# Graphical Modelling for High-Dimensional Data

# Contents

# 1 Introduction

Graphs play a vital role in representing relationships between random variables, with plenty of real-world applications. For example, they are used to represent the structure of a protein in medical science; to model relationships between financial assets; and machine learning efforts often use them to infer causality and accurately predict future outcomes.

In this essay we will study undirected graphs, where an edge represents an immediate dependence between the corresponding random variables. Directed graphs are another line of research, and are intimately linked to the problem of causality. Our focus will be the estimation of the Conditional Independence Graph (CIG), where the absence of an edge between two nodes indicates that the random variables are **conditionally independent**, given the others as known.

It has long been known that for Gaussian graphical models, conditional independence between two variables $X_j$ and $X_k$ occurs precisely when there is a zero entry in the precision matrix, $\Omega$. The low-dimensional problem, where the number of variables is small, is often easily solvable by standard techniques to invert the covariance matrix and estimate $\Omega$ directly.

However, frequently in modern statistics we encounter situations where the number of variables is **huge**, leading to problems in computational feasibility. This motivates a set of desirable qualities we look for in a graph estimation procedure:

- Computationally feasible
- Accurate
- Flexible

The second of these should be fairly self-explanatory; we would like our estimates to accurately represent the true underlying structure. Flexibility is an interesting problem which many recent proposals attempt to address; traditional CIG estimation procedures have been designed under the assumption of **Gaussian data**. If the data we receive does not truly follow a multivariate Gaussian distribution, then our estimators and consistency results become invalid.

Figure 1 gives an example of observed data not necessarily being multivariate normal. It plots solar radiation levels against ozone levels in New York, and so we are looking at $p = 2$ random variables. Note that this data does not possess the elliptic shape that we would expect from Gaussian data, as seen in the right hand plot of simulated bivariate normal data.

Authors recently have been proposing ways of allowing for non-Gaussian data and hence more and more flexible techniques have been developed, for example Liu et al. (2012)

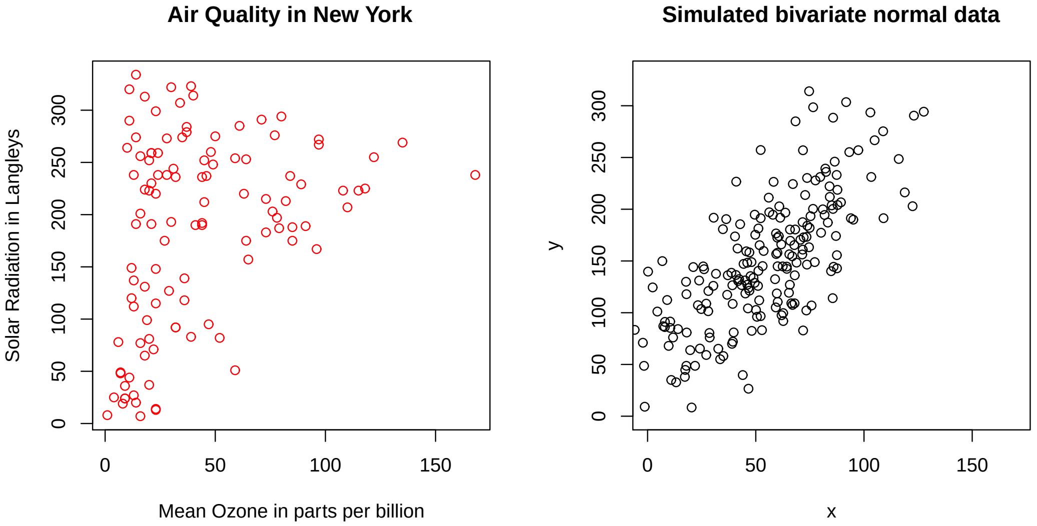

Figure 1: A plot of the amount of solar radiation against ozone levels (both air quality indicators) in New York in 1973. Next to it is a plot of simulated bivariate normal data with correlation $\rho = 0.75$. Here, the number of variables is $p = 2$.

extended the allowed distributions to a **nonparanormal family**. Here we will study how these new techniques work, and how they compare with each other both theoretically and empirically.

An outline for the essay is as follows: in Section 2 we will summarise the two key early approaches to understand undirected graph estimation, being Neighbourhood Selection and the Graphical Lasso. Then in each subsequent section we will discuss a different recent addition to these frameworks, from CLIME (2011) up until MQGM (2016). We will aim to cover how they relate to the early approaches as well as each other, which desirable qualities they possess and also where there is room for improvement. In Section 8 we will look at how each procedure compares numerically in a variety of situations, before concluding the essay.

In the Appendix there are details which have been omitted, and offer extra insight into a few of the procedures. However, it should be noted that emphasis throughout this essay is on the methods and how they compare rather than the proofs and their details.

# 2 Notation and background

In this section we will set up the notation we will be using throughout this essay. Following this will be a brief overview of two key procedures: Neighbourhood Selection and the Graphical Lasso.

## 2.1 Notation

In the approaches that follow, we consider the problem of estimating high-dimensional undirected graphical models. A graph $G = (V, E)$ is a collection of vertices $V = \{1, \dots, p\}$ along with an edge set $E \subseteq V \times V$. In this essay we only look at **undirected graphs**, that is $(i, j) \in E$ if and only if $(j, i) \in E$. We assume the availability of $n$ observations from a $p$-dimensional random vector $X = (X_1, \dots, X_p)^T$, which are stored as rows in the $n \times p$ matrix $\mathbf{X}$.

**Definition 1** *We say that the distribution of $X$ is* pairwise Markov *with respect to $G$ if, writing $X_{-ij} = \{X_k : k \neq i, j\}$, there is an absence of an edge (i.e. $(i, j) \notin E$) if and only if $X_i$ is conditionally independent of $X_j$ given all of the other variables, written*

$$X_i \perp X_j \mid X_{-ij}.$$

We use $G$ to represent relationships between the random variables $X_1, \dots, X_p$ by assuming the distribution of $X$ is pairwise Markov with respect to $G$.

In the case where $X \sim N_p(\mu, \Sigma)$ has a multivariate normal distribution, it has long been known that the problem of identifying conditional independences is reduced to finding zeroes of the covariance matrix $\Omega := \Sigma^{-1}$. That is,

$$X_i \perp X_j \mid X_{-ij} \iff \Omega_{ij} = 0. \tag{1}$$

In the low-dimensional setting $p < n$, the covariance selection technique proposed by Dempster (1972) directly finds an inverse covariance matrix estimator $\hat{\Omega}$ by inverting the maximum likelihood estimator (MLE) $\hat{\Sigma}$. However, in the high dimensional setting $p \gg n$, this approach breaks down. For a start, the MLE may not exist when $p > n$, and even if it did this procedure becomes computationally expensive as $p$ increases.

We now introduce two important early approaches which aim to tackle this high-dimensional problem. They are Neighbourhood Selection and the Graphical Lasso. We will often refer to these as NS and Glasso.

## 2.2 Neighbourhood Selection

Meinhausen and Bühlmann (2006) introduced Neighbourhood Selection as a way of estimating a sparse graph $G$. They allow the number of nodes $p = |V|$ to grow as a function of the number of observations $n$, but not exponentially fast.

**Definition 2** *The* neighbourhood $n(i)$ *of node* $i \in V$ *as the the smallest subset of* $V \setminus i$ *such that*

$$X_i \perp \{X_k : k \neq i,\ k \in V \setminus n(i)\} \mid X_{n(i)}.$$

However, it is more motivational for the NS approach to view the neighbourhood of $i$ in the following way.

Consider a linear estimator of $X_i$ as a function of $X_{-i}$, namely $\hat{X}_i = \sum_{j \neq i} \beta_j X_j$. Write $\beta^i = (\beta^i_1, \ldots, \beta^i_p)^T$ (with $\beta^i_i = 0$) as the vector of coefficients which minimise the mean squared error $\mathbb{E}(X_i - \hat{X}_i)^2$. It can be shown that these optimal coefficients are directly related to the inverse covariance matrix via $\beta^i_j = -\Omega_{ij}/\Omega_{ii}$. We can therefore interpret the neighbourhood of node $i$ as **precisely** the set of nodes $j$ for which the coefficient $\beta^i_j \neq 0$.

In a nutshell, Neighbourhood Selection regresses $X_i$ on $X_{-i}$ for each variable $i = 1, \ldots, p$. It uses the Lasso penalty, in order to perform simultaneous variable selection and estimation on the linear coefficients. It therefore forms an estimate for the optimal **linear** model and hence an estimate for the neighbourhood.

That is, for $i = 1, \ldots p$, estimate $\beta^i$ by its Lasso estimate $\hat{\beta}^{i,\lambda}$ defined as

$$\hat{\beta}^{i,\lambda} = \underset{\beta:\beta_i=0}{\operatorname{argmin}} \Big\{ n^{-1} ||\mathbf{X}_i - \mathbf{X}\beta||_2^2 + \lambda ||\beta||_1 \Big\} \tag{2}$$

where $\mathbf{X}_i$ is the $i^{\text{th}}$ column of the $n \times p$ matrix $\mathbf{X}$ of observations. Our neighbourhood estimate is the corresponding set of non-zero coefficients

$$\hat{n}(i)^\lambda = \{j \in V : \hat{\beta}^{i,\lambda}_j \neq 0\}. \tag{3}$$

Now that we have neighbourhood estimates for each $i \in V$, we would like to see if, and with what speed, they converge to the true neighbourhood. In other words, *are these estimates consistent?* Under a set of assumptions, including sparsity, Meinhausen and Bühlmann proved that the probability of falsely estimating the neighbourhood converges to 0 **exponentially fast**.

