in our analysis. Without more information about the procedure they have performed to select the genes used in their final model and how they have selected the tuning parameters for the penalization, we do not have much insight about the possible problems other than the choice of the tuning parameters.

Indeed, while the Glasso estimation could converge with very low values for the tuning parameters, this was not the case for the CLIME and Trans-CLIME estimations where low values for the tuning parameter would resolve in the non-convergence of the estimators. To tackle this problem, we decided to select only values for which the convergence of the estimators were possible, leading to these results. It is important to note that we also do not have the same number of edges selected to be used in the model, as we used 147 genes while 141 genes were used in their paper.

To compare the estimators on the same basis, the next results will be performed on the same grid points of tuning parameters values for the cross-validation step. the values c_1, c_2 are chosen from 50 values, from 0.01 to 0.2 equally spaced. As said before, with this range of values, the CLIME estimator did not converge. In this case, Li et al. (2022) have suggested an heuristic value for the estimator,

$$\hat{\Omega} = (S + \lambda I)^{-1}$$

without any explanation on this decision. Also note that unfortunately, due to computational time problem, the graph structure of the *Cortex*, *C.B. Ganglia* and *N.A.B Ganglia* were not estimated.

The loss results for these new estimations are displayed in Figure 6.10. The CLIME performs poorly for the *F. Cortex*, *Hippocampus* and *P.B. Ganglia* tissues but the Trans-CLIME estimator achieved much better results due to the aggregation step. Still, the Glasso estimator has the best performance and Trans-CLIME does not perform better than CLIME for the other tissues.

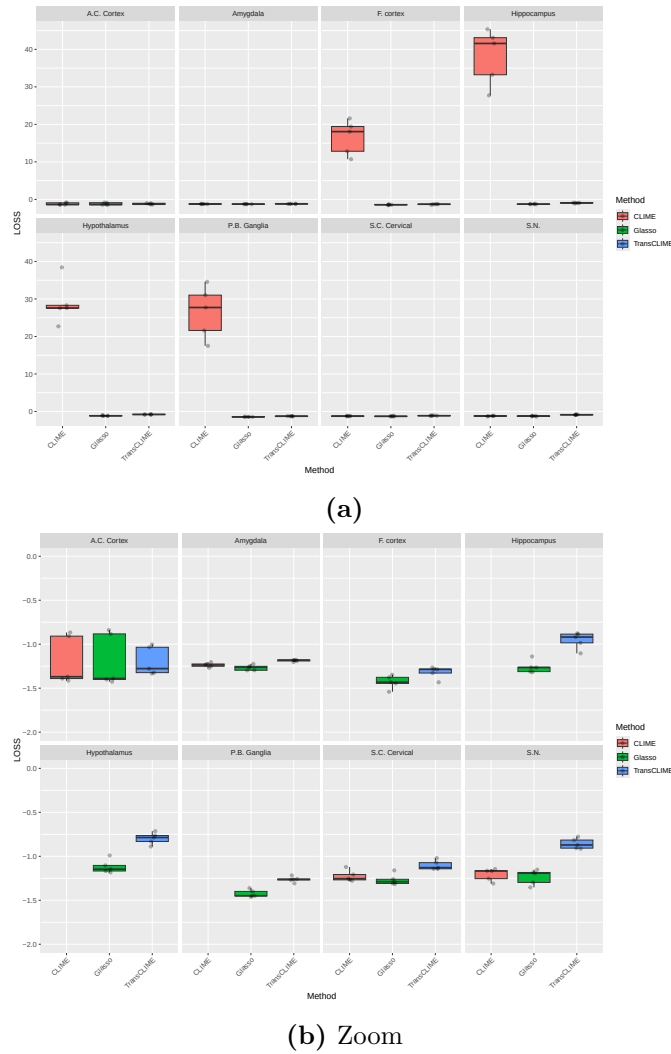


Figure 6.10: Boxplots of the loss results for the three estimators across the different brain tissues, using the same grid of values for the tuning parameters.

The proportion of edges detected with FDR control in the same settings for the three estimators are shown in Figure 6.11. As expected, we obtain the same results for the CLIME estimator with the same tissues and the Glasso still produces sparser estimates. The Trans-CLIME estimator has better results than in the previous settings but still produces dense estimates for some tissues.

