Department of Mathematics
Robust Statistics
Celestijnenlaan 200B B-3001 LEUVEN

Robust Estimates of Precision Matrix

Adhithya Unni Narayanan, Kristen Michelle Nader,
Nika Pountouchachvili & Sifan Shen

April 2021

Contents

1 Introduction

Many statistical applications make use of the inverse of the sample covariance matrix Σ^{-1}, also called the precision matrix. However, as high dimensionality ($p > n$) introduces singularity in the covariance matrix, we cannot calculate the precision matrix in this setting.

The next section is going to motivate why we would still like to use the precision matrix. To fix the singularity issue, high-dimensional estimators like graphical lasso (glasso) exist. This might solve the singularity problem, but is not very robust. In the following sections we will study the effect of inputting different robust estimators for the sample covariance matrix in glasso and compare them with each other.

2 Motivation

Covariance matrix estimation has become an interesting area of research in recent years. It is unbiased, consistent under different assumptions and easily computed. However, the sample covariance matrix only performs well in an idealistic world–where data is clean. In this instance, they are considered highly non-robust which has fueled the need for robust statistical techniques to be derived in order to estimate the covariance matrix under real-world circumstances–when outliers are present in the data known as contaminated data.

In many fields, the precision matrix is used (usually instead of the covariance matrix). The precision matrix is traditionally computed as the inverse of the covariance matrix where $\Theta = \Sigma^{-1}$. Applications where the precision matrix is used instead of the covariance include computation of the Mahalanobis distance (equation 1), linear discriminant analysis, and Gaussian graphical models.

$$D_M(x, y) = \sqrt{(x-y)^T S^{-1}(x-y)} \text{ where } S^{-1} \text{ is the precision matrix.} \tag{1}$$

Such findings have been used in network reconstruction such as in genetics and neuroscience.

When working in a low-dimensional setting ($p \leq n$, where p is the number of explanatory variables and n is the number of samples), a reasonable method for computing a robust precision matrix is by computing a robust estimate of the covariance matrix and taking its inverse.

However, in fields such as medicine, e-commerce, industrial applications and bioinformatics, researchers often work with high-dimensional datasets, which has incentivized the need for high-dimensional analogues of the robust estimators. Several methods have been developed such as the Graphical lasso (glasso), quadratic approximation method for sparse inverse covariance learning (QUIC) and the constrained L1 minimization for inverse matrix estimation (CLIME).

So, what exactly makes the precision matrix so attractive that we want it in high dimensions? There are a couple of reasons for this. Firstly, a precision matrix is always symmetric positive definite (which we also expect from its estimator). Remember, we can only speak of the actual precision matrix if the covariance matrix is non-singular. This makes it symmetric positive definite because of there being no zero eigenvalue. The inverse of a symmetric matrix is also symmetric. And the inverse of a positive definite matrix is also positive definite. This is not terribly difficult to see. If λ is an eigenvalue of an invertible matrix A, then $\frac{1}{\lambda}$ is an eigenvalue of A^{-1}, and vice versa. As all eigenvalues of a positive definite matrix are positive, so are the eigenvalues of its inverse.

Another good reason for using it is because conditional independence of variables translates to sparsity in the precision matrix. This could be illustrated [2] by a random walk $X_1, X_2, X_3, \ldots, X_n$ where $X_i = X_{i-1} + Z_i$, $X_1 = Z_1$ and Z_j $(j = 1, 2, \ldots, n)$ stands for a random event with distribution $N(0, 1)$. This random walk can be written as $X = AZ$, where

$$A = \begin{pmatrix} 1 & 0 & 0 & \ldots \\ 1 & 1 & 0 & \ldots \\ 1 & 1 & 1 & \ldots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}.$$

The covariance matrix Σ of X can be written as AA^T, and the precision matrix Θ is the inverse of this. This gives us the following two matrices with interesting structures:

$$\Sigma = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & 2 & 2 & \ldots & 2 \\ 1 & 2 & 3 & \ldots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & 3 & \ldots & n \end{pmatrix}, \Theta = \begin{pmatrix} 2 & -1 & 0 & \ldots & 0 \\ -1 & 2 & -1 & \ldots & 0 \\ 0 & -1 & 2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -1 \\ 0 & 0 & \ldots & -1 & 2 \end{pmatrix}.$$

In addition, the precision matrix is very sparse. In fact, the only non-zero entries are for neighbouring events. This makes sense, as the value of X_i depends solely on X_{i-1} and not on anything that came before that. This is what we mean with conditional independency creating sparsity. This notion of sparsity allows computational efficiencies when working with the precision matrix.

Bayesian analysis of the multivariate normal distribution also finds a good use for the precision matrix. Suppose the prior probability distribution of an uncertain quantity–in other words, a distribution that would express one's beliefs about this quantity before some evidence is taken into account–and the likelihood both have Gaussian form. If the precision matrices of both exist, then the precision matrix of the posterior probability is just the sum of the two.

3 Brief introduction to robust estimators

Graphical lasso (glasso), the quadratic approximation method for sparse inverse covariance learning (QUIC) and the constrained L_1-minimization for inverse matrix estimation (CLIME) are the three most proper methods for the computation of high-dimensional sparse precision matrices. These estimators are limited in the calculation for clean data considering that they start with nonrobust sample covariance matrix.

However, we always need to deal with contaminated data. In this situation, one single outlier can cause a huge loss of information when we downweigh this observation. Additionally, the selection mechanism for outliers can vary with different variables, thus resulting in very small probability of clean data.

To improve on the loss of information in cellwise contamination and lack of robustness from the traditional robust procedures, different precision matrix estimators in high dimensions are used. Instead of the nonrobust sample covariance matrix, we use a cellwise robust covariance matrix in glasso, which returns a sparse precision matrix estimator robust to cellwise observation (see section 4). In this case, we would require an input matrix S which is symmetric and positive semidefinite.

The robust precision matrix estimator $\hat{\Theta}_S$ is 'correlation based' when the covariance matrix S is based on pairwise correlation, and 'covariance based' when S is based on pairwise covariance. With cellwise outliers, the correlation based precision matrix estimator has a breakdown value of 50 % (see section 3.2), and the covariance based precision matrix estimator is with a breakdown value of at most 25 % (see section 3.1). In either situation, S should be positive semidefinite to get a positive definite $\hat{\Theta}_S$.

3.1 Covariance-based Precision Matrix Estimators

Covariance-based precision matrix estimator is based on the idea that we can compute robust covariance by robust variance of two variables X and Y:

$$Cov(X,Y) = \frac{1}{4\alpha\beta}[Var(\alpha X + \beta Y) - Var(\alpha X - \beta Y)] \tag{2}$$

where $\alpha = 1/\sqrt{Var(X)}$ and $\beta = 1/\sqrt{Var(Y)}$. We get a robust covariance estimate when we use a robust variance estimator.

However, although the approach is direct and easy to understand, there exist two major problems:

1. the calculation between different variables may generate unexpected outliers, thus resulting in a breakdown point of less than 25%;
2. the output covariance matrix may not be positive semidefinite.

