THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

ST443

Part I: Real World Data

Part II: Estimation of Graphical Models Using Lasso-Related Approaches

Malika Malik

Department of Statistics
London School of Economics And Political Science
December 2017

1 Real World Data

There has been an exponential increase in the output of news generated in our society and most of the information does not reach the masses as originally intended. What allows a certain article to arrive on our computer screen? What makes some news travel very fast and to a lot of people while other news is soon dismissed? This project tries to find the best way to predict popularity of news published online based on news article characteristics. We compared different models to predict news popularity and found that Random Forests were the best at predicting whether a news article reaches a certain threshold of popularity. It is important to note that once an article has surpassed a certain threshold its growth is exponential.

To do this we use a dataset from The UCI Machine Learning Repository donated by [1]. This dataset includes almost 40,000 articles published on the website Mashable with details on 61 characteristics, including the number of shares they achieved. Other characteristics include details about the topic and content of the news, how it is written, what style it is written in and whether it includes other links, images and videos. The variable that we are going to use as our dependent variable can be seen as continuous (number of shares) or as binary (popular/non-popular). We will briefly discuss the case for y continuous, but we will focus on the classification case.

We want to find a threshold that defines whether the variable is popular or not. We will be using the threshold of 1,400 shares as this is the median and will yield a balanced classification problem. Although choosing the median as our measure may not seem reasonable given the maximum is 843,300, it is important to note that there is an exponential growth as one share leads to a large amount of people that could share it and so on. This means that once a news has reached a threshold it will keep growing extremely fast, so after 1,400 shares it can be considered popular.

1.1 Regression Method

To predict the number of shares achieved by a news article, we perform variable selection, using ridge regression and the lasso approach, to improve our prediction model. For that we analyse the standardised ridge regression coecients as a function of λ and fractional deviation. From this we conclude that λ is close to the value of 0 and there are no constraints on the coefficients. As λ exponentially increases, the coefficients tend towards 0. Ridge regression shrinks the coefficients toward 0, however it does not perform any variable selection.

We fitted the lasso model and performed cross-validation to obtain a lower value for the MSE compared to the test MSE of the null model, whilst being relative to the test MSE of ridge regression with λ (chosen by cross-validation).

We observe that 4 out of the 19 coecient estimates are exactly zero. Hence, the lasso model with λ chosen by cross-validation contains 15 variables. This project is going to focus on the classification problem to improve our prediction since it seems to be a more interesting problem.

1.2 Classification - Logit, LDA, QDA and KNN Methods

First, we start with logistic regression for the popular classification problem. We apply the trained models to the testing data and achieve a misclassification error rate of 0.48 for logistic regression, 0.36 for the linear discriminant analysis (LDA), and 0.44 for the quadratic discriminant analysis (QDA).

For k-nearest neighbours, we find that the optimal K that minimises the training misclassification error rate is 57. This gives us a test misclassification rate of 0.41.

The summary of the misclassification error rates of the different methods we attempted are illustrated in Table 1.

	Logit	LDA	QDA	KNN
Rate	0.483202	0.3621941	0.4450436	0.4061593

Table 1: Missclassification Error Rates

We plotted the ROC curves for the four different classifiers in Figure 1.

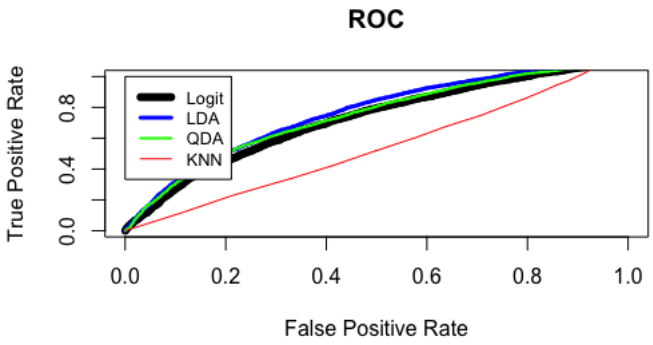


Figure 1: ROC Curve for Logit, LDA, QDA and KNN

The areas under the ROC curve (AUROC) for logistic regression, LDA, QDA, kNN with optimal K value are 0.65, 0.69, 0.67, 0.46 respectively. We therefore conclude that LDA performs the best among the four methods.