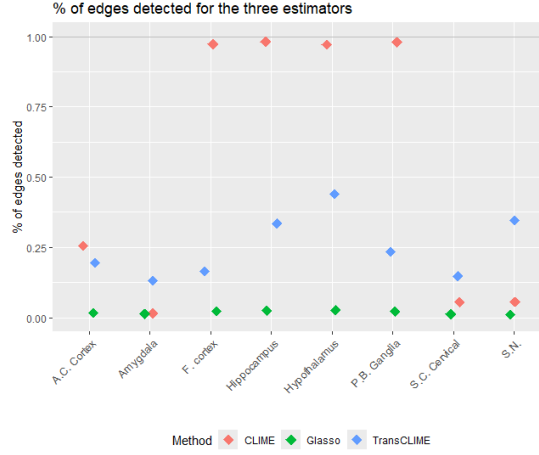


Figure 6.11: Proportion of edges detected for the different estimators and for each brain tissue, using the same grid of values for the tuning parameters. Lower values denote a sparser matrix. Inspired from Li et al. (2022).

Using the same grid of values, the performance of the Trans-CLIME is better but still lower than its two competitors. Moreover, as we do not obtain the regular CLIME estimator but an heuristic version chosen by Li et al. (2022), we cannot ensure the good performance of the CLIME estimator. Nevertheless, these estimations have proven that the aggregation step performed during the estimation of the Trans-CLIME allows to produce stable results, which was not the case for the CLIME estimator.

Eventually, we decided to compare the performance of the three estimators under the same tuning parameter value fixed at $\sqrt{\log p/n}$ to be able to compare them without using the heuristic version of the CLIME estimator. The loss results are displayed in Figure 6.12. In this situation, the Trans-CLIME estimator is the one outperforming its competitors and the CLIME estimator having the worst results with the outcome diverging largely between iterations. Note that the Glasso estimator has been generally very stable in its estimates in each setting.

The proportion of edges detected by FDR are depicted in Figure 6.13 and as seen, the Glasso estimator produces denser matrices than before (because its optimal tuning parameter is far from $\sqrt{\log p/n}$). Even if the Trans-CLIME demonstrates better performance in terms of the loss function, it still produces too dense matrices. The CLIME has better results with sparser estimates.

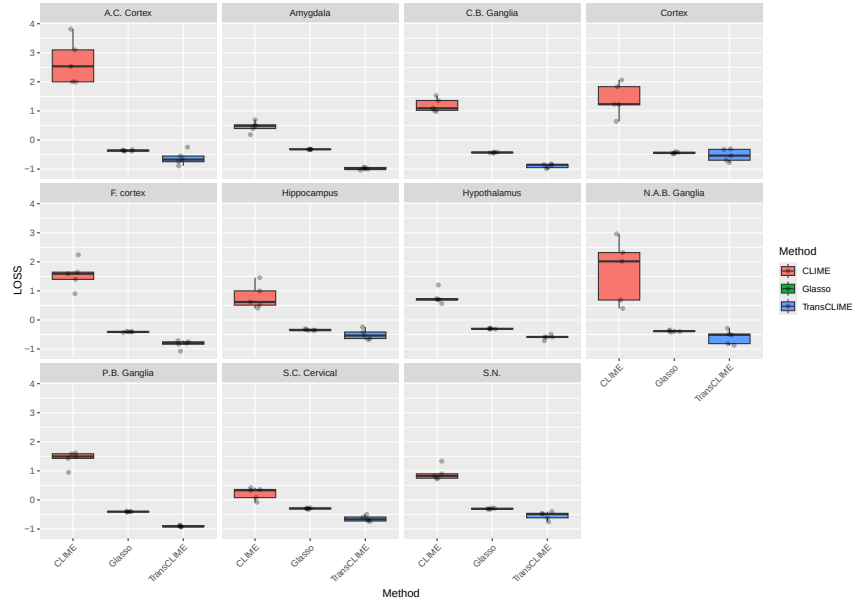


Figure 6.12: Boxplots of the loss results for the different estimators and brain tissues.