To improve on the second drawback, orthogonalized Gnanadesikan-Kettenring (OGK) and the computation of the nearest positive (semi)definite matrix (NPD) is needed to transform the output matrix into something positive semidefinite.

3.2 Correlation-based Precision Matrix Estimators

To obtain the robust estimator S of the covariance matrix, Croux and Öllerer [3, 1] start with the robust correlation estimator. The formula is shown below:

$$s_{jk} = scale(x^j)scale(x^k)r(x^j, x^k) \text{ where } j, k = 1, ..., p. \tag{3}$$

Here, we takes the robust Q_n-estimator as the scale estimator scale(), since Q_n has the highest possible breakdown value of 50 % [4] and efficient at the normal model.

3.2.1 Two-step Estimators

For the correlation $r(x^j, x^k)$ in equation 3, Öllerer and Croux [3, 1] considered 3 ways to get the robust correlation estimator:

1. **The Quadrant Correlation** [4], defined as

$$r_{Quadrant}(x^j, x^k) = \frac{1}{n}\sum_{i=1}^{n} sign((x_{ij} - med_{l=1,..,n}x_{lj})(x_{ik} - med_{l=1,..,n}x_{lk})) \tag{4}$$

   Here, sign(.) represents the sign function.

2. **The Spearman correlation** [4] This correlation iss defined as the sample correlation of the ranks of the observations:

$$r_{Spearman}(x^j, x^k) = \sum_{i=1}^{n} \frac{(R(x_{ij} - \frac{n+1}{2})(R(x_{ik} - \frac{n+1}{2})}{\sqrt{\sum_{i=1}^{n}(R(x_{ij} - \frac{n+1}{2})^2 \sum_{i=1}^{n}(R(x_{ik} - \frac{n+1}{2})^2}} \tag{5}$$

   with $R(x_{ij})$ the rank of x_{ij} among all elements of x_j, for any $1 \leq j \leq p$ and $1 \leq i \leq n$.

3. **The Gaussian rank correlation** [5] is defined as the sample correlation estimated from the normal scores of the data:

$$r_{Guass}(x^j, x^k) = \frac{\sum_{i=1}^{n} \Phi^{-1}(\frac{R(x_{ij})}{n+1})\Phi^{-1}(\frac{R(x_{ik})}{n+1})}{\sum_{i=1}^{n}(\Phi^{-1}(\frac{i}{n+1}))^2} \tag{6}$$

   with $\Phi()$ the standardized and normalized cumulative distribution function .

Among the three methods, Quadrant correlation was determined to be the least efficient estimator and Gaussian rank correlation is the only one consistent at the normal distribution. Both Quadrant correlation and Spearman correlation, are inconsistent at the bivariate normal model. Although simulations (see [1] and section 6.1) show that Spearman's asymptotic bias (bias that the estimator converges to as n grows) is not that bad. One would need to conduct a transformation to obtain consistency. Besides, Quadrant correlation usually has higher asymptotic bias than Spearman correlation. Gaussian rank correlation is also the easiest for computation because the covariance matrix output is already in a positive semidefinite form.

3.2.2 Three-step Estimators

As we discussed above, an additional transformation step is needed for Spearman and Quadrant correlation estimation so that the corresponding S becomes a consistent estimator of Σ.

To resolve the inconsistency the authors suggest the following transformation of the correlation measures

$$\tilde{r}_{Spearman} = 2\sin\left(\frac{\pi}{6} r_{\text{Spearman}}\right) \tag{7}$$

where r_{Spearman} was computed by equation 5. The Quadrant correlation will be transformed using equation 4 as

$$\tilde{r}_{Quadrant} = \sin\Big(\frac{\pi}{2} r_{Quadrant}\Big) \tag{8}$$

These transformations erase the positive definiteness of S and therefore cannot be used as an input matrix to glasso. This issue is solved by applying an additional step which can be implemented in 2 ways:

1. **Pertubation Method**: In this method, one simply adds a non-negetive value to all diagonal elements of S. Thus after the addition of this step, the resulting covariance matrix will not have any negetive eigenvalues.

$$S_{pertub} = S + |\min(0, \min_j \lambda_j)| \tag{9}$$

2. **Nearest Positive (Semi)Definite(npd) Method [6]** : is defined as:

$$S_{npd} = \sum_{j=1}^{p} \max(0, \lambda_j) \mathbf{v}_j \mathbf{v}_j^t \tag{10}$$

   where S_{npd} is the nearest positive semi-definite matrix to S and the metric of *nearness* is measured through the Frobenius norm.

3.2.3 Overview

In conclusion, the paper highlights two methods in which the correlation based precision matrices can be used :

For a two-step estimate:

1. Compute S using either inconsistent Quadrant and Spearman or Gaussian Rank correlation obtained from equation 6
2. Use graphical lasso to compute $\hat{\Theta}_S$

For a three-step estimate:

1. Compute S using Quadrant or Spearman correlation using equations 8 or 7 respectively
2. Transform S into a positive semi-definite matrix using either the perturb method shown in equation 9 or nearest positive definite(npd) method in equation 10
3. Use graphical lasso to compute the $\hat{\Theta}_S$

The next section will delve deeper into graphical lasso and why a robust covariance matrix through glasso gives us a robust precision matrix in the next section.

4 Graphical Lasso

Graphical lasso (glasso) is a technique used to estimate a sparse, inverse covariance matrix using an L1 penalty. In order to understand glasso, we introduce concepts from statistical learning known as regularization and shrinkage. These techniques include the least absolute shrinkage and selection operator (lasso) or L1 regularization, and Ridge L2 regularization.

4.1 Understanding Lasso

Regularization techniques are often explained in the context of regression. For example, in linear regression one attempts to find the model coefficients β that minimize the loss function:

$$L(\beta) = \|y - X\beta\|_2^2 \tag{11}$$

One could solve this by means of ordinary least squares, which gives us an unbiased estimate due to its convexity. Nevertheless, in case of multicollinearity, their variances are large which deviates the observed value far away from the true value. Adding a constraint to the regression estimates reduces the possibility of the model becoming too complex by regularizing the coefficients of the estimates or even shrinking the estimates towards zero. Depending on the shrinkage technique, estimates can be shrunk towards zero or set to exactly zero.

In Ridge regression, a penalty term is added which is equal to the sum of the square coefficient estimates. E.g. the linear regression problem extends from minimizing the loss function to minimizing the function

$$L^{(2)}(\beta) = \|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2 \text{ where } \|\beta\|_2^2 \text{ represents the L2 norm} \tag{12}$$

A tuning parameter λ is added to this expression in order to control the penalty term. When $\lambda = 0$, the penalty term has no effect and the ridge regression produces ordinary least squares estimates. However as $\lambda \to \infty$, the effect of the shrinkage increases because the penalty increases. In this case, ridge regression results in coefficient estimates that shrink towards zero. Due to the fact that the coefficients are shrunken towards zero but never set to zero, Ridge regression does not perform variable selection.