Finally, one estimates the edge set $E$ by $\hat{E}$ where $(i, j) \in \hat{E}$ if and only if $i \in \hat{n}(j)^\lambda$ and/or $j \in \hat{n}(i)^\lambda$ (both ways of choosing $\hat{E}$ can be shown to be consistent). This is the final NS estimate of the CIG.

### 2.3 Graphical Lasso

We will now summarise another key early development in high-dimensional graph estimation, called the Graphical Lasso (Friedman et al, 2007). It aims to find the MLE for $\Omega$ directly and deduce the structural zeroes $\Omega_{ij} = 0$. It also provides a nice conceptual link between itself and Neighbourhood Selection, being that NS is an approximation to the true maximum likelihood problem.

After partially maximising the log-likelihood for $X \sim N_p(\mu, \Sigma)$ with respect to $\mu$, one can obtain the following expression as a function of the inverse covariance matrix $\Omega$:

$$\ell(\Omega) = \frac{n}{2}\{\log\det(\Omega) - \mathrm{tr}(S\Omega)\},$$

where we have written $S$ to be the empirical covariance matrix. The Graphical Lasso aims to maximise this (or minimise $-\ell(\Omega)$) over $\Omega \succ 0$ using a Lasso penalty on the elements of $\Omega$. Dropping the constant factor $\frac{n}{2}$, the problem therefore becomes

$$\min_{\Omega:\Omega\succ 0}\{-\log\det(\Omega) + \mathrm{tr}(S\Omega) + \lambda||\Omega||_1\}. \tag{4}$$

First, we transform this problem and target each column individually. Let $W$ denote our estimate of $\Sigma$, then partition into blockwise form:

$$\begin{pmatrix} W_{11} & w_{12} \\ w_{12}^T & w_{22} \end{pmatrix},$$

where $w_{12}$ is a $(p-1)$ column vector, and we treat the final column as the target for coordinate descent. One can verify that $w_{12}$ satisfies the Karush–Kuhn–Tucker (KKT) conditions for (4) if and only if $\beta = W_{11}^{-1}w_{12}$ satisfies the KKT conditions of

$$\min_{\beta\in\mathbb{R}^{p-1}}\{\tfrac{1}{2}\|W_{11}^{1/2}\beta - b\|_2^2 + \lambda\|\beta\|_1\}, \tag{5}$$

where we have defined $b = W_{11}^{-1/2}s_{12}$. Thus the problem for $w_{12}$ is equivalent to solving (5) for $\beta$.

The KKT conditions for (4) also directly imply that the optimal estimate for $w_{ii}$ is $w_{ii} = s_{ii} + \lambda$ for each $i$. We then solve multiple Lasso problems, and cycle over target columns repeatedly until convergence. This is a **coordinate descent procedure**, since we are updating the $W$ estimate at each stage of the iteration.

As explained in Friedman et al (2007), after obtaining our estimate $W$ we can recover $\hat{\Omega} = W^{-1}$ (and therefore estimate the edge structure of the desired graph) relatively easily once the algorithm above has been completed.

Rothman et al. (2008) proved the following convergence rate in the matrix operator norm.

**Theorem 1** *Under a set of assumptions, the minimiser $\hat{\Omega}$ to (4) converges to the true $\Omega$ as $n \to \infty$ at rate*

$$\|\hat{\Omega} - \Omega\|_2 = O_P\Big(\sqrt{\tfrac{(s+1)\log p}{n}}\Big). \tag{6}$$

*where $s$ is the sparsity parameter.*

The point of this approach has already been touched on above, where we mention that problem (5) is indeed a **Lasso regression problem**, which we already know how to solve using soft-threshold operators. Neighbourhood Selection also uses a slight variant of this, and Friedman et al (2007) explain how they differ from the true maximum likelihood problem.

Suppose we replaced $W_{11}$ with $S_{11}$ in (5). Then the solutions $\hat{\beta}$ would be the Lasso estimates obtained when regressing the $p^{\text{th}}$ variables on all of the others, and hence would recover the Neighbourhood Selection estimates in Meinhausen and Bühlmann (2006). However, $W_{11} \neq S_{11}$ in general, and so the NS approach does not yield the MLE, instead it provides a simpler (and faster) approximation.

In summary, Friedman et al (2007) have shown how the maximum likelihood problem (4) can be reduced to $p$ **coupled Lasso problems** (5), which share a common $W = \Omega^{-1}$. They then describe an efficient algorithm to estimate $\Omega$ and therefore the structural zeroes in the underlying graph.

### 2.4 Discussion

We have now summarised the two key procedures that we will refer to throughout this essay. It is convenient to interpret recent developments in terms of extending or adjusting the two frameworks introduced by NS and Glasso in the sense that, broadly speaking:

1. the NS framework attempts to model the conditional distributions of $(X_1, \ldots, X_p)$, and
2. the Glasso framework aims to estimate the structural zeroes of $\Omega_{ij}$.

As we saw in Friedman et al, the two approaches are linked by an approximation.

It should be noted here that **both** of these procedures fall under the Gaussian data assumption; indeed with the Glasso we are using the Gaussian log-likelihood from the start, and NS also implicitly uses this assumption.

Indeed, any model where the regression of $X_i$ on $X_{-i}$ is linear, and the distribution of $X_i|X_{-i}$ depends on $X_{-i}$ only through its mean will (along with some mild assumptions) **necessarily** require $X$ to be distributed multivariate-normally. For a proof of this, see Khatri & Rao (1976) and see Note 6 of Theorem 2. Many of the recent advances we will discuss are designed with the intention of handling non-Gaussian data as well as Gaussian data.

Another drawback that both procedures share, is: *how do we choose an appropriate tuning parameter, $\lambda$?* This can be computationally expensive, and most common approaches are not ideal. The topic of tuning parameter selection will be discussed further in Section 7. For now, however, we look at approaches which extend the NS and Glasso frameworks without worrying too much about choosing $\lambda$; starting with CLIME.

# 3 CLIME

The first recent approach we will look at in this essay is CLIME (constrained $\ell_1$-minimisation for inverse matrix estimation), which shares many similarities with the Graphical Lasso but it computationally much quicker to implement. Working in the same setting: we would like to estimate $\Omega$ from independent samples $X_1, \dots X_n \sim N_p(\mu, \Sigma)$.

## 3.1 A modification to the Graphical Lasso

Recall the Glasso solves

$$\hat{\Omega}_{\text{Glasso}} = \underset{\Omega:\Omega\succ 0}{\arg\min}\{-\log\det(\Omega) + \text{tr}(S\Omega) + \lambda\|\Omega\|_1\}.$$

By the KKT conditions, we deduce that a solution $\hat{\Omega}_{\text{Glasso}}$ must satisfy

$$\hat{\Omega}_{\text{Glasso}}^{-1} - S = \lambda\hat{\nu} \quad \text{where} \quad \hat{\nu} \in \partial\|\hat{\Omega}_{\text{Glasso}}\|_1.$$

Using the expression for elements in the differential $\partial\|\cdot\|_1$, this leads us to the optimisation problem:

$$\text{minimise } \|\Omega\|_1 \text{ subject to } \|\Omega^{-1} - S\|_\infty \leq \lambda. \tag{7}$$

CLIME can be viewed here as a **modification** to the Glasso optimisation problem (7), which adjusts the constraint to something more easily solvable. Let $\tilde{\Omega}$ be a solution to the following optimisation problem:

$$\text{minimise } \|\Omega\|_1 \text{ subject to } \|S\Omega - I_p\|_\infty \leq \lambda. \tag{8}$$

The final CLIME estimator $\hat{\Omega}$ is then given by symmetrising the solutions $\tilde{\Omega} = (\tilde{\omega}_{ij})$ to (8) using $\hat{\omega}_{ij} = \min(\tilde{\omega}_{ij}, \tilde{\omega}_{ji})$. Cai, Liu and Luo (2011) show that $\hat{\Omega}$ is positive definite with high probability.

### Example

Following [3], suppose we want to compare the constraint sets for $\Omega$ as above. Work in 2 dimensions with

$$\Omega = \begin{pmatrix} x & y \\ y & x \end{pmatrix} \quad \text{and} \quad S = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

(where we set $\Omega$ to be in this form for simplicity), set $\lambda = 0.4$. After some short matrix algebra, the Glasso and CLIME constraints as in (6) and (7) then become

$$\text{Glasso:} \quad \max\left\{\left|\tfrac{x}{x^2-y^2} - 1\right|,\ \left|\tfrac{y}{x^2-y^2} - 0.5\right|\right\} \leq 0.4$$

$$\text{CLIME:} \quad \max\left\{\left|x + 0.5y - 1\right|,\ \left|y + 0.5x\right|\right\} \leq 0.4.$$

Figure 2 shows graphically how these sets differ; CLIME's constraint set is a simpler approximation to the more complicated Glasso (the true log-likelihood constraint set). Notice that the CLIME constraint will **always** be a linear set, since it only ever uses matrix multiplication. We can extend this into more dimensions by allowing more degrees of freedom in the $\Omega$.