1.3 Tree-Based Model

We now attempt to improve the predictions using tree-based methods for classification. Once the tree has been pruned we can observe the structure in Figure 2.

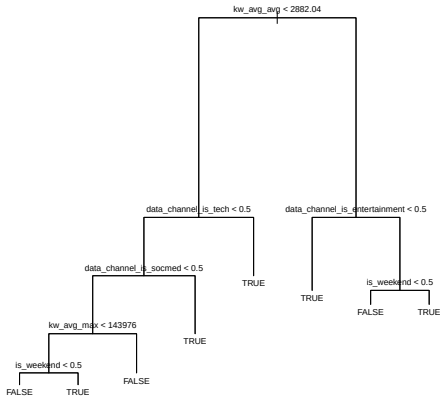


Figure 2: Pruned Tree Model

The misclassification error rate for the unpruned tree is 0.37. The misclassification error rate for the pruned tree is 0.37, which while not an improvement it does vastly simplify the interpretation.

To further improve our model we used a Random Forest which decreased our test misclassification error rate to 0.33. This model will help to predict whether a news article is popular or not with an accuracy of 67% before publishing it. It is also interesting to see the importance of the variables in the random forest model, showing us which ones help predict the popularity of the articles. This plot is in the appendix.

We now try creating more categories and see how well the Random Forest method performs.

We choose the three class classification case. If we add a fourth category, some of the defined classes do not exhibit different behaviours. In particular, if we create a higher threshold called "very very popular" this behaves similarly to category "very popular", so we instead combine them to one category. Thus our final categories are "Unpopular", "Popular" and "Very popular" (defined with quantiles at 34% and 66%). Running the Random Forest on these categories results in a misclassification error rate of 0.46 as shown in table 2.

	(Unpopular]	(Popular]	(Very Popular]
(Unpopular]	3717	1438	1175
(Popular]	256	303	257
(Very Popular]	622	671	1205

Table 2: Confusion Matrix for Three Class Classification, quantiles 34% and 66%

Defining the thresholds as the 50% and 98% quantiles we obtain a better misclassification error rate of 0.35, which is an improvement but it is unbalanced.

	(Unpopular]	(Popular]	(Very Popular]
(Unpopular]	3403	1631	47
(Popular]	1585	2830	147
(Very Popular]	0	0	1

Table 3: Confusion Matrix for Three Class Classification, quantiles 50% and 98%

We observe in Table 3 that there are problems in identifying the third category. This appears to indicate that after crossing the threshold of getting 1,400 shares, the 50% quantile, there is no structural differences on the recordable characteristics of those articles- the increase in shares is as a result of the exponential growth discussed previously.

2 Estimation of Graphical Models Using Lasso-Related Approaches

2.1 Gaussian Graphical Models

A Gaussian Graphical Model (GGM) is a graphical model that illustrates the statistical dependencies between variables of interest. For p variables this is a graph consisting of p nodes, and edges connecting a subset of the nodes. The edges describe the conditional dependence structure of the p variables such that there is an edge between two different variables X_j and X_l if they are conditionally dependent [2]. A graph with vertices $V = \{1, \ldots, p\}$, covariance structure

$$c_{jl} = Cov(X_j, X_l | X_k, 1 \leq k \leq p, k \neq j, l) \tag{1}$$

and edge set

$$E = \{(j, l) : c_{jl} \neq 0, 1 \leq j, l \leq p, j \neq l\} \tag{2}$$

is denoted $G = (V, E)$.

2.2 Graphical Lasso Approach

The graphical lasso approach attempts to learn the structure of a GGM [2]. It aims to maximise the log likelihood function for the data [2], subject to a l_1 penalty on the inverse covariance matrix, denoted by $\boldsymbol{\Theta}$.

Suppose we want to learn a GGM with dimension p, i.e. with p nodes, from a p-variate multivariate normal distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^{p\text{x}1}$, and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{pxp}$. As $\boldsymbol{\Sigma}_{i,j} = 0$, for some $i \neq j$, if and only if X_i and X_j are conditionally independent given all other observations, X_k, $k = \{1, ..., n\} \backslash \{i, j\}$ [2], it follows that the GGM is produced from the non-zero entries of $\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Theta}$. Thus, the edges of the graph, E, can be found using;

$$E = \{(i, j) : \Theta_{i,j} \neq 0, 1 \leq i, j \leq p, i \neq j\},$$

where $\Theta_{i,j}$ is the (i, j)th component of the matrix $\boldsymbol{\Theta}$.