Lasso regression and Ridge regression have similar formulations. The penalty term that is added to the least squares problem is the sum of the absolute values of the coefficient estimates, which results in the following minimizing function:

$$L^{(1)}(\beta) = \|y - X\beta\|_2^2 + \lambda\|\beta\|_1 \text{ where } \|\beta\|_1 \text{ represents the L1 norm} \tag{13}$$

Similar to Ridge, lasso shrinks the coefficients towards zero and a tuning parameter is added to the penalty term. If the tuning parameter is sufficiently large, the L1 norm ($|\beta\|_1$) may force some coefficients to be set to exactly zero resulting in a variable selection technique. Therefore, lasso results in sparse models which are easier to interpret. In both techniques, the choice of tuning parameters λ is critical. Croux and Öllerer [3] [1] highlight 2 methods that can be used to determine the optimal value of the regularization term λ. In a more traditional approach, λ can be chosen based on its ability to minimize some criterion, e.g. AIC or BIC. Alternatively, it can be chosen using a machine-learning approach known as cross-validation to minimize some metric such as the sum of the squared residuals. Cross-validation appeals to the ability to choose parameters that generalize well on unseen data.

4.2 Understanding Glasso

Let X be a matrix from a p-dimensional Gausian distribution with zero mean and a positive semidefinite covariance matrix Σ to be estimated. Let Θ be the precision matrix in high-dimension space. If $\Theta_{ij} = 0$, then X_i and X_j are conditionally independent. Glasso is a regularization technique that tries to estimate a sparse precision matrix. It minimizes the L1 regularized negative log-likelihood function.

$$\hat{\Theta}_S = \underset{\substack{\Theta=(\theta_{jk})\in\mathbb{R}^{pxp} \\ \Theta\succ 0}}{\operatorname{argmax}} \log\det(\Theta) - tr(S\Theta) - \lambda \sum_{\substack{j,k=1 \\ j\neq k}}^{p} |\boldsymbol{\Theta}_{jk}|. \tag{14}$$

Here, S stands for the sample covariance matrix. The third term represents the sum of the absolute values of the precision matrix and λ is the tuning parameter that controls the amount of L1 shrinkage and hence the sparsity of the resultant precision matrix. Equation 1 defines a maximization over all positive definite precision matrices $\Theta \succ 0$. If $\lambda = 0$, the resultant matrix is the inverse covariance matrix. As $\lambda \to \infty$, the more sparse the precision matrix becomes thus presenting researchers with a trade-off between sparsity and a high likelihood function. Therefore, glasso employs techniques seen previously: BIC or cross-validation.

The choice for λ through the Bayesian Information Criterion is by picking the lowest BIC value for ten pre-picked values, where BIC is calculated as

$$BIC(\lambda) = -\log\det\hat{\boldsymbol{\Theta}}_\lambda + \operatorname{tr}\left(\hat{\boldsymbol{\Theta}}_\lambda \mathbf{S}\right) + \frac{\log n}{n}\sum_{i\leq j}\hat{e}_{ij}(\lambda),$$

where $\hat{e}_{ij} = 1$ if $\left(\hat{\boldsymbol{\Theta}}_\lambda\right)_{ij} \neq 0$ and $\hat{e}_{ij} = 0$ otherwise. The ten values for λ come from a heuristic approach implemented by the package **huge** in R: the largest of the values is calculated by

$$\lambda_{\max} = \max\left(\max_{(i,j)\in\{1,\ldots,p\}^2}(\mathbf{S}-\mathbf{I}_p)_{ij}, -\min_{(i,j)\in\{1,\ldots,p\}^2}(\mathbf{S}-\mathbf{I}_p)_{ij}\right)$$

and the smallest value $\lambda_{\min}$ is defined as $0.1\lambda_{\max}$. The values for λ are spaced logarithmically by taking ten equally spaced values between $\lambda_{\min}$ and $\lambda_{\max}$ and transforming them through the exponential function. Notice that this means that there is always done some regularization, even for non-singular covariance matrices.

For K-fold cross-validation the same ten values are used. To optimize λ the following procedure is followed:

1. Split the data in K equally sized blocks
2. fix a λ
3. leave out one of the K blocks and estimate $\hat{\Theta}_\lambda^{(-k)}$ $(k = 1, \ldots, K)$ through the glasso algorithm

4. calculate the negative log-likelihood on the data from the left-out block (where $\mathbf{S}^{(k)}$ is computed from the test data)

$$L^{(k)}(\lambda) = -\log\det\hat{\Theta}_{\lambda}^{(-k)} + \operatorname{tr}\left(\mathbf{S}^{(k)}\hat{\Theta}_{\lambda}^{(-k)}\right).$$

5. Once $L^{(k)}(\lambda)$ is computed for all k and λ, we choose the λ value that minimizes the negative log-likelihood

$$\hat{\lambda} = \arg\min \frac{1}{K}\sum_{k=1}^{K} L^{(k)}(\lambda).$$

Öllerer and Croux [1] recommend a 5-fold cross-validation and highlight that BIC is computed quicker than cross-validation but tends to select more sparse models. While cross-validation may be computationally intensive, it does result in more accurate models and is recommended by the authors.

4.2.1 How does it actually work?

Graphical lasso represents the extension of lasso to graphs where the goal is to induce from the data a sparse undirected graph. As stated in previous sections, if a value of the precision matrix Θ at entry (i, j) is 0, this indicates conditional independence between variables X_i and X_j. Which translates to the absence of an edge between vertices X_i and X_j in its undirected-graph representation. Undirected graphs are often used due to the fact that each node represents a single variable and the absence of an edge represents this conditional independence [7]. The algorithm proposed by Friedman et al. describes glasso as an algorithm that loops through all variables and fits a lasso regression onto each of the variables. These lasso models are then solved by coordinate descent.

Algorithm 1: Glasso algorithm

```
Start with an estimate of the covariance matrix W where $W = S + \lambda I$
while has not converged or the max number of interactions has been reached do
    For each dimension $1 \to p$ solve the lasso problem using $W_{11}$ and $s_{12}$. The output will
      be a $p-1$ vector $\hat{B}$. Update $W$ using $w_{12} = W_{11}\hat{B}$
end
Recover $\hat{\Theta}$. Up to this point, the algorithm has estimated $\hat{\Sigma} = W$ where $\hat{\Theta} = W^{-1}$
```

Figure 1 displays results from flow cytometry data from Sachs et al. [8]. The figure displays the effect of various L1 norms and their effect on the connectivity of a graphical network. Figure 1A represents the true acyclic graph from cell signaling data. As the L1 norm decreases(1B,C,D,E,F), the networks becomes increasingly sparse.

As input, glasso requires a symmetric positive semidefinite matrix and returns a symmetric positive definite matrix. A matrix M is considered symmetric positive semidefinite if it is symmetric and all its eigenvalues are non-negative. Because this implies that zero could be an eigenvalue (and therefore the nullspace is non-trivial), the inverse matrix does not exist. In the case of a positive definite matrix, eigenvalues are strictly positive.