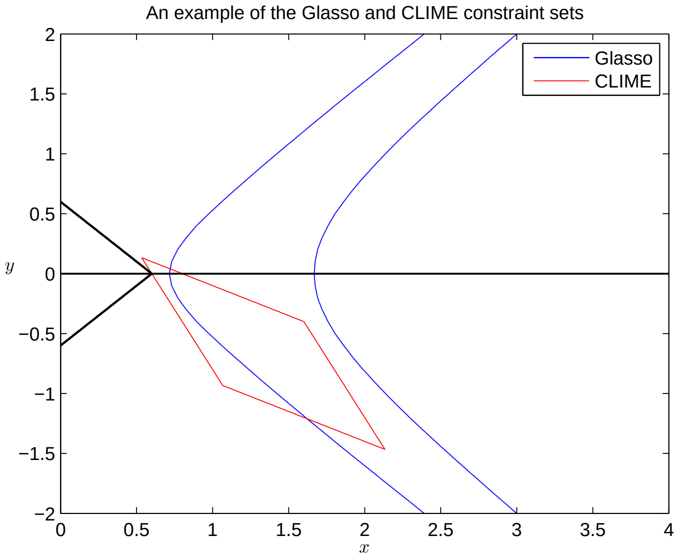

Figure 2: A MATLAB plot of the two constraint sets for our example. By drawing contours of $|x| + |y|$ (the objective), one can see that Glasso and CLIME will predict similar values of $\hat{\Omega}$.

## 3.2 Computation

Let $\hat{\beta}_i$ be the solution to

$$\text{minimise } \|\beta\|_1 \text{ subject to } \|S\beta - e_i\|_\infty \leq \lambda \tag{9}$$

where $e_i$ is the $i^{\text{th}}$ standard basis vector, $i = 1, \dots p$. Then we can write solutions $\tilde{\Omega}$ to (8) as $\tilde{\Omega} = (\hat{\beta}_1, \dots, \hat{\beta}_p)$ with columns solving (9). This reduces computation significantly, and all we have to do to implement the CLIME estimator is perform $p$ optimisation problems given by (9). We call this a **linear program**, more detail on this can be found in the Appendix.

Cai, Liu and Luo showed that the CLIME estimator $\hat{\Omega}$ does indeed converge to the true precision matrix and we can thus infer conditional independence relationships. They proved the following convergence rates.

**Theorem 2** *Under certain assumptions, the CLIME estimator $\hat{\Omega}$ converges to the true $\Omega$ under the following rates, in matrix operator norm and infinity norm*

$$\|\hat{\Omega} - \Omega\|_2 = O_P\left(s\sqrt{\tfrac{\log p}{n}}\right)$$

$$\|\hat{\Omega} - \Omega\|_\infty = O_P\left(\sqrt{\tfrac{\log p}{n}}\right)$$

*where $s$ denotes the sparsity; the maximum number of non-zero entries on each row.*

Comparing this with the Glasso convergence rate for the $\| \cdot \|_2$ norm in (6), we see that they only differ significantly when $s$ is large. Assuming the graphs are sparse, we deduce that CLIME and Glasso have very similar convergence rates.

We have seen that CLIME can be a useful approximation to the Glasso by adjusting its constraint set, allowing for quicker computation. However, it fails to tackle the non-Gaussian problem as discussed earlier. We now move on to a technique which does admit non-Gaussian data.

# 4 Nonparanormal SKEPTIC

We now introduce the first of three techniques studied in this paper which aim to tackle the problem of non-Gaussianity. Liu et al (2012) introduce the nonparanormal SKEPTIC (Spearman/Kendall estimates preempt transformations to infer correlation) which relaxes the Gaussian assumption to a family distributions called the **nonparanormal family**.

**Definition 3** *Given an ordered set $f = \{f_1, \dots, f_p\}$ of monotone univariate functions and a positive definite covariance matrix $\Sigma \in \mathbb{R}^{p \times p}$ with $\operatorname{diag}(\Sigma) = \mathbf{1}$, we say that $X$ has a nonparanormal distribution $X \sim NPN_p(f, \Sigma)$ if*

$$f(X) := (f_1(X_1), \dots, f_p(X_p))^T \sim N_p(0, \Sigma).$$

The key to the nonparanormal SKEPTIC is its exploitation of Spearman's rho and Kendall's tau estimators using a neat property (see Lemma 1). This approach first estimates $\Sigma$, before plugging the estimate into existing techniques, such as CLIME, to find the pattern of zeroes in the precision matrix.

Liu et al. (2012) modelled this approach after their earlier paper (2009) and showed that the resulting graph recovery is consistent with optimal (same as parametric techniques) computation time. There are techniques for estimating $\{f_1, \dots, f_p\}$ and hence obtaining the distribution of $X$, but they are not necessary for graph recovery purposes.

## 4.1 Spearman's rho and Kendall's tau

Spearman's rho and Kendall's tau are two nonparametric measures of correlation between the rankings of random variables. Here, ranking means the ordinal number relating the size of a data point amongst others (so that the $k^{\text{th}}$ smallest data point has rank $k$). The population versions of these correlation coefficients are given by

$$\rho_{jk} := \text{Corr}(F_j(X_j), F_k(X_k))$$

$$\tau_{jk} := \ \text{Corr}(\text{sgn}(X_j - \tilde{X}_j), \text{sgn}(X_k - \tilde{X}_k)$$

respectively, where $F_j$ is the CDF of $X_j$ and $(\tilde{X}_j, \tilde{X}_k)$ are independent copies of $(X_j, X_k)$. We now define estimators for $\rho$ and $\tau$.

Recall, we have a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ of observations, with entries $x_j^{(i)}$, where $i = 1, \dots, n$ denotes the $i^{\text{th}}$ observation, and $j = 1, \dots, p$ denotes the $j^{\text{th}}$ random variable. Let $r_j^{(i)}$ be the **rank** of $x_j^{(i)}$ among $x_j^{(1)}, \dots, x_j^{(n)}$, so that the smallest of these observations ($x_j^{(k)}$, say) will have rank $r_j^{(k)} = 1$, and the largest will have rank $n$. Also write $\bar{r}_j = \frac{1}{n} \sum_{i=1}^n r_j^{(i)}$ (and hence equals $\frac{n+1}{2}$).

**Definition 4** *We define the following statistics for empirical realisations of $X_j$ and $X_k$:*

$$\hat{\rho}_{jk} = \frac{\sum_{i=1}^n (r_j^{(i)} - \bar{r}_j)(r_k^{(i)} - \bar{r}_k)}{\sqrt{\sum_{i=1}^n (r_j^{(i)} - \bar{r}_j)^2 \cdot \sum_{i=1}^n (r_k^{(i)} - \bar{r}_k)^2}} \tag{10}$$

$$\hat{\tau}_{jk} = \frac{2}{n(n-1)} \sum_{1 \le i < i' \le n} sgn\left( (x_j^{(i)} - x_j^{(i')})(x_k^{(i)} - x_k^{(i')}) \right). \tag{11}$$

These are estimators of Spearman's rho and Kendall's tau respectively.

Underpinning the whole of the nonparanormal SKEPTIC is the following lemma, which neatly connects these rank correlation coefficients to the covariance matrix $\Sigma$.

**Lemma 1** *Assume $X$ follows a nonparanormal distribution $X \sim NPN_p(f, \Sigma)$. Then*

$$\Sigma_{jk} = 2\sin\left(\frac{\pi}{6}\rho_{jk}\right) = \sin\left(\frac{\pi}{2}\tau_{jk}\right).$$

*A proof for this can be found in [12],[13].*

This motivates very naturally the following estimators for $\Sigma_{jk}$:

$$\hat{S}^{\rho}_{jk} = \begin{cases} 2\sin\left(\frac{\pi}{6}\hat{\rho}_{jk}\right), & \text{if } j \neq k \\ 1, & \text{if } j = k \end{cases} \quad \text{and} \quad \hat{S}^{\tau}_{jk} = \begin{cases} \sin\left(\frac{\pi}{2}\tau_{jk}\right), & \text{if } j \neq k \\ 1, & \text{if } j = k. \end{cases} \tag{12}$$

It is important to note that rank-based estimators such as those above **remain unchanged** under monotonic functions $(f_1, \ldots f_p)$, so we can apply Gaussian methods to our covariance matrix estimate.

## 4.2 Graph recovery

Now that we have an estimate for $\Sigma$, we simply insert it into existing procedures. However, we cannot guarantee that $\hat{S}^{\rho,\tau}$ are positive semidefinite. Certain algorithms may fail with indefinite or non-positive semidefinite inputs, such as NS and Glasso. Liu et al. recommend the use of the graphical Dantzig selector or CLIME since they can be cast as linear programs, and therefore do not require the input $\hat{S}$ to be positive semidefinite.

Here, we focus on the case of plugging $\hat{S}^{\rho,\tau}$ into the CLIME procedure. The only extra computational cost is in computing the $p(p-1)/2$ pairs of $\hat{\rho}_{jk}$ or $\hat{\tau}_{jk}$. Both of these can be computed in $O(n \log n)$ time (see [14]), so we therefore deduce that the additional complexity is $O(p^2 n \log n)$.

However, how do we know that using $\hat{S}^{\rho}$ or $\hat{S}^{\tau}$ instead of the usual empirical covariance matrix $S$ will still result in convergence for CLIME (and other techniques)? The following **concentration properties** guarantee that this is still the case.