Using samples $(x_1, \ldots, x_n)$ from the p-variate multivariate normal distribution defined above, the graphical lasso can used to produce a sparse estimate for $\boldsymbol{\Theta}$, denoted $\hat{\boldsymbol{\Theta}}$ by considering an optimisation problem, equivalent to maximising the penalised log-likelihood [4]:

$$\underset{\boldsymbol{\Theta}}{\text{minimise}} \quad -\log(\det(\boldsymbol{\Theta})) + \text{trace}(\boldsymbol{S}\boldsymbol{\Theta}) + \lambda \left\|\Theta\right\|_1 \tag{3}$$

$$\text{subject to} \quad \boldsymbol{\Theta} \succeq \boldsymbol{0}, \tag{4}$$

where $\boldsymbol{S} = \frac{1}{n-1} \sum_{i=0}^{n} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T$, is the sample covariance matrix and λ is a l_1 shrinkage parameter.

The edge set of $\hat{\boldsymbol{\Theta}}$ is

$$\hat{E}_3 = \{(i, j) : \hat{\Theta}_{i,j} \neq 0, 1 \leq i, j \leq p, i \neq j\},$$

determined for a certain value of λ which minimises the optimisation problem. We use $\hat{\boldsymbol{\Theta}}$ to obtain an estimate of the true edge set.

Since Equation 3 is not convex [3], a coordinate descent algorithm is required. This algorithm updates $\boldsymbol{\Sigma}$ by row and column, without loss of generality. In this,

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\sigma}_{12} \\ \boldsymbol{\sigma}_{12}^T & \boldsymbol{\sigma}_{22} \end{pmatrix}, \boldsymbol{S} = \begin{pmatrix} \boldsymbol{S}_{11} & \boldsymbol{s}_{12} \\ \boldsymbol{s}_{12}^T & \boldsymbol{s}_{22} \end{pmatrix},$$

[3] where $\boldsymbol{\Sigma}_{11}$ is the covariance matrix of the first $p-1$ variables and $\boldsymbol{S}_{11}$ is the sample covariance matrix of the first $p-1$ variables, $\boldsymbol{\sigma}_{12}$ and $\boldsymbol{s}_{12}$ are the covariance and sample covariance between the first $p-1$ variables and the last variable respectively, and $\boldsymbol{\sigma}_{22}$ and $\boldsymbol{s}_{22}$ are the variance and sample variance of the last variable, respectively [3]. These formulas are then manipulated to obtain $\hat{\boldsymbol{\Theta}}$, at which the algorithm converges.

2.3 Node-Wise Lasso Approach

Node-wise lasso begins with node-wise regression. For each node $j \in V$, regress the node X_j on the remaining X_l, $l \in V, l \neq j$:

$$X_j = \sum_{1 \leq l \leq p, l \neq j} \beta_{jl} X_l + \epsilon_{jl}.$$

To obtain a sparse estimate for the coefficients β_{jl}, and thus obtain an estimate of the true edge set, we apply the lasso approach to the above with a tuning parameter λ. The lasso estimate of β_{jl} is $\hat{\beta_{jl}}$, with a chosen tuning parameter λ. By this method, if $\hat{\beta_{jl}} \neq 0$, then we say that nodes j and l, corresponding to the variables X_j and X_l respectively, are estimated to be connected in the GGM.

The rules for the estimated edge set of the GGM considered when studying the node-wise lasso approach are defined as follows:

$$\hat{E_1} = \{(j,l) : \text{both } \beta_{jl} \text{ and } \beta_{lj} \text{ are nonzero}, 1 \leq j, l \leq p, j \neq l\}, \tag{5}$$

$$\hat{E_2} = \{(j,l) : \text{either } \beta_{jl} \text{ or } \beta_{lj} \text{ are nonzero}, 1 \leq j, l \leq p, j \neq l\}. \tag{6}$$

2.4 Aim of the Simulation

The aim of this simulation is to compare the sample performance of two different approaches, node-wise lasso and graphical lasso, in recovering the edge set of a Gaussian graphical model, for differing values of p, the dimension, and n, the sample size.