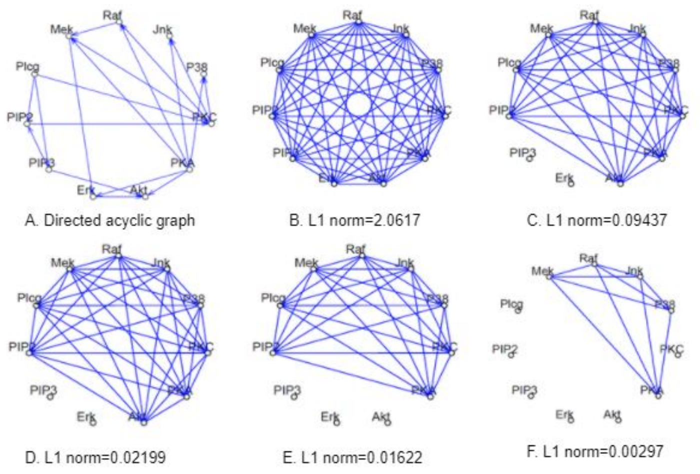


Figure 1: **Cell Signaling data** Sachs et al. (2003) **A**: Believed network **B-F**:glasso representation using undirected graphs with various L1 norm

4.2.2 Breakdown Value of Graphical Lasso

When working with a robust estimator, one is usually interested in its breakdown point. For the glasso estimator we should pay extra attention; it is, a priori, not clear whether the cellwise robustness of the sample covariance matrix translates well to its robust precision matrix counterpart. However, Öllerer and Croux [3] proved that the final precision matrix estimator $\hat{\Theta}_S$ inherits the breakdown point of the covariance matrix estimator S. More formally, let us define the *finite-sample breakdown point under cellwise contamination* of a precision matrix $\hat{\Theta}$ as

$$\epsilon_n(\hat{\Theta}, X) = \min_{m=1,2,\ldots,n} \left\{ \frac{m}{n} \mid \sup_{X^m} D\left(\hat{\Theta}(X), \hat{\Theta}(X^m)\right) = \infty \right\}.$$

Here, X^m is a contaminated dataset where at most m cells in each column have arbitrary values, $D(A,B) := \max\left\{\left|\lambda_1(\mathbf{A}) - \lambda_1(\mathbf{B})\right|, \left|\lambda_p(\mathbf{A})^{-1} - \lambda_p(\mathbf{B})^{-1}\right|\right\}$ and the eigenvalues of A are sorted as $0 \leq \lambda_p(A) \leq \cdots \leq \lambda_1(A)$. In other words, we take m as large as possible until an eigenvalue gets either arbitrarily large or close to zero. Similarly, we can define the *explosion* finite-sample breakdown point under cellwise contamination of a covariance matrix S as

$$\epsilon_n^+(S, X) = \min_{m=1,\ldots,n} \left\{ \frac{m}{n} \mid \sup_{X^m} \left|\lambda_1(S(X)) - \lambda_1\left(S\left(X^m\right)\right)\right| = \infty \right\}.$$

Suppose S has a breakdown value of $\frac{m}{n}$, then the following theorem [3] tells us that $\hat{\Theta}_S$ (obtained through glasso) has a breakdown value of at least that high.

Theorem 1. *The finite sample breakdown point under cellwise contamination of the robust precision matrix estimator $\hat{\Theta}_S(X)$ fulfills*

$$\epsilon_n\left(\hat{\Theta}_{\mathrm{S}}, \mathbf{X}\right) \geq \epsilon_n^+(\mathbf{S}, \mathbf{X})$$

with S a positive semidefinite covariance estimator.

The above result is absolutely great. It means that, the more robust we can make our covariance matrix estimator, the more robust the precision matrix estimate derived from that covariance matrix is. In short, our glasso-obtained estimate is at least as good as our robust covariance estimate. All we need to do is find a consistent robust estimator for S. The explosion breakdown point of a univariate scale estimator scale(.) is defined as

$$\epsilon_n^+(\text{scale}, x) = \min_{m=1,\ldots,n} \left\{ \frac{m}{n} \mid \sup_{x^m} \text{scale}\left(\mathrm{x}^m\right) = \infty \right\}.$$

The following theorem tells us something about the breakdown value of the covariance matrix estimator whenever it is based on pairwise correlations.

Theorem 2. *Suppose the covariance matrix S has elements*

$$S_{jk} = \text{scale}\left(x^j\right) \text{scale}\left(x^k\right) r\left(x^j, x^k\right),$$

where scale(.) *is a univariate scale estimator and $r(.,.)$ is a correlation estimator. Then the explosion breakdown point of the covariance estimator depends on the explosion breakdown point of the scale estimator used*

$$\epsilon_n^+(S, X) \geq \max_{j=1,\ldots,p} \epsilon_n^+\left(\text{scale}, x^j\right).$$

Both MAD and the Q_n-estimator have an explosion breakdown point of 50% [9], which in turn gives the correlation based precision matrix estimator a breakdown point of 50% under cellwise contamination. This means that performing glasso on a Q_n-estimator based covariance matrix yields a highly robust precision matrix.

4.2.3 Applications of Glasso

GLASSO has been used in more practical setting such as in modeling cancer genetic networks using gene expression data [10] and in network modeling for Functional magnetic resonance imaging [11].

Zhao et al. [10] investigated how Gaussian graphical models were able to infer genetic networks to reveal the conditional dependence between genes from RNA-seq data from 15 types of human cancer . Due to the fact that GLASSO infers an un-directed graph, a node represents an expressed gene and an edge represents the conditional dependence between two expressed genes. The set of edges $E = \{e_{i,j}\}$ describe the conditional dependencies. A value set to 1 indicates that the genes i and j are conditionally dependent whereas no edge- or a value set to 0- indicates conditional independence. This relationship can be visualized in Figure 2 However, they reported that graphical lasso may still be computationally expensive when used to construct genome-wide networks [10].

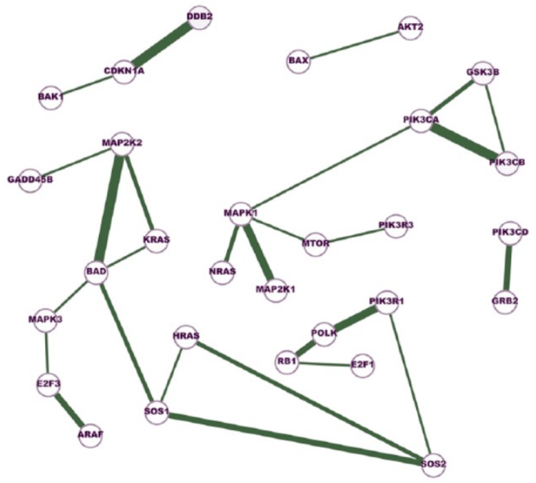


Figure 2: **Network of Cancer gene interactions**. Each gene is represented as a node and an edge depicts the conditional dependence

5 Results of the paper

In this section we will be looking into the results of the paper by Croux and Öllerer [1]. They investigated performances of different precision matrix estimators–which we reviewed in Section 3.2–when inputted in glasso by comparing them using a simulation study. Section 5.5 provide the results of the paper.