**Theorem 3** *For any $n \geq \frac{21}{\log p} + 2$, with probability at least $1 - 1/p^2$,*

$$\sup_{j,k} \left|\hat{S}^{\rho}_{jk} - \Sigma_{jk}\right| \leq 8\pi\sqrt{\frac{\log p}{n}} \tag{13}$$

*Similarly, for any $n > 1$, with probability at least $1 - 1/p$,*

$$\sup_{j,k} \left|\hat{S}^{\tau}_{jk} - \Sigma_{jk}\right| \leq 2.45\pi\sqrt{\frac{\log p}{n}}. \tag{14}$$

When one observes the analysis in proving consistency of CLIME, the **only** property of $S$ that is needed is that $\exists C_0$ such that $\mathbb{P}\big(\|\hat{S}^{\rho,\tau} - \Sigma\|_\infty \leq C_0\sqrt{\frac{\log p}{n}}\big) \leq 1 - \frac{1}{p}$. We can therefore use equations (13) and (14) as sufficient for the analysis in Cai, Liu and Luo (2011). Hence the nonparanormal SKEPTIC converges, and with the same rates as the parametric CLIME.

We therefore deduce the extra flexibility of the nonparanormal SKEPTIC comes at very little extra cost; that of computing $\hat{S}$. This new estimator still converges at the same parametric rate, except now one can model **many** more distributions using the nonparanormal family rather than being limited by the Gaussian family.

Liu et al. identify the issue of semidefiniteness when attempting to use this procedure with NS, however there is a way around it with Glasso/CLIME. Because of this fact, this essay will view the nonparanormal SKEPTIC as part of the Glasso framework family rather than NS. In their paper they show that both the $\hat{S}^\rho$ and $\hat{S}^\tau$ approaches give similar results.

# 5 Joint Additive Models

In Section 2.4 we saw that modelling the distribution of $X_i$ given $X_{-i}$ as linear (such as in Neighbourhood Selection), is implicitly using the assumption that $X \sim N_p(\mu, \Sigma)$. Voorman et al. (2014) proposed an approach which models the conditional means additively, allowing us to apply graph estimation to more general distributions.

## 5.1 An extension of Neighbourhood Selection

In Neighbourhood Selection we are applying a linear model:

$$X_i|X_{-i} = \sum_{k\neq i} \beta_{ik}X_k + \epsilon_i \tag{15}$$

where $\beta_{ik}$ are constants and $\epsilon_i$ is the mean-zero error term. The Sparse Conditional Estimation with Joint Additive Models (Spacejam) approach is essentially extending this into non-linearity:

$$X_i|X_{-i} = \sum_{k\neq i} f_{ik}(X_k) + \epsilon_i \tag{16}$$

where $f_{ik} \in \mathcal{F}$, and $\mathcal{F}$ is some space of functions. We wish to estimate the conditional means $\mathbb{E}_{X_i}(X_i|X_{-i}) = \sum_{k\neq i} f_{ik}(X_k)$.

Recall that we are given $n$ data points stored in the matrix $\mathbf{X}$ with $(\mathbf{X})_{ij} = x_{ij}$, and let $x_k$ denote the $k^{\text{th}}$ column, i.e the $n-$vector of observed data associated with $X_k$. Write

$f_{ik}(x_k)$ as the $n-$vector with $j^{\text{th}}$ entry $f_{ik}(x_{jk})$.

Voorman et al. apply a **Group Lasso penalty** to encourage sparsity and ensure that estimates of $f_{ik}$ and $f_{ki}$ are simultaneously zero or non-zero. This results in the following problem:

$$\underset{f_{ik}\in\mathcal{F},1\leq i,k\leq p}{\text{minimise}}\left[\frac{1}{2n}\sum_{i=1}^{p}\|x_i-\sum_{k\neq i}f_{ik}(x_k)\|_2^2+\lambda\sum_{k>i}\{\|f_{ik}(x_k)\|_2^2+\|f_{ki}(x_i)\|_2^2\}^{1/2}\right]. \tag{17}$$

We choose a set of basis functions, $\{\psi_1(\cdot),\ldots,\psi_r(\cdot)\}$, with respect to which we estimate $f_{ik}(\cdot)$. Then we consider some linear combination $f_{ik}(x_k)=\Psi_{ik}\beta_{ik}$ where $\beta_{ik}\in\mathbb{R}^r$ is a vector of coefficients and $\Psi_{ik}\in\mathbb{R}^{n\times r}$ is the matrix whose $j^{\text{th}}$ column is $\psi_j(x_k)$.

### 5.2 Example

Suppose we want to use a single linear basis function: $\psi_1(x_k)=x_k$.

Then $f_{ik}(x_k)=\beta_{ik}x_k$ and objective of the $i^{\text{th}}$ optimisation problem becomes

$$\frac{1}{2n}\|x_i-\sum_{k\neq i}\beta_{ik}x_k\|_2^2$$

for $i=1,\ldots,p$. Thus for a single linear basis function, we recover the Neighbourhood Selection technique; linear regression on the other variables with a Lasso penalty.

Voorman et al. go on to propose a coordinate descent algorithm to perform this regression given an appropriate set of $r$ basis functions. The algorithm requires an $r\times r$ matrix inversion (requiring $O(r^3)$ operations), for every pair $(j,k)\in V\times V$.

To use this procedure one must choose an appropriate set of basis functions; there will be a trade-off between **accuracy** (more basis functions means a wider span of functions we can use to approximate) and **practicality**, since choosing very large $r$ will be computationally expensive. Motivated by the results of Voorman et al (2014) for different bases, we will use 3 basis functions $\psi_s(x_k)=(x_k)^s$, $s=1,2,3$ for our numerical experiments.

# 6 MQGM

The Multiple Quantile Graphical Model, or MQGM (2016), is an extension of NS. Instead of modelling conditional means, it models a set of **conditional quantiles** and can handle situations where the data is non-Gaussian very effectively.

### 6.1 Modelling quantiles

Firstly, we will define what we mean by a quantile and what it means to perform a quantile regression.[1]

**Definition 5** *The $\alpha$-quantile of a random variable $X$ is*

$$Q_X(\alpha) = \inf\{t : \mathbb{P}(X \leq t) \geq \alpha\}$$

Motivated by Neighbourhood Selection, we attempt to estimate the conditional $\alpha$-quantile of $X_k|X_{-k}$, for each $k = 1 \ldots, p$. This is done via $\alpha$-quantile regression based on $n$ observations, which solves

$$\underset{\theta}{\text{minimise}} \sum_{i=1}^{n} \psi_\alpha \Big( x_k^{(i)} - \sum_{j \neq k} \theta_j x_j^{(i)} \Big) \tag{18}$$

where $\psi_\alpha(z) = \max\{\alpha z, (\alpha - 1)z\}$ is called the quantile loss.

Quantile regression is more robust against outliers and is flexible enough to handle non-Gaussian data, so we should expect that it is a useful tool for estimating graphs. The quantile regression described above is **parametric** in the sense that we attempt to estimate parameter $\theta$. In the MQGM, we instead model the conditional quantiles in a nonparametric, **additive** way.

Firstly, we fix an ordered set of quantile levels $\mathcal{A} = \{\alpha_1, \ldots, \alpha_r\}$. For each $X_k$, we assume that the $\alpha_\ell$-conditional quantile given $X_{-k}$ can be expressed in the additive form

$$Q_{X_k|X_{-k}}(\alpha_\ell) = b^*_{\ell k} + \sum_{j \neq k} f^*_{\ell k j}(X_j) \tag{19}$$

where $b^*_{\ell k}$ is a real constant intercept term, and $f^*_{\ell k j}$ is a smooth function (not necessarily parametric in form). Note that the indices run through $j = 1 \ldots, p$, $k = 1, \ldots, p$ and $\ell = 1, \ldots, r$.

The most general functional form of the MQGM estimator is defined as a collection of optimisation problems over $\ell$ and $k$ based on $n$ observations. It replaces the $\theta_j$ in (18) with univariate functions $f_{\ell k j} \in \mathcal{F}_{\ell j k}$ of the $x_j^{(i)}$ allowing for a more general additive model, and it includes two penalty terms; one to encourage **sparsity** and the other to encourage **smoothness**.

In this essay, we will not use the most general form but would like the reader to be aware that a generalisation of the following does exist. Fix the quantile level $\ell$ and also the variable of interest, $k$. We specify that the function space can be written as a span of basis

[1] Ali et al. use the notation $y_k$ for the observed random variables and $x_j$ for the latent/unobserved variables. One can easily include latent variables $Z_1, \ldots Z_d$ in our analysis by adding extra parameters.

functions $\mathcal{F}_{\ell jk} = \text{span}\{\phi_j^1, \ldots, \phi_j^m\}$. This reduces the problem to a parametric one, since we can write

$$f_{\ell kj}(x) = \theta_{\ell kj}^T \phi_j(x)$$

where $\phi_j(x) = (\phi_j^1(x), \ldots, \phi_j^m(x))$ and the vector of coefficients $\theta_{\ell jk} \in \mathbb{R}^m$ is our parameter of interest. We use the penalties $\|.\|_2$ (Group Lasso) for sparsity and $\|.\|_2^2$ (ridge) for smoothness.

Then the MQGM optimisation problem is, for $\ell = 1, \ldots, r$ and $k = 1, \ldots, p$:

$$\underset{b_{\ell k}, f_{\ell kj} \in \mathcal{F}_{\ell kj}, j=1,\ldots,p}{\text{minimise}} \sum_{i=1}^{n} \psi_{\alpha_\ell}\Big(x_k^{(i)} - b_{\ell k} - \sum_{j \neq k} \theta_{\ell kj}^T \phi_j(x_j^{(i)})\Big) + \sum_{j \neq k} \Big(\lambda_1 \|\theta_{\ell kj}\|_2 + \lambda_2 \|\theta_{\ell kj}\|_2^2\Big). \quad (20)$$

We can add necessary constraints on top of the MQGM optimisation problem above, notably the **non-crossing constraint**. This arises from the fact that, for $\alpha_\ell < \alpha_{\ell'}$, we must have ordered quantiles $Q_X(\alpha_\ell) < Q_X(\alpha_{\ell'})$. This results in

$$b_{\ell k} + \sum_{j \neq k} f_{\ell kj}(x_j^{(i)}) \leq b_{\ell' k} + \sum_{j \neq k} f_{\ell' kj}(x_j^{(i)}). \quad (21)$$

A variety of other constraints can be applied, as discussed in Liu et al (2016). The MQGM, along with each of these structural constraints, can be implemented using the *alternating direction method of multipliers* (ADMM).

