Homework 3 - Analysis of High Dimensional Data

Tavis Abrahamsen, Ray Bai, Syed Rahman and Andrey Skripnikov

Consider the Gaussian graphical model as presented in class. In a seminal paper Meinshausen and Buhlmann (Annals of Statistics, 2006) showed that one can estimate the model by using node-wise penalized (lasso) regression, followed by post-processing to obtain a symmetric and positive definite estimate of Ω.

1. Using your code from HW 2, estimate a Gaussian graphical model using the lasso node-wise regression method.

2. Apply your algorithms to a chain network, a nearest neighbor network, and a scale free network of size p = 25 with n = 150 observations. Set the density level of your edge set at 10 (see Section 4 of Guo et al. (Biometrika, 2011)).

3. Replicate the setting in (2) 10 times and provide an estimate of the false positive and false negative rates, as well as the Frobenius norm of the difference between your estimate and the true Ω.

The Meinshausen-Buhlmann method to estimate the partial correlations consists of running the following *lasso* regressions for $i = 1, ..., p$:

$$\min_{\tilde{\beta}_i} \frac{1}{2}\|X_i - \sum_{j \neq i} \beta_{ji} X_j\|_2^2 + \lambda \|\tilde{\beta}_i\|_1 \tag{1}$$

where $\tilde{\beta}_i = (\beta_j)_{j \neq i, 1 \leq j \leq p}$. Our goal in this assignment is to estimate the sparsity patterns and the partial correlation matrices corresponding to chain graphs, nearest neighbor networks and Barabasi graphs with density equal to approximately 10%. Without some post-processing this method doesn't usually produce positive definite estimates and hence is mainly used as a method of estimating the sparsity pattern as opposed to the concentration matrix itself.

To generate Ω corresponding to a chain graph we first construct a 25×25 covariance matrix, Σ where the (i, j)th element of Σ, $\sigma_{i,j}$, is defined as

$$\begin{aligned} \sigma_{i,j} &= e^{-|s_i - s_j|/2} \quad \text{for } s_1 < s_2 < ... < s_p \\ \text{and } s_j - s_{j-1} &\sim U(0.5, 1) \quad \text{for } j = 2, ..., p. \end{aligned}$$

Then we set $\Omega = \Sigma^{-1}$. Our precision matrix thus constructed is a tridiagonal matrix. In the case of the nearest neighbor network, we generate 25 points randomly on the unit square $(0,1) \times (0,1)$. We calculate all $p(p-1)/2$ pairwise distances and find m nearest neighbors of each point in terms of this distance. The nearest neighbor network is obtained by linking any two points that are m-nearest neighbors of each other. The integer $m = 2$ is chosen for our study.

To find the solutions to the *lasso* regressions in 1 we use a proximal gradient algorithm using a constant step size, denoted by t, of 0.0001. The algorithm is as follows:

$$\beta^{k+1} = \beta^k - t\nabla\frac{1}{2}\|X_i - \sum_{j\neq i}\beta_{ji}^k X_j\|_2^2$$
$$\beta^{k+1} = S_{t\lambda}(\beta^{k+1})$$

where $S_\lambda(.)$ denotes the soft-thresholding operator. The next step was to estimate the optimal λ for each graph type. Let

$$F(\hat{\Omega}) = \frac{\|\Omega - \hat{\Omega}\|_F^2}{\|\Omega\|_F^2}$$
$$FP(\hat{\Omega}) = \frac{\sum_{i,j} I(\omega_{ij} = 0, \hat{\omega}_{ij} \neq 0)}{\sum_{i,j} I(\omega_{ij} = 0)}$$
$$FN(\hat{\Omega}) = \frac{\sum_{i,j} I(\omega_{ij} \neq 0, \hat{\omega}_{ij} = 0)}{\sum_{i,j} I(\omega_{ij} \neq 0)}$$

We did this by comparing F, FP and FN at various λ between 0 and 114. In general, F and FP decreased as λ increased while the FN increased with λ and therefore there usually wasn't any ideal λ all these measures pointed to. In the case of the chain graph $\lambda = 55$ worked well because both the FN and FP were equal to 0 as Figure **??** illustrates. Also, at this value of λ, we were able to recover the correct sparsity pattern as shown in Figure **??**. For the other two graphs we chose λ to minimize FN given that the FP was below 0.1. For the nearest neighbor network, this happens at around $\lambda = 25$.

Finally let

$$F_K(\hat{\Omega}) = \frac{1}{K}\sum_{k=1}^{K}\frac{\|\Omega^k - \hat{\Omega}^k\|_F^2}{\|\Omega^k\|_F^2}$$

$$FP_K(\hat{\Omega}) = \frac{1}{K}\sum_{k=1}^{K}\frac{\sum_{i,j} I(\omega_{ij}^k = 0, \hat{\omega}_{ij}^k \neq 0)}{\sum_{i,j} I(\omega_{ij}^k = 0)}$$

$$FN_K(\hat{\Omega}) = \frac{1}{K}\sum_{k=1}^{K}\frac{\sum_{i,j} I(\omega_{ij}^k \neq 0, \hat{\omega}_{ij}^k = 0)}{\sum_{i,j} I(\omega_{ij}^k \neq 0)}$$

We set $K = 10$ and replicate the numerical simulations tens times and calculate F_K, FP_K and FN_K for our estimate. A disadvantage of this method is that all the *lasso* regressions are independent with a consequence being the loss of positive definiteness of $\hat{\Omega}$. To calculate a positive definite estimate for Ω, denoted by $\hat{\Omega}^{pd}$, we can follow steps 1 and 2 as outlined below.

1. If $i \neq j$, set $\Omega_{ij} = \max\{\Omega_{ij}, \Omega_{ji}\}$.
2. While $\Lambda_{min}(\Omega) < 0$, then $\Omega = \Omega - \Lambda_{min}(\Omega) * I$

Let $F_K^{pd} = F_K(\hat{\Omega}^{pd})$ where $\hat{\Omega}^{pd}$ is positive definite. As Table **??** shows most of the results were consistent with our initial calculations. For the chain graph, the prediction of the sparsity pattern tends to be very accurate, but the Frobenius Norm Loss can be higher than for other graph types. In the case of the Nearest Neighbor Network, the Frobenius Norm loss is smaller, but the False Negative rate is much higher indicating that this method doesn't work as well in predicting the sparsity pattern for these type of graphs.

	Chain	Nearest Neighbor	Barabasi
F_{10}^{pd}	0.7137008	0.3130917	xxx
F_{10}	0.6640735	0.3137885	xxx
FP_{10}	0.0001897533	0.07741935	xxx
FN_{10}	0	0.6784593	xxx

Table 1: A comparison of the Meinshausen-Buhlmann method for Chain, Nearest Neighbor and Barabasi graphs