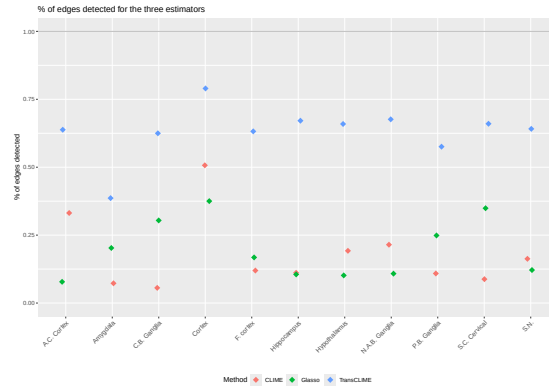


Figure 6.13: Boxplots of the loss results for the different estimators and brain tissues. Inspired from Li et al. (2022).

6.3 Conclusion

In this chapter, we have seen the performance of the three methods to estimate the graph structure of different brain tissues. These results have shown that the Trans-CLIME and CLIME estimators were outperformed by the Glasso estimator in terms of loss function as well as in detecting significant edges in the graph structure, while the Glasso estimator was able to produce much sparser estimates than the CLIME and Trans-CLIME estimators.

The results obtained by Li et al. (2022) are different from our results. Without more information on their methodology, we conjecture that the procedure used here to find the tuning parameters was not optimal and has probably led to bad performance. The attempts to make more comparisons between the different estimators, using the same grid of values for the tuning parameters or using the same specific value $\sqrt{\log p/n}$, did result in better performance in general for the loss outcomes (except for some tissues) but the estimates produced with the FDR control procedure were not as sparse as expected. For future projects, more focus needs to be taken on the choice of the tuning parameters to ensure a suitable comparison between the competitors and ensure the optimal performance for every estimator.

Chapter 7

Conclusion

The objective of this thesis was to integrate the Transfer Learning strategy into Penalized Gaussian Graphical Models, with the use of Trans-CLIME, an efficient estimator combining the information from different studies throughout the use of a divergence matrix summarizing the similarities between auxiliary and target studies.

The simulation results have allowed to assess the performance of this estimator. We have managed to show that the estimator resulted in better results only when all (or in some settings, almost all) studies are informative. This raises a possible issue as this information is not accessible in practice. Therefore this estimator needs to be investigated and benchmarked against other competitors in practice to ensure its full performance, and verifying the usefulness and reasonable similarities of the possible auxiliary studies beforehand will be crucial.

Unfortunately, the performance of the Trans-CLIME to estimate the graph structure of the different brain tissues was not coherent with the simulation results. We have shown that by using a different grid of tuning parameters for the Glasso and CLIME/Trans-CLIME estimators in order to obtain optimal loss results, the Glasso was the most performant with optimal loss results and leading to sparse matrices, while CLIME and Trans-CLIME led to dense, non-interpretable graph structures. However, the results using the same tuning parameter has allowed to show the better performance of the Trans-CLIME with the same settings.

Nevertheless, this thesis has proven the efficiency of using auxiliary studies to improve the performance. By assuming high similarity between the target study and the auxiliary ones, this approach could be used in several domains such as model predictions from genomics to medical applications, where high-dimensional studies are thriving.

During this thesis, many challenges have been highlighted. First, the use of different grids between the different matrix settings was not optimal and may have impacted the results and comparisons between the different settings. Also, the random matrix diverging in each setting have led to more difficulties to compare and interpret the results between the distinct values of the parameters. Eventually, many issues have been encountered during the application procedure, starting with the different grid of tuning parameters used, leading to less-interpretable comparisons. The loss function used was not optimal for finding a suitable value for the tuning parameters. It would be more efficient to use another metric, such as a RMSE metric if a label variable is available. Moreover, we did not obtain the same results as Li et al. (2022). As the information on the genes selected and the settings used were not available in their paper, the source of these differences is still unknown. For future projects, it will be essential to focus mainly on the procedures and the possible problems beforehand.

For future research, one could improve the Transfer Learning method by developing a statistical method to select the informative studies from the set of available studies, as it has been done in regression settings (Li et al., 2021), in multitask learning (Hanneke & Kpotufe, 2020) or in Machine Learning (Wang et al., 2019). This progress could lead to optimal results and improve greatly the Transfer Learning estimation. Additionally, this strategy could be applied to other possible estimators in order to assess the possible gain in performance.

In conclusion, the integration of Transfer Learning into penalized Gaussian Graphical Models is a promising approach that needs to be developed and improved in future works.