2.5 Setting Up the Simulation

Let $\boldsymbol{\Theta} = \boldsymbol{B} + \delta \boldsymbol{I}_p \in \mathbb{R}^{p \text{x} p}$ where $\boldsymbol{I}_p$ is the identity matrix. $\boldsymbol{B}$ is a symmetric matrix with each off-diagonal entry taking values 0.5 and 0 with probabilities 0.1 and 0.9 respectively. $\delta > 0$ is such that $\boldsymbol{\Theta}$ is positive definite. As $\boldsymbol{\Theta}$ is a symmetric matrix, it is positive definite if and only if all of its eigenvalues are greater than zero [5]. Finally $\boldsymbol{\Theta}$ is standardised to have unit diagonals, and the sparsity pattern in $\boldsymbol{\Theta}$ corresponds to the true edge set. Our aim is to estimate the true edge set of $\boldsymbol{\Theta}$ using node-wise lasso and graphical lasso. Then compare the two approaches by plotting ROC curves for each combination of p and n.

We generated the $\boldsymbol{B}$ matrix by sampling $\frac{p(p-1)}{2}$ elements from the set $\{0, 0.5\}$ with probabilities $\{0.9, 0.1\}$ respectively and used these values to fill the upper triangle of an empty matrix, mirroring this over the diagonal to ensure symmetry. To determine a valid δ, we considered $\delta' \in \{0, 0.1, \ldots, 9.9, 10\}$. We chose to allow δ to vary freely as we are focusing on the impact different values of p and n have on the effectiveness of recovering the true edge set, this will be discussed further in Section 2.10. For each of these possible δ' we calculated the eigenvalues of each $\boldsymbol{\Theta} = \boldsymbol{B} + \delta' \boldsymbol{I}_p$ and returned any values of δ' for which the eigenvalues were all greater than zero. We then sampled one value from the set of valid δ' values to generate $\boldsymbol{\Theta}$ before standardising to obtain the final matrix.

Using the above, it was then possible to generate n samples from a multivariate Gaussian distribution with zero mean and the covariance matrix $\mathbf{\Sigma} = \mathbf{\Theta}^{-1}$.

We chose to focus on the values $p \in \{30, 60, 90\}$ and $n \in \{100, 250, 500\}$.

2.6 Implementing the Node-Wise Lasso Approach

When implementing this approach we chose $\lambda \in \{0, 1\}$ increasing in increments of 0.005. For each combination of p and n:

1. Use the n samples generated to implement the node-wise lasso for different values of tuning parameter λ using the `R` function `glmnet`.
2. Fill a p x p matrix with the estimated lasso coefficients $\hat{\beta}_{jl}$, as defined in Section 2.3.
3. Compare these estimated coefficients with the true edge set, to obtain estimates for $\hat{E_1}$ and $\hat{E_2}$, as defined in (5) and (6).
4. Calculate the true positive rate (TPR_λ) and false positive rate (FPR_λ) in terms of edges correctly identified, defining a positive outcome as two nodes being connected by an edge.
5. Plot TPR_λ vs FPR_λ to produce a ROC curve and calculate the area under the curve (AUROC).

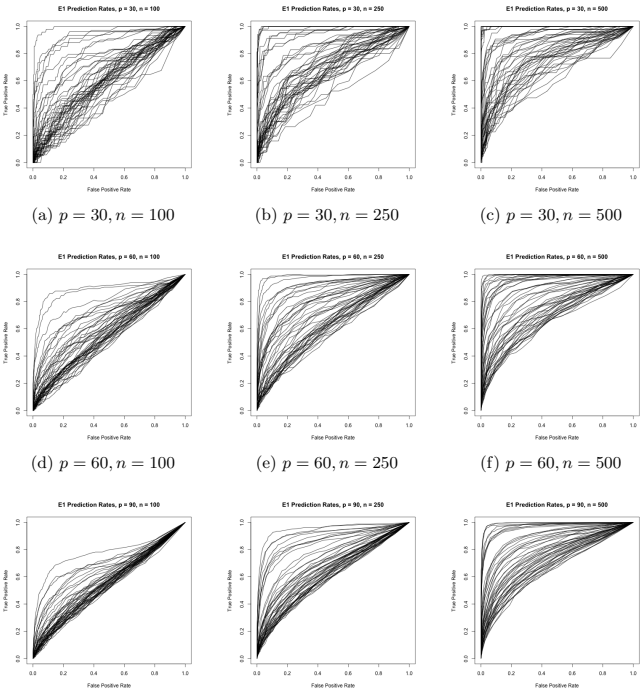
This procedure was repeated 50 times for each combination of p and n, with all the results plotted onto one ROC graph. The mean and standard error of the AUROC were calculated and used as the basis of our comparison between the methods and between the different values of p and n.