## 6.2 Graph recovery

A question one might ask is: *how do we recover conditional independence relationships from the conditional quantiles?*

Suppose that the variable $X_j$ has no contribution to the $k^{\text{th}}$ conditional quantile $Q_{X_k|X_{-k}}$. Then we have that $Q_{X_k|X_{-jk}}(\alpha) = Q_{X_k|X_{-jk}}(\alpha)$ for every $\alpha \in [0, 1]$. Using the following expression for the conditional CDF of some $U|V$:

$$F_{U|V}(t) = \sup\{\alpha \in [0, 1] : Q_{U|V}(\alpha) \leq t\}$$

it is clear that we must also have $F_{X_k|X_{-k}}(t) = F_{X_k|X_{-jk}}(t)$ for every $t$. Therefore, if we know that the $k^{\text{th}}$ conditional quantiles don't depend on $X_j$ **for every** $\alpha \in [0, 1]$ then we can deduce that $X_k \perp X_j | X_{-jk}$.

In the MQGM, however, we are only estimating the quantiles at $\alpha_1, \ldots, \alpha_r$. We cannot know in general that the conditional distribution of $X_k$ doesn't depend on $X_j$, but we can infer the conditional independence relationships in an **approximate sense**. If our estimates $\hat{f}_{\ell kj} = 0$ and $\hat{f}_{\ell jk} = 0$ for every $\ell = 1, \ldots, r$ then, up to an $r$-step approximation, we estimate that variables $X_k$ and $X_j$ are conditionally independent (and hence do not fill an edge between these variables in the underlying graph).

Ali et al (2016) show that, using this definition of the estimated edge set, and under some assumptions on $p$, $\lambda_1$, $\lambda_2$ and on the underlying distribution, that **the MQGM graph estimate is consistent**. In the Appendix, one can find a statement of the theorem and assumptions used in its proof.

Theoretical benefits of using MQGM over previous methods are that it can model a large family of nonparametric, or even heteroskedastic, data.

# 7 TIGER

We now return to the setting where the data is Gaussian, and address an entirely different but related question. In each of the techniques we have discussed so far, there has involved one or more unspecified **tuning parameters**, $\lambda$. In order for these techniques to be computationally practical, we need an efficient way of choosing $\lambda$.

However, many modern algorithms for tuning parameter selection are not ideal. In this section we will discuss TIGER, a method with a tuning-insensitive property meaning we can avoid selecting an appropriate tuning parameter altogether.

## 7.1 Tuning parameter selection

Theoretical justifications for graph estimation methods might assume we have chosen some optimal, or **oracle** choice of tuning parameter, $\lambda^*$, which cannot be implemented practically. When applying these methods, one must employ some criterion or algorithm for selecting a sensible $\lambda$.

Some popular criterions used include, but are not limited to:

- $v$-fold and leave-one-out **cross validation**
- **AIC** (Akaike information criterion)
- **BIC** (Bayesian information criterion)

In practice, there are problems with each of these methods, and often they are linked to one another. It is often the case, such as in cross validation, that we must calculate many regression estimates (on a smaller dataset) before selecting $\lambda^{\text{cv}}$. Furthermore, many criteria are only justified for low dimensional settings, as $p$ becomes extremely large in comparison with $n$ these methods can become unreliable, or at least take up a significant amount of computation time.

It would be desirable if our graph estimation procedure did not require any time be spent on tuning parameter selection, and chooses $\lambda$ automatically whilst still obtaining a consistent graph estimate. This is what TIGER (Tuning-Insensitive Graph Estimation and Regression) achieves, and it exploits a property of the SQRT-Lasso in order to do this.

## 7.2 SQRT-Lasso and TIGER

For a linear regression problem $Y = \mathbf{X}\beta + \epsilon$, based on $n$ observations of a $p$-dimensional random vector $X$, the SQRT-Lasso estimates $\beta$ via:

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \{ \tfrac{1}{\sqrt{n}} \|Y - \mathbf{X}\beta\|_2 + \lambda \|\beta\|_1 \}.$$

Belloni et al (2012) show that the choice of $\lambda$ is asymptotically universal; this is the key to the tuning-insensitive nature of TIGER. Motivation for the SQRT-Lasso is based around the score function - more detail can be found in the Appendix.

TIGER performs both estimation of the conditional independence relationships (graph estimation) and also the precision matrix itself. In this essay, we will not be concerned with the precision matrix and so only focus on its zeros; $\Omega_{ij} = 0$. We work in the same setting as Neighbourhood Selection: conditional means are modelled linearly

$$X_i = \sum_{j \neq i} \beta_j X_j + \epsilon_i$$

where $\beta^i = (\beta_1^i, \ldots, \beta_p^i)^T$, $\beta_i^i = 0$ and $\epsilon_i$ has zero mean.

Instead of performing a regular Lasso regression as in Neighbourhood Selection (see Section 2.2), one uses the SQRT-Lasso to estimate these coefficients:

$$\hat{\beta}^{i,\lambda} = \underset{\beta : \beta_i = 0}{\operatorname{argmin}} \Big\{ \tfrac{1}{\sqrt{n}} \|\mathbf{X}_i - \mathbf{X}\beta\|_2 + \lambda \|\beta\|_1 \Big\} \tag{22}$$

As before in NS, the zeroes of $\beta_{ij}$ are precisely those coefficients $\beta_j^i$ which are zero. We therefore estimate conditional independence relationships as those which have $\hat{\beta}_j^{i,\lambda} = 0$ and/or $\hat{\beta}_i^{j,\lambda} = 0$.

## 7.3 Computational benefits

For finite samples, we still need to choose an appropriate $\lambda$. Liu and Wang proved consistency results for $\lambda = \pi \sqrt{\frac{\log p}{n}}$, and they recommend $\lambda = \sqrt{\frac{\log p}{n}}$ for practical (finite sample) situations. In any case, virtually zero extra calculations are required to specify the $\lambda$ we wish to use in this SQRT-Lasso regression.

Negating the need for selecting $\lambda$ should lead to recognisable improvements in computational speed. However, it should be noted that this method is still only formulated under the **Gaussian assumption**; we cannot say much about its performance when the data is not truly Gaussian.

It would be an interesting development for future work to attempt to formulate other methods in a tuning-insensitive way, perhaps also by exploiting the SQRT-Lasso. In particular, those methods which remain consistent in the non-Gaussian setting would be desirable to adjust in this sense. A technique such as a 'tuning-insensitive MQGM procedure' would enjoy the computational benefits derived from a lack of tuning parameter selection, whilst also being flexible enough to model a much larger family of distributions.

# 8 Numerical results

Here we will explore how these methods compare with each other numerically in a variety of data settings. The emphasis will be on the accuracy of each procedure in non-Gaussian situations, and comparing with that of Gaussian situations.

R packages used are `huge` (for NS, Glasso and Nonpara.), `flare` (for CLIME and TIGER), `spacejam` (for Spacejam) and `igraph` for plotting graphs, based on an adjacency matrix calculated for each of these estimators. At time of writing, there is no available R package for the MQGM. We will therefore omit it from our experiments, but would like to direct the reader to Ali et al (2016) for more details on how it performs numerically.

## 8.1 Generating data from a graph

The procedure we use for generating data is outlined as follows, and is based off of the procedure in both Meinhausen and Bühlmann (2006) and Liu et al (2009, 2012). First, we simulate $Y_i = (Y_i^{(1)}, Y_i^{(2)}) \overset{\text{iid}}{\sim} U[0,1]^2$ for $i = 1, \dots, p$. Then, we include an edge between nodes $i$ and $j$ with probability $\mathbb{P}\big((i,j) \in E\big) = \frac{1}{2\pi} \exp\big(-\frac{\|y_i - y_j\|_2^2}{2s}\big)$. We restrict the maximum degree to $N = 4$ to encourage sparsity; for each node with degree greater than 4 we remove edges at random until this is satisfied. This defines our 'true' underlying graph.

To generate data from it, we then take the adjacency matrix $A$ from the resulting graph and form a precision matrix $\Omega$ as follows. Set each $\Omega_{ii} = 1$, and each non-zero non-diagonal entry $A_{ij}$ is modified to be $\Omega_{ij} = 0.245$ (equal correlation between dependent variables). All other entries of $\Omega$ are set equal to 0. [Note: the value of 0.245 ensures positive-definiteness].

We then invert $\Omega$ to obtain the form of our correlation matrix, but we must rescale in order to obtain the final $\Sigma$. Using this correlation matrix associated with the random graph generated above, one can sample multivariate normal data as well as other distributions using known methods.

When looking at the performance of each procedure, we will look at a sparse and a non-sparse case. Note that we have a couple of parameters here which control the sparsity of the underlying graph. They are $s$, which controls the probability of including an edge between nodes, and $N$ the maximum degree of each node in the graph. If we increase $N$, we also must change the entries in $\Omega$ to retain positive-definiteness. Below is an example of how the graph changes when we change both $s$ and $N$.