5.1 Simulation Setup

In the setup of the simulation study, Croux and Öllerer [1] used four sampling schemes to cover different patterns of the true precision matrix $\Theta_0 \in \mathbb{R}^{p \times p}$

1. **banded**: $(\Theta_0)_{ij} = 0.6^{|i-j|}$
2. **sparse**: $\Theta_0 = B + \delta I_p$ with $\mathbb{P}[b_{ij} = 0.5] = 0.1$ and $\mathbb{P}[b_{ij} = 0] = 0.9$ for $i \neq j$. The parameter δ is chosen such that the conditional number of Θ_0 equals p. Then the matrix is standardized to have unit diagonals.
3. **dense**: $(\Theta_0)_{ii} = 1$ and $(\Theta_0)_{ij} = 0.5$ for $i \neq j$.
4. **diagonal**: $(\Theta_0)_{ii} = 1$ and $(\Theta_0)_{ij} = 0$ for $i \neq j$.

For each sampling scheme, the authors generate M simulations with samples of size n from a multivariate normal $N(0, \Theta_0^{-1})$. Take dimensions as p and to each dataset and add cellwise contamination which is ε. The performance of the different estimaters $\hat{\Theta}$ were comapred using the Kullback-Liebler(KL) divergence. This is explained in the Section 5.3 'Performance Measures' .

5.2 Simulation Results

Simulation results are the comparison of KL values of the consistent three-step estimator to the inconsistent two-step estimator. For the three-step estimator, Croux and Öllerer used both Nearest Positive Definite(NPD) and pertubation methods. They also included the consistent two-Step estimators, Gaussian rank correlation, considering our assumption of normality on the simulation data. As a benchmark, they made a comparison with the classical Glasso where the sample covariance matrix is given as input matrix S in equation(2). The false positive, false negative rates 5.4 and the Kulback-Leiber divergence 5.3 were also compared for different values of n, p and ε.

5.3 Performance Measures

Two commonly used scoring systems for divergence are the Kullback-Leibler(KLD) Divergence and the Jensen-Shannon Divergence(JSD).

The Kullback-Leibler divergence is a measure of statistical distance. KLD calculates a score that measures the divergence of one probability distribution to another. In the context of precision matrix and its estimation, KLD is used as a measure to determine how close the obtained estimate of the precision matrix $\hat{\Theta}$ is from the true precision matrix Θ_0. The lower the score, the closer the probability distributions are or the closer the estimated precision matrix is to the true matrix.

An interesting property of the Kullback-Leiber divergence is the fact that it is non-symmetric and unbounded and can therefore take values from [0- $+\infty$]. A score of 0 represents identical statistical entities being compared. In the context quantification of the performance of precision matrix estimators , the kullback-Leiber can be reformulated such that the estimate of the precision matrix and the true matrix are taken as input. The closer the estimate is to the true matrix, the closer the KL score is to 0. The less accurate the estimate, the larger the score becomes. Equation 15 formalizes this concept:

$$KL(\hat{\Theta}, \Theta_0) = \mathrm{tr}\Big(\Theta_0^{-1}\hat{\Theta}\Big) - \log\det\Big(\Theta_0^{-1}\hat{\Theta}\Big) - p \tag{15}$$

The Jensen-Shannon Divergence is another measure in which the difference between statistical entities can be quantified. It is formalized as a normalized extension of the KL divergence that is also symmetric and bounded [0,1]. In this case, a value of 0 indicates identical entities being compared while 1 represents maximally different entities.

The paper presented by Croux and Öllerer [1] use the Kulback-Leiver diveregence and therefore the rest of the report will focus on KLD as a measure of divergence between the estimated matrix and the true matrix.

5.4 FP and FN

As stated in earlier sections, glasso is often used in order to derive an undirected graph from the data. In this case, another unit of measurement can be implemented to determine how well the graph structure,in terms of sparsity, can be reformulated.

For the measurement of the recovered sparseness, false positive (FP) and false negative(FN) are introduced:

$$FP = \frac{|(i,j : i = 1, ..., n; j = 1, ..., p : (\hat{\Theta}_{ij} \neq 0 \wedge (\Theta_0)_{ij} = 0|}{|(i,j) : i = 1, ..., n; j = 1, ..., p : (\Theta_0)_{ij} = 0|} \tag{16}$$

$$FN = \frac{|(i,j : i = 1, ..., n; j = 1, ..., p : (\hat{\Theta}_{ij} = 0 \wedge (\Theta_0)_{ij} \neq 0|}{|(i,j) : i = 1, ..., n; j = 1, ..., p : (\Theta_0)_{ij} \neq 0|} \tag{17}$$

Öllerer and Croux [3] define the false positive rate as the percent of zero elements in the true precision matrix that are estimated incorrectly as nonzero. In terms of graphical models, a false positive is placing an edge between nodes when no edge should exist.

The false negative rate gives the percent of nonzero elements in the true precision matrix that is estimated incorrectly as a zero.Once again using the illustration of graphical models, a false negative is not placing an edge between two nodes when one should exist.For both the False positive and False negative rates, a smaller value indicates better result.

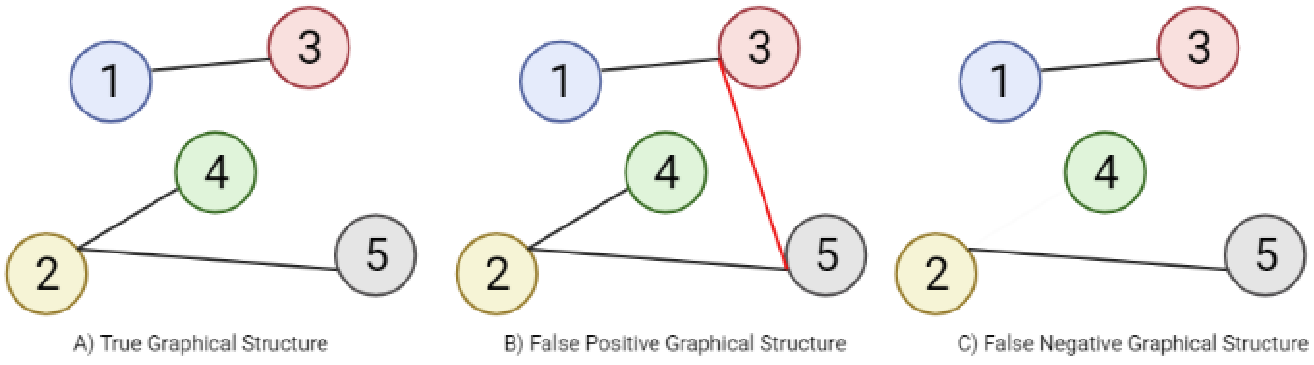

1
3
4
2
5
A) True Graphical Structure
1
3
4
2
5
B) False Positive Graphical Structure
1
3
4
2
5
C) False Negative Graphical Structure

Figure 3: **Graphical representation of False positive and False negative edgesA:** represents the true undirected graph.**B:** Graphical structure where an edge is wrongly added between node 3 and 5-representing a false positive **C:** An edge is wrongly removed between node 2 and 4- a false negative

5.5 Results from Croux et al. [1]

Under a low dimensional setting, shown in Table 1 where p=3, the Spearman correlation method performed better in terms of the Kullback-Leiber divergence criterion when compared to the Gaussian Rank correlation which in turn performed better than the Quadrant Correlation. Under contaminated model at p=3, the 2 step methods report better KLD values.