Bibliography

- Banerjee, O., Ghaoui, L. E., & d’Aspremont, A. (2008). Model Selection Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data. *Journal of Machine Learning Research*, 9(15), 485–516.
- Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1), 289–300.
- Bhushan, N., Mohnert, F., Sloot, D., Jans, L., Albers, C., & Steg, L. (2019). Using a Gaussian Graphical Model to Explore Relationships Between Items and Variables in Environmental Psychology research. *Frontiers in Psychology*, 10(1050).
- Bunea, F., Tsybakov, A., & Wegkamp, M. (2007). Aggregation for Gaussian regression. *The Annals of Statistics*, 35(4), 1674–1697.
- Cai, T., Liu, W., & Luo, X. (2011). A constrained ℓ_1 Minimization Approach to Sparse Precision Matrix Estimation. *Journal of the American Statistical Association*, 106(494), 594–607.
- Cai, T., Ren, Z., & Zhou, H. (2016). Estimating Structured High-Dimensional Covariance and Precision Matrices: Optimal Rates and Adaptive Estimation. *Electronic Journal of Statistics*, 10(494), 1–59.
- Dai, W., Xue, G., Yang, Q., & Yu, Y. (2007). Transferring Naive Bayes Classifiers for Text Classification. *Proceedings of the National Conference on Artificial Intelligence*, 1, 540–545.
- Dempster, A. (1972). Covariance Selection. *Biometrics*, 28(1), 157–175.
- Drton, M., & Perlman, M. (2004). Model selection for Gaussian concentration graphs. *Biometrika*, 91(3), 591–602.
- Edwards, D. (2000). *Introduction to Graphical Modelling*. Springer.
- Fan, J., & Li, R. (2001). Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *Journal of the American Statistical Association*, 96(456), 1348–1360.

- Ford, W. (2015). *Numerical Linear Algebra with Applications*. Academic Press.
- Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3), 432–441.
- Hanneke, S., & Kpotufe, S. (2020). A No-Free-Lunch Theorem for MultiTask Learning. <https://arxiv.org/abs/2006.15785>
- Janková, J., & van de Geer, S. (2015). Confidence intervals for high-dimensional inverse covariance estimation. *Electronic Journal of Statistics*, 9, 1205–1229.
- Ke Wang, S.-Y. O., Alexander Franks. (2023). Learning Gaussian graphical models with latent confounders. *Journal of Multivariate Analysis*, 198(105213).
- Koller, D., & Friedman, N. (2009). *Probabilistic Graphical Models*. The MIT Press.
- Kullback, S. (1968). Probability Densities with Given Marginals. *The Annals of Mathematical Statistics*, 39(4), 1236–1243.
- Kullback, S. (1997). *Information Theory and Statistics*. Dover Publications.
- Lauritzen, S. L. (1996). *Graphical Models*. Oxford University Press.
- Lecué, G., & Rigollet, P. (2014). Optimal Learning with Q-aggregation. *The Annals of Statistics*, 42(1), 211–224.
- Li, S., Cai, T., & Li, H. (2022). Transfer Learning in Large-Scale Gaussian Graphical Models with False Discovery Rate Control. *Journal of the American Statistical Association*, 118(543), 1–13.
- Li, S., Cai, T. T., & Li, H. (2021). Transfer Learning for High-Dimensional Linear Regression: Prediction, Estimation and Minimax Optimality. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1), 149–173.
- Liu, R., Shi, Y., Ji, C., & Jia, M. (2019). A Survey of Sentiment Analysis Based on Transfer Learning. *IEEE Access*, 7, 85401–85412.
- Liu, S., Trenkler, G., Kollo, T., von Rosen, D., & Baksalary, O. M. (2023). Professor Heinz Neudecker and matrix differential calculus. *Statistical Papers*, 65, 2605–2639.
- Liu, W. (2013). Gaussian Graphical Model Estimation with False Discovery Rate Control. *The Annals of Statistics*, 41, 2948–2978.
- Meinshausen, N., & Bühlmann, P. (2006). High-dimensional graphs with the Lasso. *The Annals of Statistics*, 34, 1436–1462.
- Morid, M., Borjali, A., & Del Fiol, G. (2021). A scoping review of transfer learning research on medical image analysis using ImageNet [art. no. 104115]. *Computers in Biology and Medicine*, 128.