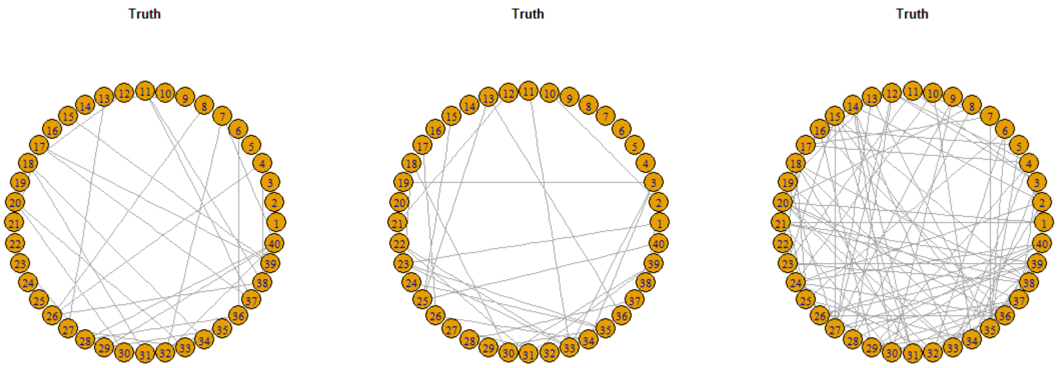

Figure 3: (a) A sparse graph, with $s = 0.1$ and $N = 4$, (b) A graph with $s = 10$ and $N = 4$, and (c) A graph with $s = 0.1$ and $N = 8$. Increasing the value of $s$ has little difference to the graph, whereas increasing $N$ leads to significant change.

The more telling results appear when, for the denser datasets, we keep $s = 0.1$ but increase the maximum degree to $N = 5$ or 6. In this essay we will use $N = 5$.

### 8.2 Area under the ROC curves

A **receiver operating characteristic** (ROC) curve describes the performance of a procedure. It plots the True Positive Rate (TPR, predicting an edge correctly) against the False Positive Rate (FPR, predicting an edge incorrectly) for the output of our procedure across a range of $\lambda$ values. An ideal procedure would have a particular value of $\lambda$ such that the TPR is 100% and the FPR is 0% (that is, the ROC curve passes through the top left corner). An example for Neighbourhood Selection is given in Figure 5.

However, we often do not have such an ideal procedure for finite $n$. An effective quantity for measuring its performance (how close the curve passes to the top left corner) is the Area Under the Curve (AUC) over FPR between 0 and 1. An AUC value close to 1 indicates a well-performing procedure, whereas an AUC value close to 0.5 indicates the method is not

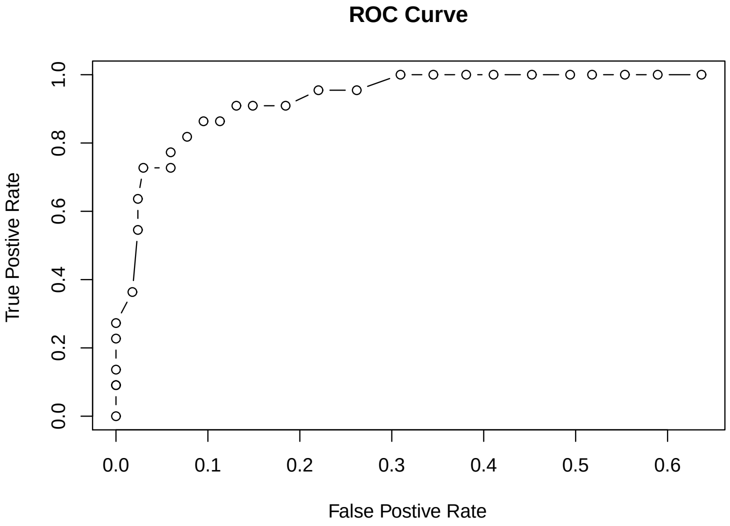

Figure 4: An ROC curve for Neighbourhood Selection when $p = 20$ and Gaussian simulated data, over a range of 30 values of $\lambda$ (i.e. 30 points plotted). The AUC here is 0.953.

much better than randomly guessing the edge set.

In what follows, we will assess the accuracy of each method according to the values of AUC (of the un-smoothed ROC curve) it gives. Similarly to other authors, we will average the AUC by taking the median value across 100 runs, each over 30 $\lambda$-values. Justification for taking the median is that, due to the random nature of our data generation, there will be some outliers which don't accurately represent a data setting. Median AUC values are unaffected by these outliers and therefore provide a better reflection of the *typical* performance of each procedure in a certain data setting.

### 8.3 Results

We will look at data generated a different levels of $p$, in both the sparse case ($N = 4$) and not sparse case ($N = 5$). We do this for multivariate **Gaussian data**, **power-transformed (cubic) Gaussian data** and **Student $t$-distributed data** with 3 degrees of freedom. The procedure used within the nonparanormal SKEPTIC is `glasso`, and in the Spacejam approach the set of basis functions used is $\{x, x^2, x^3\}$.

For the non-Gaussian simulated datasets, we will not consider CLIME or TIGER since they offer little insight, and were designed to aid computation in the Gaussian case. We will however include NS and Glasso in order to compare with the performance of Gaussian procedures.

Timing comparisons will not be measured, since different packages compute a number of other quantities at once (therefore taking longer to complete), and so the time taken for each function to run cannot be compared in a meaningful way.

For the AUC results that follow, standard errors from the 100 trials are of the order of $10^{-3}$, so we have reported figures to 3 significant figures.

***Gaussian data.*** Below is a table of AUC results for Gaussian simulated data.

| | $p = 20$ | | $p = 100$ | | $p = 200$ | |
|---|---|---|---|---|---|---|
| | Sparse | Not sparse | Sparse | Not sparse | Sparse | Not sparse |
| NS | **0.952** | 0.856 | 0.938 | 0.845 | 0.921 | 0.820 |
| Glasso | 0.952 | **0.857** | **0.946** | **0.858** | **0.940** | **0.849** |
| CLIME | 0.915 | 0.807 | 0.888 | 0.767 | 0.842 | 0.725 |
| Nonpara. | 0.937 | 0.833 | 0.934 | 0.838 | 0.924 | 0.830 |
| Spacejam | 0.921 | 0.807 | 0.903 | 0.796 | 0.888 | 0.777 |
| TIGER | 0.875 | 0.727 | 0.752 | 0.621 | 0.702 | 0.585 |

Table 1: AUC results (best in **bold**) averaged over 100 trials for **Gaussian** simulated data, based on $n = 100$ samples.

Broadly speaking, the Glasso, NS and Nonpara. procedures perform similarly and better than the others. Spacejam and CLIME also perform similarly, with TIGER giving lower AUC values than the others. Glasso and NS marginally outperform Nonpara, and the Spacejam approach marginally outperforms CLIME. All procedures are more accurate when the precision matrix is sparse.

We can attribute the fact that the Nonpara. is slightly outperformed by NS and Glasso to us not using the MLE, $\hat{\Sigma}$, instead approximating it using rank based estimators. Spacejam also produces lower AUC values than NS and Glasso, which is to be expected since non-linear terms are not needed to model the conditional independence relationships, so two of the basis functions $\{x^2, x^3\}$ are redundant. This is discussed more in Voorman et al (2014).

TIGER and CLIME are not as accurate as other Gaussian procedures. This can be explained from the theory, since both were designed as **approximations** to NS and Glasso respectively with an aim to aid computation. TIGER approximates the NS objective $n^1\|\mathbf{X}_i - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_1$ by the SQRT-Lasso objective $n^{-1/2}\|\mathbf{X}_i - \mathbf{X}\beta\|_2 + \lambda\|\beta\|_1$. CLIME approximates Glasso's constraint set by a linear one (see section 3.1). These approximations mean that the resulting estimators will inevitably suffer a loss of accuracy in finite sample settings.

***Power transformed (cubic) Gaussian data.*** Below is a table of AUC results for Gaussian simulated data which is then transformed by a power law (in our case, a cubic transformation).

| | $p = 20$ | | $p = 100$ | | $p = 200$ | |
|---|---|---|---|---|---|---|
| | Sparse | Not sparse | Sparse | Not sparse | Sparse | Not sparse |
| NS | 0.796 | 0.702 | 0.780 | 0.684 | 0.755 | 0.670 |
| Glasso | 0.807 | 0.711 | 0.806 | 0.707 | 0.789 | 0.691 |
| Nonpara. | **0.938** | **0.833** | **0.926** | **0.831** | **0.925** | **0.825** |
| Spacejam | 0.782 | 0.671 | 0.769 | 0.681 | 0.774 | 0.675 |

Table 2: AUC results (best in **bold**) averaged over 100 trials for **power transformed** (cubic) simulated Gaussian data, based on $n = 100$ samples.

The Nonpara. procedure outperforms the others, which have similar performances. This is what we would expect, since the random variables are $(Y_1, \ldots, Y_p) = (X_1^3, \ldots, X_p^3)$ where $(X_1, \ldots, X_p) \sim N(0, \Sigma)$. This is a monotone transformation of Gaussian data, which the nonparanormal SKEPTIC is designed for, and the rank-based estimators will work well.