In a higher dimensional setting, where p=100, and under little contamination 2 step quadrant outperforms 3 step quadrant. Spearman 2 step and 3step have similar results and Gaussian also performs well. Under contamination where p=100, GLASSO results are not reliable, having a

Table 1: KL divergences simulated [1] by taking $n = 50$, $M = 1000$, $p = 3\&100$ and the contamination rates $\varepsilon = 0.0\&0.1$.

KL divergence	$p = 3$		$p = 100$	
ε	0%	10%	0%	10%
2-step Quadrant	0.59	1.08	38.70	55.54
3-step Quadrant (npd)	0.33	0.81	49.54	71.02
2-step Spearman	0.29	0.98	38.62	55.66
3-step Spearman (npd)	0.27	0.96	39.47	57.01
2-step Gaussian Rank	0.27	1.05	38.62	55.49
Glasso	0.23	4.12	38.00	145.56
Sample Covariance	0.14	3.54	NA	NA

KLD of 145! The best results are reported to be done using two-step estimators: Quadrant and Spearman.

Croux et al. also investigated on the performance of recovery of the sparse true precision matrix for p=30 and P=100, which is measured by FP and FN. From their result, it is shown that all estimators have similar FP and FN value. Besides, an increase in FN value is observed with a higher outlier rate and high dimension.

6 Our results and Discussion

In this section, we will be implementing the methods described above on datasets simulated under the 'banded' scheme. Along with this, the simulation study will be applied to two real datasets and their results will be discussed and compared to conclude which estimator worked best.

6.1 Simulation Study

Simulation study involves datasets with dimensions $p = 5$ and $p = 100$ of sample size $n = 50$. The number of simulations M is kept at 100 and the cellwise contamination ε is restricted to 0 and 10%. We comparesthe values of KLD, FP and FN of the Two-Step Quadrant and Spearman estimators, Three-Step Quadrant and Spearman estimators, Two-Step Gaussian estimators and the classical glasso. For the Three-Step Quadrant and Spearman estimator, we use only the npd method. The sampling scheme of the datasets is 'banded'.

Table 2: Kullback-Leibler divergence for simulation done with $p = 5$, $n = 50$, $M = 100$ and both 0% and 10% cellwise contamination.

	FP rate		FN rate		KL divergence	
ε	0%	10%	0%	10%	0%	10%
Two-Step Quadrant	NA	NA	0.3912	0.5408	1.0397	1.8314
Three-Step (npd) Quadrant	NA	NA	0.1728	0.2848	0.7256	1.4379
Two-Step Spearman	NA	NA	0.2616	0.4408	0.6407	1.6604
Three-Step (npd) Spearman	NA	NA	0.2400	0.9732	0.6025	1.6067
Two-Step Gaussian	NA	NA	0.2680	0.9744	0.6080	1.7434
Glasso (Sample Covariance)	NA	NA	0.2320	0.7272	0.5209	6.7000
Sample Covariance Inverted	NA	NA	0.0000	0.0000	0.3636	5.6519

1. Simulation with $p = 5$, $n = 50$, $M = 100$ and $\varepsilon = 0$ and 10% (The clean dataset and dataset with 10% of cellwise cotamination)

   - Comparing the KL values: for the clean dataset, the inconsistent Two-Step Quadrant estimator displays higher KL value when compared to the consistent Three-Step estimator. This trend is expected as it indicate that on addition of the npd step to the estimator, the performance improves. Even in the presence of outliers, the value of Two-Step Quadrant is higher than that of the Three-Step one. The value difference within the Two-Step and Three-Step estimators however has reduced by in the presence of ouliers but the difference in the value measures are not significant. There is no significant difference in the value of Spearman in both the scenarios.
   - Comparing the FP and FN values: table 2 shows empty values for FP. This is due to the way our precision matrix is set up: in the i'th diagonal, all the elements are $0.6^{|i|}$. Because $p = 5$, the lowest value in this matrix is 0.6^5, which is far from zero. The denominator in equation 16 is the number of zero elements in the precision matrix. As we have zero zero elements, we divide by zero, and as we all know, this is not possible. In the case of the false negatives, we see that the inverted sample covariance matrix does not nullify any non-zero elements in the actual precision matrix, which is remarkable. The second-best achiever in low dimensions–for both contaminated and uncontaminated data–is the three-step quadrant. All other methods, except for the inconsistent two-step quadrant, share similar results in the uncontaminated dataset. However, once contamination is introduced we notice that the very consistent estimators (Gaussian and npd Spearman) nullify far too many non-zero elements in the precision matrix.
   - Comparing with the classical Glasso: for the clean dataset, the KL value for the Glasso is shown to be the least when compared to all other estimators while for the dataset with the outliers, Glasso indicates the highest value in the lot. This behaviour is anticipated as classical Glasso is not expected to perform well in the presence of outliers.

2. Simulation with $p = 100$, $n = 50$, $M = 100$ and $\varepsilon = 0$ and 10% (High dimensional dataset as $n < p$)

Table 3: Kullback-Leibler divergence, FN- and FP-rate for the simulation done with $p = 100$, $n = 50$, $M = 100$ and both 0% and 10% cellwise contamination.

	FP rate		FN rate		KL divergence	
ε	0%	10%	0%	10%	0%	10%
Two-Step Quadrant	0.0040	0.0048	0.9713	0.9736	36.9139	53.8530
Three-Step (npd) Quadrant	0.0045	0.0039	0.9711	0.9748	46.9347	69.6952
Two-Step Spearman	0.0060	0.0038	0.9582	0.9735	34.5811	54.1576
Three-Step (npd) Spearman	0.0060	0.0040	0.9586	0.9732	35.3732	55.3345
Two-Step Gaussian	0.0064	0.0039	0.9566	0.9744	34.2626	54.1092
Glasso (Sample Covariance)	0.0068	0.0066	0.9545	0.9749	33.1049	142.3242

- Comparing the KL values: unlike the datasets before, the KL values of inconsistent Two-Step quadrant is lower than that of the consistent Three-Step Quadrant for both the clean dataset and in the presence of outliers. This means that the addition of a step have not improved the performance of the estimator. There is a change in the value measure and this the change is not significant. For the Spearman method, both the Two-Step and Three-Step estimators shows almost same values and thus we can conclude that addition of a step in the Three-Step estimator haven't really improved the performance.
- Comparing the FP and FN values: the False Positive values for all the estimators have slightly increased comparing in the high-dimensional case, but still quite low (close to 0), which shows that the association between variables are well found. For the False Negative values, the values for all estimators are close to 1, implying high missing rate of non-zero edges that should have been detected. With the breakdown point increasing from 0 to 10%, the False Negative values has slightly increased with all approaches, which means that all the estimators have high tendency to estimate a lot more zero elements that should be non-zero.
- Comparing with the classical Glasso: for the clean dataset, the KL value for the Glasso is still the smallest comparing to those estimators, however, Glasso shows the highest KL value with contamination. This is consistent with our formal discussion: classical Glasso doesn't preform well with the presence of outliers, and robust estimators greatly increased the performance.