- Pan, S., Kwok, J., Yang, Q., & Pan, J. (2007). Adaptive localization in a dynamic WiFi environment through multi-view learning. *Proceedings of the National Conference on Artificial Intelligence*, 2, 1108–1113.
- Ravikumar, P., Wainwright, M. J., Raskutti, G., & Yu, B. (2011). High-dimensional covariance estimation by minimizing ℓ_1 -penalized log-determinant divergence. *Electronic Journal of Statistics*, 5, 935–980.
- Rosenstein, M. T., Marx, Z., Kaelbling, L. P., & Dietterich, T. G. (2005). To transfer or not to transfer. *Neural Information Processing Systems*. <https://api.semanticscholar.org/CorpusID:597779>
- Speed, T., & Kiiveri, H. (1986). Gaussian Markov Distributions over Finite Graphs. *The Annals of Statistics*, 14(1), 138–150.
- Taboga, M. (2021). *Multivariate normal distribution - Maximum Likelihood Estimation*. <https://www.statlect.com/fundamentals-of-statistics/multivariate-normal-distribution-maximum-likelihood>
- Vandenberghe, L., Boyd, S., & Wu, S.-P. (1998). Determinant Maximization with Linear Matrix Inequality Constraints. *SIAM Journal on Matrix Analysis and Applications*, 19, 499–533.
- Vershynin, R. (2018). Concentration of Sums of Independent Random Variables. In *High-dimensional Probability: An Introduction with Applications in Data Science* (pp. 11–37). Cambridge University Press.
- Wang, Z., Dai, Z., Póczos, B., & Carbonell, J. (2019). Characterizing and avoiding negative transfer. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11285–11294.
- Yuan, M., & Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1), 19–35.
- Yuan, X. (2012). Alternating direction methods for sparse covariance selection. *Journal of Scientific Computing*, 51, 261–273.

Appendix A

Appendix

A.1 Karush Kuhn Tucker conditions

The Karush-Kuhn-Tucker (KKT) conditions have been developed to solve constrained optimization problems. Given the following problem

$$\begin{aligned} & \min_{x \in \mathbb{R}} f(x) \\ \text{subject to : } & h(x) = 0 \\ & g(x) \leq 0 \end{aligned} \tag{A.1.1}$$

where $f(x) : \mathbb{R}^p \rightarrow \mathbb{R}$ is the objective function, $h(x) : \mathbb{R}^p \rightarrow \mathbb{R}^m$ are the equality constraints and $g(x) : \mathbb{R}^p \rightarrow \mathbb{R}^l$ the inequality constraints. Any points x respecting these constraints is said feasible. Then any local minimum of the optimization problem, denoted by x^* , respects the following KKT conditions

$$\begin{cases} \nabla f(x^*) + \lambda^{*t} \nabla h(x^*) + \mu^{*t} \nabla g(x^*) = 0 \\ \mu_i^* g_i(x^*) = 0, \quad j = 1, 2, \dots, l \\ \mu^* \geq 0, \end{cases} \tag{A.1.2}$$

where λ^* and μ^* are Lagrange multipliers.

For example define the following optimization problem

$$\begin{aligned} & \min_{x \in \mathbb{R}} f(x) = 2x_1^2 + 4x_2^2 \\ \text{subject to : } & x_1 + x_2 = 4 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{aligned} \tag{A.1.3}$$

We can express it using $h(x)$ and $g(x)$ as follows

$$\begin{aligned} \min_{x \in \mathbb{R}} f(x) &= 2x_1^2 + 4x_2^2 \\ \text{subject to : } h(x) &= x_1 + x_2 - 4 = 0 \\ g_1(x) &= -x_1 \leq 0 \\ g_2(x) &= -x_2 \leq 0, \end{aligned} \tag{A.1.4}$$

and the gradients of these functions are

$$\begin{aligned} \nabla f(x) &= (4x_1 \ 8x_2)^t \\ \nabla h(x) &= (1 \ 1)^t \\ \nabla g(x) &= (-1 \ -1)^t. \end{aligned}$$

The KKT conditions for this problem are

$$\begin{cases} 4x_1 + \lambda^* - \mu_1^* = 0 \\ 8x_2 + \lambda^* - \mu_2^* = 0 \\ \mu_1^* x_1 = 0 \\ \mu_2^* x_2 = 0 \\ \mu^* \geq 0, \end{cases} \tag{A.1.5}$$

Then if we find a minimizer x^* respecting the KKT conditions, this is a stationary point. For example, $x^* = (8/3, 4/3)^t$ respects the KKT conditions and is a stationary point (in reality the only minimum and so the global minimizer of the problem (A.1.3)).