The results from running the node-wise lasso approach to estimate the prediction rates for the edge set $\hat{E_1}$ and $\hat{E_2}$ are displayed in Figures 3 and 5 respectively. The graphs show the ROC curves for the 50 repetitions of each combination of p and n over a fine grid of λ values. We found the difference between the results for 3 and 5 to be negligible, therefore we will discuss the results of both together.

In Figures 3 and 5, we can see that for each value of p, as n increases the area under the ROC curve tends more towards 1, which is the ideal ROC curve. This is illustrated clearly in Figures 4 and 6 which show that for a constant value of p, as n increases the area under the ROC curve increases significantly.

Using Figures 4 and 6 to compare the mean area under the ROC curve plus/minus two standard errors we can see that there is no significant difference in the AUROC values between $p = 30$ and $p = 60$ and between $p = 60$ and $p = 90$ - there is overlap in the confidence bands in each case. However, comparing $p = 30$ and $p = 90$ we can see in Figures 4 and 6 that there is a significant difference between the areas under the ROC curve for $n = 100$, but that as n increases the effect of p lessens and there is no significant difference. We believe that if we were to run the simulation for a larger range of p values that this trend would have been exaggerated and that we would see a significant decrease in the area under the ROC curves as p increases over larger values of n. If it were computationally viable, we would have liked to have explored this trend over a larger range of p values to solidify our understanding. Intuitively, an increase in performance for smaller p and constant n makes sense as we are using regression and there are fewer predictors in the model.

If we were to repeat the simulation, we would like to experiment with the $\mathbf{\Theta}$ matrix and see if this results in a difference in performance between E_2 and E_2. In particular we would like to generate a less sparse matrix B and see what effect this has.



(a) $p = 30, n = 100$ (b) $p = 30, n = 250$ (c) $p = 30, n = 500$

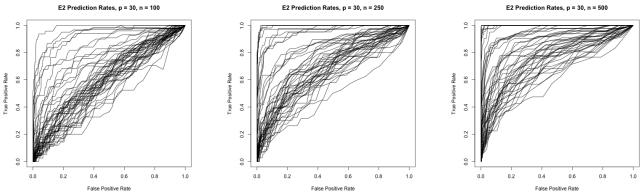
(d) $p = 60, n = 100$ (e) $p = 60, n = 250$ (f) $p = 60, n = 500$

(g) $p = 90, n = 100$ (h) $p = 90, n = 250$ (i) $p = 90, n = 500$

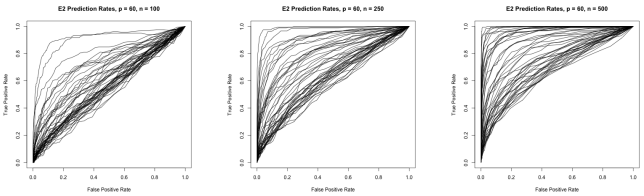
Figure 3: Prediction Rates for $\hat{E}_1$

E1_AUROC_plot.png

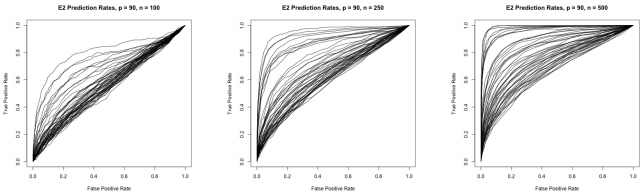
Figure 4: Mean and standard error for $\hat{E}_1$ AUROC