6.2 Application to real datasets

We use 3-step Quadrant, 3-step Spearman and the Gaussian rank correlation to get the robust estimator S of the covariance matrix respectively, and then apply glasso to get the corresponding precision matrix $\hat{\Theta}_S$.

6.2.1 Breast Cancer Wisconsin Dataset [12]

Breast cancer Wisconsin dataset is a record for the measurement of breast cancer cases. In this dataset there are 683 observations ($n = 683$) and 10 ordinal variables ($p = 10$). The data is contaminated with a 35% outlier rate.

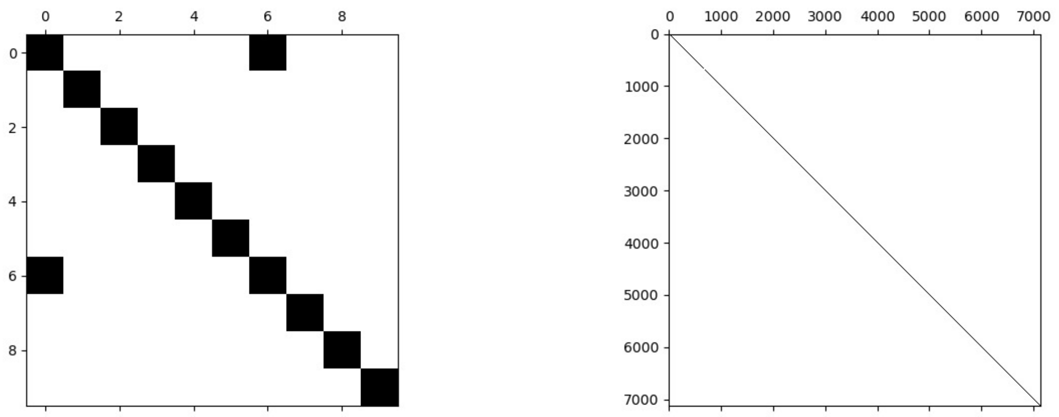
0.278802545158344	0	0	0	0	0	-0.158255440092003	0	0	0
0	2.56304559683161	0	0	0	0	0	0	0	0
0	0	2.56304559683161	0	0	0	0	0	0	0
0	0	0	2.56304559683161	0	0	0	0	0	0
0	0	0	0	2.56304559683161	0	0	0	0	0
0	0	0	0	0	2.56304559683161	0	0	0	0
-0.158255440092003	0	0	0	0	0	0.278802545158344	0	0	0
0	0	0	0	0	0	0	2.56304559683161	0	0
0	0	0	0	0	0	0	0	2.56304559683161	0
0	0	0	0	0	0	0	0	0	2.56304559683161

Table 4: Precision Matrix Based on Quadrant Correlation

0.233674879461011	0	0	0	0	0	-0.102204560010693	0	0	0
0	2.56304559683161	0	0	0	0	0	0	0	0
0	0	2.56304559683161	0	0	0	0	0	0	0
0	0	0	2.56304559683161	0	0	0	0	0	0
0	0	0	0	2.56304559683161	0	0	0	0	0
0	0	0	0	0	2.56304559683161	0	0	0	0
-0.102204560010693	0	0	0	0	0	0.233674879461011	0	0	0
0	0	0	0	0	0	0	2.56304559683161	0	0
0	0	0	0	0	0	0	0	2.56304559683161	0
0	0	0	0	0	0	0	0	0	2.56304559683161

Table 5: Precision Matrix Based on Spearman Correlation

The precision matrices we get from the Quadrant, Spearman and Gaussian estimations are shown in table 4, 5 and 6, and the spy plot for $\hat{\Theta}_S$ (shared by three estimators due to the same matrix structure) is shown in figure 4 left panel. We can observe that the estimated matrix is diagonal with one non-zero off-diagonal element, which can be (if this is not a false positive) due to a conditional dependence between variable 1 and 6, which are clump thickness and the amount of bare nuclei. Also, all tables share fairly similar entries.



Breast Cancer Wisconsin Dataset Golub Dataset

Figure 4: Spy Plot of the Precision Matrix

0.228271372559259	0	0	0	0	0	-0.0947140840335356	0	0	0
0	2.56304559683161	0	0	0	0	0	0	0	0
0	0	2.56304559683161	0	0	0	0	0	0	0
0	0	0	2.56304559683161	0	0	0	0	0	0
0	0	0	0	2.56304559683161	0	0	0	0	0
0	0	0	0	0	2.56304559683161	0	0	0	0
-0.0947140840335356	0	0	0	0	0	0.228271372559259	0	0	0
0	0	0	0	0	0	0	2.56304559683161	0	0
0	0	0	0	0	0	0	0	2.56304559683161	0
0	0	0	0	0	0	0	0	0	2.56304559683161

Table 6: Precision Matrix based on Gaussian Rank Correlation

6.2.2 Golub(1999) [13]

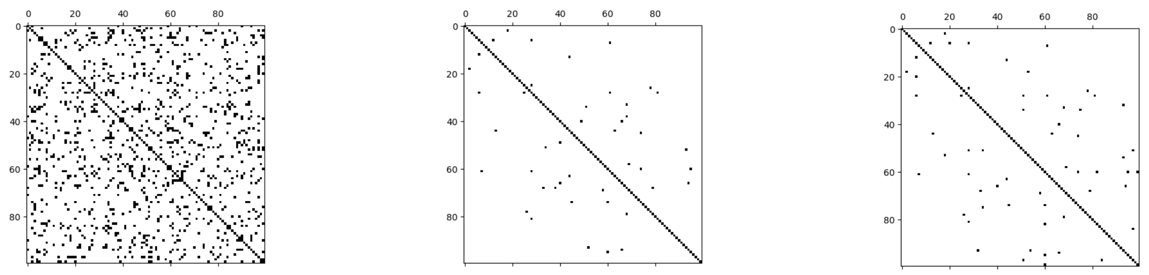
Golub is a dataset which records the expression of 7129 genes of 72 patients with acute leukemia. There is 72 observations (n=7) and 7129 variables (p=7129). Considering the very high dimensions this dataset holds, the outliers are assumed to exist.

The precision matrices we get from the three estimators are too big to show in the report, so here we use a spy plot instead in figure 4. The matrix is strictly diagonal, which implies that there is no conditional probability between the different variables. We get this result because of a couple of factors:

1. The heuristic for λ-selection we mentioned in section 4.2 did not work for this specific case. The computed λ_{max} was in the order of $8 \cdot 10^5$. Which means that λ_{min} was in the order of $8 \cdot 10^4$. There is no doubt about it that this greatly and unnecessarily penalizes the occurrence of any non-diagonal element. Artificially lowering the λ values results in an unreasonably long waiting time for glasso to converge.

2. The computation of BIC depends on computing the determinant. As the values of this dataset (and therefore also its estimated covariance matrix) can be very large, the determinant can often evaluate to infinity no matter the method we use to compute it.

So, instead of taking all 7129 variables into account, we decide to sort all variables by their absolute sums and take the first 100. This should fix all issues listed above and is still high-dimensional ($n = 72, p = 100$).