The main talking point of these results is how the Spacejam procedure is not performing well. In fact, it is performing slightly worse than NS and Glasso, which is concerning since we would expect the opposite from non-Gaussian data. In Voorman et al (2014), Spacejam is significantly more accurate in their non-Gaussian data scenario. However, their setup was constructed with conditional means of **additive form**; that is $\mathbb{E}(Y_j|Y_{-j}) = \sum_{i \neq j} f_i(Y_i)$. This is a crucial assumption of their theory - for the power transformed data above, we do not have such additive conditional means. This can be seen in a simple example below.

### Example

Let $(X_1, X_2, X_3) \sim N(0, \Sigma)$ with $\Sigma$ as below, and we calculate

$$\Sigma = \begin{pmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0 \\ 0.5 & 0 & 1 \end{pmatrix} \iff \Sigma^{-1} = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 1.5 & 0.5 \\ -1 & 0.5 & 1.5 \end{pmatrix}.$$

We will show that $\mathbb{E}(X_1^3|X_2, X_3) \neq f_2(X_2) + f_3(X_3)$ for any $f_2$, $f_3$. Indeed,

$$\begin{aligned} f_{X_1|X_2,X_3}(x_1|x_2,x_3) &\propto f_{X_1,X_2,X_3}(x_1,x_2,x_3) \\ &\propto \exp\left[-\tfrac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x}\right] \\ &\propto \exp\left[-(x_1 - \tfrac{x_2+x_3}{2})^2\right] \end{aligned}$$

and therefore $X_1|X_2, X_3 \sim N(\frac{x_2+x_3}{2}, \frac{1}{2}) = N(\mu, \sigma^2)$. Its moment generating function is $m(t) = \exp(\mu t + \frac{1}{2}\sigma^2 t^2)$, from which it can be calculated that

$$\mathbb{E}(X_1^3|X_2, X_3) = m^{(3)}(0) = \frac{3}{2}\left(\frac{x_1+x_2}{2}\right) + \left(\frac{x_1+x_2}{2}\right)^3.$$

This cannot be put into an additive form, due to the mixture terms $x_1x_2^2$ and $x_1^2x_2$.

Therefore, **caution is needed** when using the Spacejam approach; we should only use it as a safe alternative to NS or Glasso when we are reasonably confident that the conditional means take an additive form. If they do not take this form, the Spacejam procedure can perform poorly (that is, similar performance to NS and Glasso in the non-Gaussian setting).

***Student $t$ distributed data.*** Below is a table of AUC results for $t_3$-distributed data.

| | $p = 20$ | | $p = 100$ | | $p = 200$ | |
|---|---|---|---|---|---|---|
| | Sparse | Not sparse | Sparse | Not sparse | Sparse | Not sparse |
| NS | 0.802 | 0.695 | 0.830 | 0.744 | 0.843 | 0.744 |
| Glasso | 0.790 | 0.691 | 0.820 | 0.733 | 0.826 | 0.746 |
| Nonpara. | **0.904** | **0.793** | **0.894** | **0.796** | **0.939** | **0.934** |
| Spacejam | 0.795 | 0.697 | 0.837 | 0.736 | 0.843 | 0.745 |

Table 3: AUC results (best in **bold**) averaged over 100 trials for simulated data from a **Student $t$ distribution** with 3 degrees of freedom, based on $n = 100$ samples.

Again, the Nonpara. estimates dominate the others, and one can show that the conditional means also do not have additive form in this case (which accounts for the relatively poor performance of Spacejam).

Interestingly, each procedure seems to improve as $p$ increases. They also perform better with large $p$ when we take a mean average over trials rather than median. As a first attempt to understand what happens to the distribution as $p$ increases, we look at the PDFs of a multivariate $t$ and a multivariate normal distribution.

Student's $t$ distribution is similar to the normal distribution but with a **fat tail**, with higher degrees of freedom corresponding to a less fat tail (and hence closer to normal). The multivariate $t$ distribution with $p$ variables and $\nu$ degrees of freedom, written as $\mathbf{X} \sim t_{p,\nu}(\mu, \Sigma)$, has a PDF of $f_{p,\nu}(\Delta^2) \propto (1+\frac{1}{\nu}\Delta^2)^{-(\nu+p)/2}$, where $\Delta^2 = (\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)$ is the Mahalabonis distance. Below is a plot of this pdf with various values of $p$, alongside that of a multivariate normal distribution.

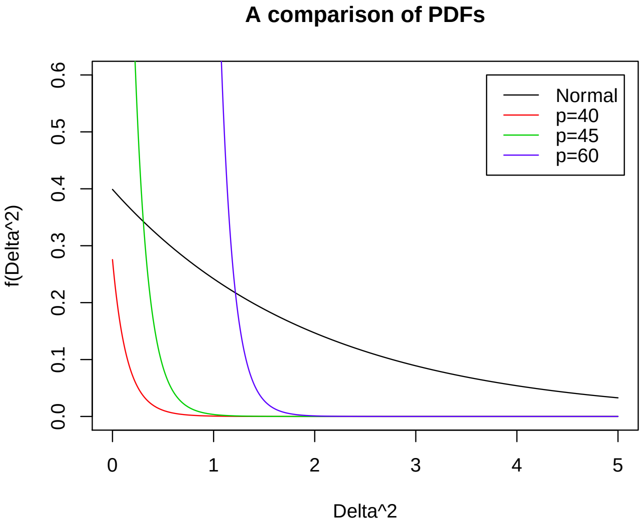

Figure 5: As $p$ increases the PDF of the multivariate $t$ distribution appears to have a thinner tail. Here, we are using $\nu = 3$ degrees of freedom and have included a Normal PDF for reference.

The multivariate $t$ distribution's tail is becoming thinner with increasing $p$ (note that the PDFs are normalised to 1, so the lines eventually overlap for large enough $\Delta^2$). This may begin to explain why our estimates are improving with $p$; it is possible that the distributions are becoming more like normal distributions, a setting we know the procedures perform well in. It would be an interesting direction to explore whether this pattern continues for different $\Sigma$ or $\Omega$, different $\nu$ and even larger $p$.

# 9 Conclusion

We have discussed the need for graphical models in representing conditional independence relationships between random variables, and then summarised the Neighbourhood Selection and Graphical Lasso techniques for the high-dimensional setting. NS estimates the conditional means linearly, and the Glasso attempts to estimate the zeroes of the precision matrix. They are linked by an approximation, as discussed in section 2.3.

Every technique we subsequently studied can be viewed as adapting either the Glasso framework or the NS framework in some way. CLIME approximates the constraint set in Glasso in order to be formulated in a computationally effective way. Nonparanormal SKEPTIC is an extension of Glasso, in the sense that distributions are allowed to be transformed monotonically beforehand, enabling us to study non-Gaussian data.

Spacejam models the conditional means in a similar vein to NS, except it models them as additive functions which allow for non-Gaussian data. In fact, if we use just one linear basis function, this approach reduces down to NS exactly. We also studied TIGER which, whilst being restricted by the Gaussian assumption, has a tuning-insensitive property which improves computational feasibility. TIGER can be seen as a modification to NS, using a SQRT-Lasso regression. Recently, MQGM has been proposed as another extension of NS, except we are modelling conditional quantiles rather than conditional means.

We then experimented with these procedures in three data generation settings; Gaussian, power-transformed Gaussian, and multivariate $t$. Our numerical results largely agree with what we would expect, but highlighted where care must be taken in choice of procedure. Firstly, CLIME and TIGER underperform in terms of accuracy compared with the others, but do possess a computational advantage. They are useful in situations where there are time/computation restrictions.

Secondly, if we are not confident that the conditional means of our random variables are additive in form, Spacejam might not be a safe alternative to Gaussian techniques. In this case, estimates in the non-Gaussian cases were marginally worse than NS and Glasso. In both non-Gaussian settings we looked at, the nonparanormal SKEPTIC estimates outperformed the others.

In modern contexts, graphical modelling procedures need to be able to handle large quantities of high-dimensional data effectively. They also need to be flexible, since the underlying distribution may not be simple. We have introduced and compared several new approaches - as well as highlighting where there is still work to be done - making it an important and exciting area of current research.

# 10 References

[1] Meinshausen, N. and Buhlmann, P. High-dimensional graphs and variable selection with the lasso, *The Annals of Statistics*, 1436-1462, (2006)

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

[3] Cai, T., Liu, W. and Luo, X. A Constrained $\ell_1$ Minimization Approach to Sparse Precision Matrix Estimation, arXiv:1102.2233, (2011)

[4] Liu, H., Han, F., Yuan, M., Lafferty, J. and Wasserman, L. High-dimensional semiparametric Gaussian copula graphical models, *The Annals of Statistics*, 22932326, arXiv:1202.2169, (2012)

[5] Voorman, A., Shojaie, A., Witten, D. Graph estimation with joint additive models, *Biometrika* 101 (1), 85-101, arXiv:1304.4654, (2014)

[6] Ali, A., Kolter, J. Z. and Tibshirani, R. J. The Multiple Quantile Graphical Model, arXiv preprint arXiv:1607.00515, (2016)

[7] Liu, H., Wan, L. TIGER: A Tuning-Insensitive Approach for Optimally Estimating Gaussian Graphical Models, arXiv:1209.2437, (2012)

[8] Shah, R. D. Modern Statistical Methods *Part III lecture notes,* available at http://www.statslab.cam.ac.uk/ rds37/modern_stat_methods.html

[9] Dempster, A. P. Covariance Selection, *Biometrics*, Vol. 28, No. 1, Special Multivariate Issue (Mar., 1972), pp. 157-175 (1972)

[10] Rothman, A. J., Bickel, P. J., Levina, E., Zhu, G. Sparse permutation invariant covariance estimation, *Electronic Journal of Statistics*, Vol. 2, 494–515 (2008)

[11] Khatri, C. G., Rao, C. R. Characterisations of Multivariate Normality I: Through Independence of some Statistics, *Journal of Multivariate Analysis* 6, 81-94 (1976)

[12] Kendall, M. G. Rank correlation methods, *Griffin* (1948)

[13] Kruskal, W. H. Ordinal Measures of Association, *Journal of the American Statistical Association* 53 No. 284. 814–861 (1958)

[14] Christensen, D. Fast algorithms for the calculation of Kendall's $\tau$, *Computational Statistics* 20 51-62 (2005)

[15] Arlot, S., Celisse, A. A survey of cross-validation procedures for model selection *Statistics Surveys* Vol. 4 40–79 (2010)

[16] Akaike, H. Information theory and an extension of the maximum likelihood principle *2nd International Symposium on Information Theory, Tsahkadsor, Armenia, USSR, September 2-8, 1971* (1973)

[17] Schwarz, G. E. Estimating the dimension of a model *Annals of Statistics*, 6 (2): 461–464 (1978)

[18] Belloni, A., Chernozhukov, V. and Wang, L. Square-root lasso: Pivotal recovery of sparse signals via conic programming, *Biometrika* 98 791–806 (2012)

# 11 Appendix

Here will be a few extra details which were mentioned but omitted from the essay, and are intended to help the reader understand the methods better.