(a) $p = 30, n = 100$ (b) $p = 30, n = 250$ (c) $p = 30, n = 500$



(d) $p = 60, n = 100$ (e) $p = 60, n = 250$ (f) $p = 60, n = 500$



(g) $p = 90, n = 100$ (h) $p = 90, n = 250$ (i) $p = 90, n = 500$

Figure 5: Prediction Rates for $\hat{E}_2$

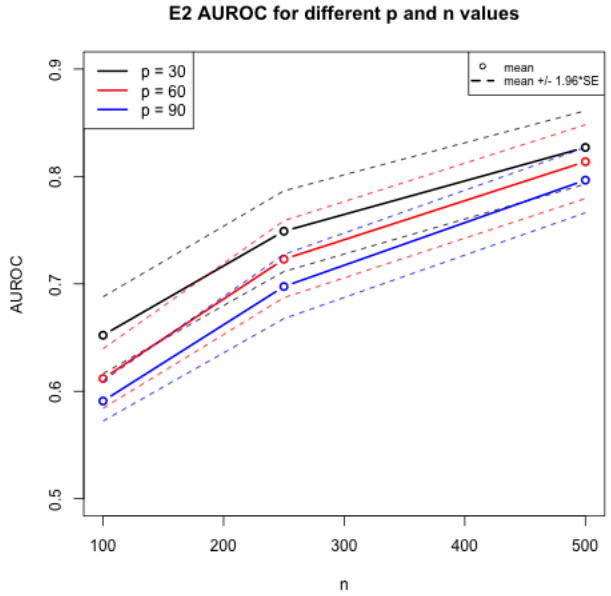
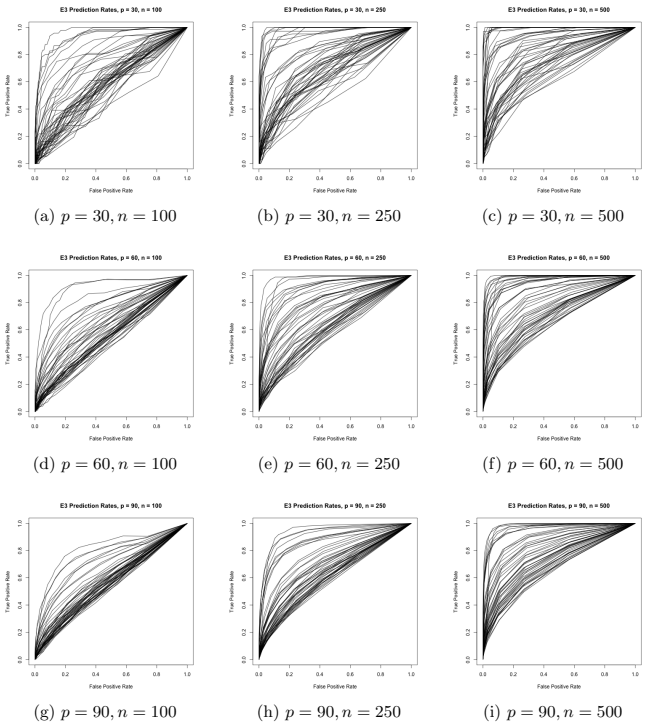


Figure 6: Mean and standard error for $\hat{E}_2$ AUROC



(a) $p = 30, n = 100$ (b) $p = 30, n = 250$ (c) $p = 30, n = 500$

(d) $p = 60, n = 100$ (e) $p = 60, n = 250$ (f) $p = 60, n = 500$

(g) $p = 90, n = 100$ (h) $p = 90, n = 250$ (i) $p = 90, n = 500$

Figure 7: Prediction Rates for $\hat{E}_3$

2.7 Implementing the Graphical Lasso Approach

When implementing this approach we chose $\lambda \in \{0, 10\}$ increasing in increments of 0.025. For each combination of p and n:

1. Use the n samples generated to determine an estimate for the covariance matrix $\mathbf{\Sigma}$.
2. Implement graphical lasso using the `glasso` function in `R` with λ in the range as defined above to obtain an estimate $\hat{\mathbf{\Theta}}$.
3. Calculate $\hat{E}_3$ by comparing $\hat{\mathbf{\Theta}}$ to $\mathbf{\Theta}$.
4. Calculate the true positive rate (TPR_λ) and false positive rate (FPR_λ) in terms of edges correctly identified, defining a positive outcome as two nodes being connected by an edge.
5. Plot TPR_λ vs FPR_λ to produce a ROC curve and calculate the area under the curve (AUROC).

This procedure was repeated 50 times for each combination of p and n, with all the results plotted onto one ROC graph. The mean and standard error of the AUROC were calculated and used as the basis of our comparison between the methods and between the different values of p and n.