Quadrant correlation Spearman correlation Gauss correlation

Figure 5: Spy plots of the estimated precision matrices.

If the results were to be analyzed and interpreted from a biological perspective, the resultant precision matrix using the breast cancer dataset displayed in Figure 4 is not biologically relevant. The biological system is not static and all elements are interacting in order to maintain homeostasis. In this way, we can consider that genes are also interacting. By some external stimulus, such as cellular stress, some genes will be up-regulated which increases the quantity of messenger RNA (mRNA) such that more proteins may be synthesized in response to the stimulus. In turn, as levels of this mRNA and hence the levels of the protein encoded by the mRNA increases, other genes will either be up-regulated or down-regulated. An illustration is shown in Figure 6 and Figure 1 described in Section 4.2.3. In this case, protein X , encoded by gene X, is needed in order for a gene Y to be transcribed and for protein Y to be synthesized thus linking gene X and gene Y as an interaction. Therefore, we would expect results as shown in Figure 5 showing such important interactions.

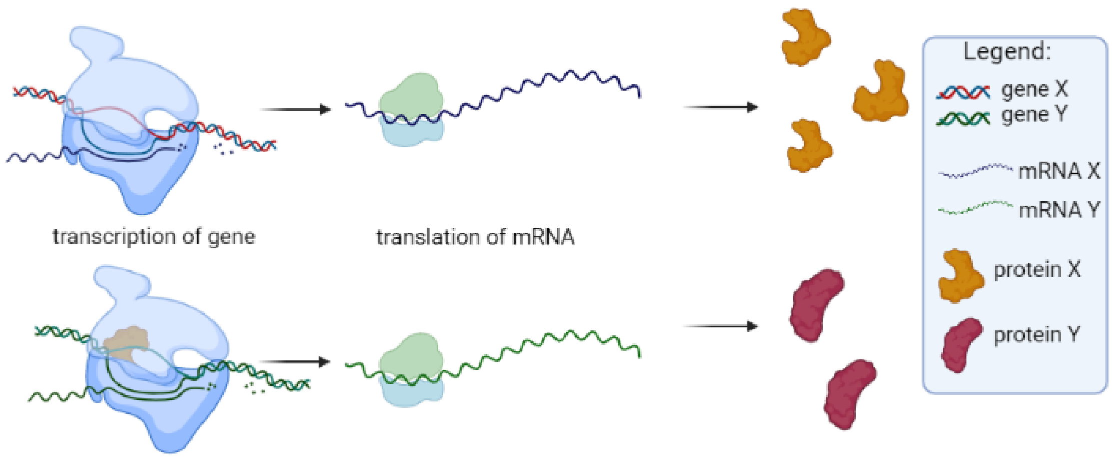


Figure 6: Illustration on the interaction of genes and gene products

According to Zhao et al. [10], a genetic network inferred from gene expression data ”is expected to be sparse”. Therefore, from our simulation we can safely assume that, in high dimensions, Spearman and Gauss correlation are more accurate. The result from quadrant correlation tells us that a lot of genes are conditionally dependent, or in other words a lot of gene expressions depend on the outcome of other gene expressions. However, genes interact in interesting ways and uncovering gene interactions is of interest to the bioinformatics and biology community. If these results were to be validated, an in-depth analysis of each of these genes would have to be performed .

An interesting discovery is that the estimated precision matrix from Quadrant correlation method results in more dependencies. However, the paper [3, 14] reveal that the Quadrant correlation method may be less efficient than the Spearman correlation and the Gaussian rank correlation.

7 Conclusion

From Table 1, 2-step Gaussian Rank correlation method outperforms all other methods, including glasso on the sample covariance matrix, under high contamination of 10% and high dimension. However, when working in a low dimension dataset, the 3-step Quadrant results in lower Kullback-Leibler Divergence indicating better performance. In low dimension data, in both low and high percent contamination, the 3-step quadrant is considered as one of the best methods (yielding low KL divergence and low FP and FN rates). Due to the fact that we are entering into an era of high dimensional data where data is never guaranteed to contain 0% contamination, a two-step Gaussian rank correlation method is recommended.

While glasso on the sample covariance matrix performed poorly in the simulations of the reference paper and our own simulations, there are modifications of graphical lasso that exist in the hopes of improving its results. For example, a weighted glasso (wglasso) was developed that incorporates prior knowledge which has improved network associations in its applications on reconstruction of gene networks [15].

References

[1] C. Croux and V. ¨ Ollerer, *Robust and Sparse Estimation of the Inverse Covariance Matrix Using Rank Correlation Measures*, 01 2016, pp. 35–55.

[2] M. Stephens, "Multivariate normal: the precision matrix," https://stephens999.github.io/fiveMinuteStats/normal_markov_chain.html, 02 2016, accessed: 2021-04-04.

[3] V. Öllerer and C. Croux, "Robust high-dimensional precision matrix estimation," 2015.

[4] C. Croux and C. Dehon, "Influence functions of the spearman and kendall correlation measures," *Tilburg University, Center for Economic Research, Discussion Paper*, vol. 19, 01 2010.

[5] K. Boudt, J. Cornelissen, and C. Croux, "The gaussian rank correlation estimator: Robustness properties," *Statistics and Computing*, vol. 22, pp. 471–483, 10 2010.

[6] P. J. Rousseeuw and G. Molenberghs, "Transformation of non positive semidefinite correlation matrices," *Communications in Statistics–Theory and Methods*, vol. 22, no. 4, pp. 965–984, 1993.

[7] T. Zhao, H. Liu, K. Roeder, J. Lafferty, and L. Wasserman, "The huge package for high-dimensional undirected graph estimation in r," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 1059–1062, 2012.

[8] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[9] C. Croux and C. Dehon, *Robust Estimation of Location and Scale.* American Cancer Society, 2014. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118445112.stat07416

[10] H. Zhao and Z.-H. Duan, "Cancer genetic network inference using gaussian graphical models," *Bioinformatics and biology insights*, vol. 13, p. 1177932219839402, 2019. [Online]. Available: https://europepmc.org/articles/PMC6456846

[11] S. M. Smith, K. L. Miller, G. Salimi-Khorshidi, M. Webster, C. F. Beckmann, T. E. Nichols, J. D. Ramsey, and M. W. Woolrich, "Network modelling methods for fmri," *NeuroImage*, vol. 54, no. 2, pp. 875–891, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053811910011602

[12] O. L. Mangasarian, W. N. Street, and W. H. Wolberg, "Breast cancer diagnosis and prognosis via linear programming," *Operations Research*, vol. 43, no. 4, pp. 570–577, 1995.

[13] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri *et al.*, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *science*, vol. 286, no. 5439, pp. 531–537, 1999.

[14] C. Croux and C. Dehon, "Influence functions of the spearman and kendall correlation measures," *Statistical methods & applications*, vol. 19, no. 4, pp. 497–515, 2010.

[15] Y. Li and S. A. Jackson, "Gene network reconstruction by integration of prior biological knowledge," *G3: Genes, Genomes, Genetics*, vol. 5, no. 6, pp. 1075–1079, 2015.