### 11.1 CLIME: formulating the problem as a linear program

**Lemma 2** *Solutions $\tilde{\Omega}$ to (8) can be written as $\tilde{\Omega} = (\hat{\beta}_1, \dots, \hat{\beta}_p)$ with columns $\hat{\beta}_i$ which solve (9), for $i = 1, \dots, p$:*

$$\text{minimise } \|\beta\|_1 \text{ subject to } \|S\beta - e_i\|_\infty \leq \lambda$$

*Proof:* Write $\Omega = (\omega_1, \dots, \omega_p)$, where $\omega_i \in \mathbb{R}^p$. Then

$$\|S\Omega - I_p\|_\infty \leq \lambda \iff \|S\omega_i - e_i\|_\infty \leq \lambda \text{ for } i = 1 \dots, p$$

and therefore the solution to (8) with columns $\tilde{\omega}_i$ **also** satisfies the constraint of (9), so we must have, for $i = 1, \dots, p$,

$$\|\tilde{\omega}_i\|_1 \geq \|\hat{\beta}_i\|_1. \tag{23}$$

Write $\hat{B} = (\hat{\beta}_1, \dots, \hat{\beta}_p)$. (23) is true for every column, therefore we can deduce that $\|\tilde{\Omega}\|_1 \geq \|\hat{B}\|_1$, with equality only when $\|\tilde{\omega}_i\|_1 = \|\hat{\beta}_i\|_1$ for every $i = 1, \dots, p$.

Moreover, since $\|S\hat{B} - I_p\|_\infty \leq \lambda$, we also have $\|\tilde{\Omega}\|_1 \leq \|\hat{B}\|_1$. From these we know that $\|\tilde{\Omega}\|_1 = \|\hat{B}\|_1$. Thus $\tilde{\Omega}$ has columns which solve (9), completing the proof. □

### 11.2 TIGER: Motivating the SQRT-Lasso

First, formulate the linear regression problem as $Y = \mathbf{X}\beta_0 + \sigma\epsilon$ where $Y \in \mathbb{R}^n$, $\mathbf{X} \in \mathbb{R}^{n \times p}$ are observed, $\epsilon_i$ are i.i.d with mean 0 and variance 1 for $i = 1, \dots, n$, and $\sigma$ is an unknown noise parameter. Let $\beta_0$ be the 'true' value that we want to estimate.

Define $\hat{Q}(\beta) := \frac{1}{n}\|Y - \mathbf{X}\beta\|_2^2$ to be the least squares objective. Then we can write the two regression problems as:

$$\text{Lasso:} \quad \tilde{\beta} = \arg\min_\beta \left\{ \hat{Q}(\beta) + \lambda\|\beta\|_1 \right\}$$

$$\text{SQRT Lasso:} \quad \hat{\beta} = \arg\min_\beta \left\{ \sqrt{\hat{Q}(\beta)} + \lambda\|\beta\|_1 \right\}.$$

One can show that, if $\epsilon_i \sim N(0,1)$, then choosing $\lambda = \sigma \cdot 2c\sqrt{n}\Phi^{-1}(1-\frac{\alpha}{2p})$ for some $c > 1$ leads to near oracle performance for the Lasso estimate; $\|\tilde{\beta} - \beta_0\|_2 \leq A\sigma\sqrt{\frac{s\log(2p/\alpha)}{n}}$ with probability at least $1-\alpha$. However, this choice of $\lambda$ depends on the unknown $\sigma$. This is where we require $\sigma$-estimation, or the $\lambda$-selection criteria described in section 7.

The key quantity involved in $\lambda$-selection is the **score**; which is an indicator of the 'estimation noise' within our problem. For the Lasso, the score is defined as

$$\hat{S}^{\text{Lasso}} = \nabla_\beta \hat{Q}(\beta)|_{\beta_0} = 2\sigma \cdot \tfrac{1}{n}\mathbf{X}^T\epsilon.$$

That fact this depends on $\sigma$ indicates that our choice of the penalty level $\lambda$ will require some kind of estimation of $\sigma$. For the SQRT-Lasso, the score is

$$\hat{S}^{\text{SQRT}} = \nabla_\beta \sqrt{\hat{Q}(\beta)}\Big|_{\beta_0} = \frac{\nabla_\beta \hat{Q}(\beta)|_{\beta_0}}{2\sqrt{\hat{Q}(\beta_0)}} = \frac{\sigma \cdot \frac{1}{n}\mathbf{X}^T\epsilon}{\sqrt{\sigma^2 \frac{1}{n}\epsilon^T\epsilon}} = \frac{\frac{1}{n}\mathbf{X}^T\epsilon}{\sqrt{\frac{1}{n}\epsilon^T\epsilon}}$$

which is **independent of** $\sigma$. Therefore, our choice of penalty level $\lambda$ will not require estimation of $\sigma$. This is the key idea behind why using the SQRT-Lasso instead of the Lasso obviates the need for tuning parameter selection.

### 11.3 MQGM: Graph recovery theorem

Here we will state the graph recovery theorem and describe the assumptions used in its proof. For the proof itself, the reader should refer to Ali et al (2016).

**Theorem 4** *Assume* $\log p = o(n^{2/21})$ *and assumptions 1-4. Assume also that the tuning parameters* $\lambda_1$*,* $\lambda_2$ *satisfy*

$$\lambda_1 = \Theta\Big(\sqrt{mn\log(p^2mr/\delta)\log^3 n}\Big) \quad \textit{and} \quad \lambda_2 = o(n^{41/42}/\theta^*_{\max}),$$

*where* $\theta^*_{\max} = \max_{\ell,k,j}\|\theta_{\ell kj}\|_2$*. Then, for* $n$ *sufficiently large, the MQGM graph estimate* $\hat{E}$ *exactly equals the true graph structure* $E^*$ *with probability at least* $1-\delta$.[2]

***Assumptions***

Fix $k$, $\ell$ and let $\Phi \in \mathbb{R}^{n\times pm}$ be the matrix consisting of blocks $\Phi_{ij} = \phi(x_j^{(i)}) \in \mathbb{R}^m$. Write also $\Phi_j$ to be the $j^{\text{th}}$ submatrix consisting of all $n$ rows and a block of $m$ columns from $\Phi$, for each $j = 1\ldots,p$. Define the effective **error** terms $\epsilon_{\ell ki} = x_k^{(i)} - b^*_{\ell k} - \sum_{j\neq k}\phi(x_j^{(i)})^T\theta^*_{\ell kj}$, over $i = 1,\ldots,n$. Define also the **underlying support** to be $S_{\ell k} = \{j : \theta^*_{\ell kj} \neq 0\}$.

The assumptions used in the main theorem can be summarised/interpreted as:

[2] Here we use big-$\Theta$ notation where $f(n) = \Theta(g(n))$ if $\exists C, D > 0$ such that $C|g(n)| \leq |f(n)| \leq D|g(n)|$, for all $n$ sufficiently large.

1. For $j \in S_{\ell k}^c$, we require an upper bound on the correlation between columns of submatrices of $\Phi_j$ and $\Phi_{\S_{\ell k}}$.

2. We assume a certain level of smoothness of the true distribution; in fact the assumption used in the proof is $f_{\epsilon_{\ell k}}$ being Lipschitz over $x$ in a neighbourhood of the origin.

3. An upper and lower bound on the minimum eigenvalue of $\Phi_{S_{\ell k}}^T \Phi_{S_{\ell k}}$.

4. An assumption on the basis and support sizes. In the proof, we use $m = O(n^{1/9})$ and $s = |S_{\ell k}| = O(n^{1/21})$. This mirrors for example, when using splines in nonparametric regression the optimal value of $m$ is $O(n^{1/3})$.

Note that, if for each $\ell = 1, \ldots, r$, $k = 1, \ldots, p$ the event $\hat{S}_{\ell k} = S_{\ell k}$ happens with probability at least $1 - \delta/pr$, then by applying a union bound over all $\ell, k$ we can deduce that $\hat{E} = E^*$ with probability at least $1 - \delta$. Broadly, the proof consists of showing that the KKT conditions of one optimisation problem match those of the restricted problem with high probability, using each of the four assumptions above at different points in the proof. For full details, see Ali et al (2016).