Figure 7 shows the results of applying the graphical lasso approach and estimated edge set as $\hat{E}_3$ for predicting the true edge set. The graphs in Figure 7 show the ROC curves for 50 repetitions of each combination of p and n over a grid of $\lambda \in [0, 1]$. By way of comparison, we can study Figure 8 which illustrates the mean area under the ROC curves, with the standard errors used to draw confidence bands onto the plot.

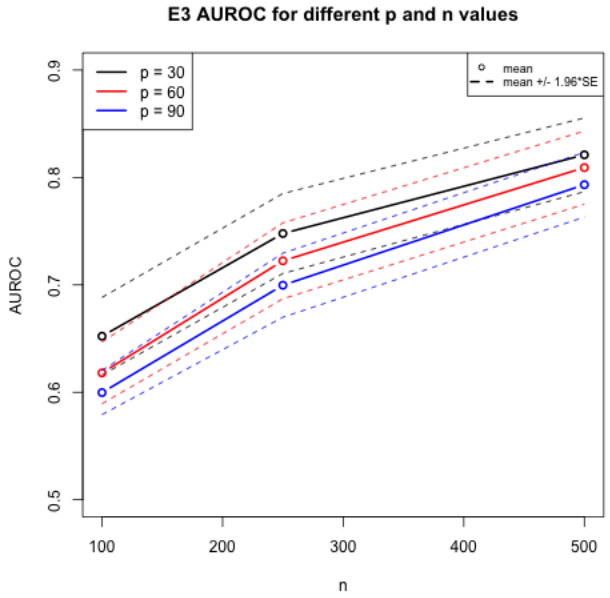


Figure 8: Mean and standard error for $\hat{E}_3$ AUROC

From the plots in Figure 7, it can be seen that as the value of p decreases, the area under the ROC curve tends towards 1, which corresponds to the perfect ROC curve. For a constant p, the area under the ROC curve also tends to 1 as the value of n increases. In order to solidify the trend observed, we would have liked to have tried a broader range of values for p and n, however as this was a computationally intensive process we were unable to do this here.

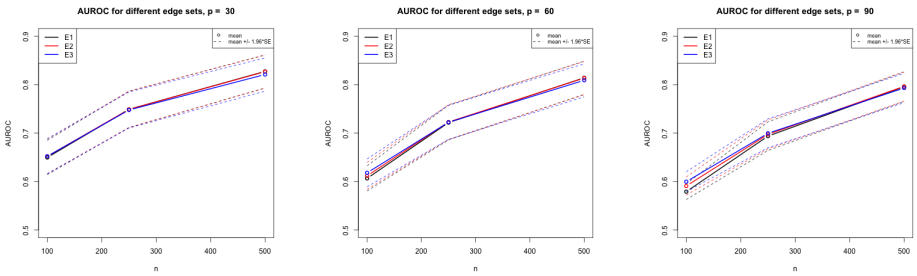
Further analysis of this approach can be conducted through the evaluation of Figure 8. For the pairs $p = 30$ and $p = 60$, and $p = 60$ and $p = 90$ the confidence bands overlap for each value of n therefore we can conclude that there is no significant difference in the ability of graphical lasso to predict the true edge set. However, when comparing the results between $p = 30$ and $p = 90$ within a constant n, there is a significant difference, and it can be seen that as p increases, the area under the curve decreases. However, as n increases, the importance of p decreases since the area under the ROC curve for all three values of p at $n = 500$ tend to a similar point.

2.8 Comparing the Two Approaches

For our chosen values of p and n we observe no significant differences between the prediction rates of $\hat{E}_1$, $\hat{E}_2$, and $\hat{E}_3$.

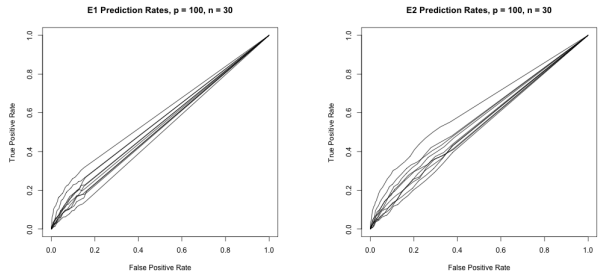
The graphs in Figure 9 plot the mean of the area under the ROC curves for each edge set, $\hat{E}_1$, $\hat{E}_2$, and $\hat{E}_3$ for a different value of p along with the confidence bands.

Variation between the mean areas under the ROC curves of the three predicted edge sets occurs at $n = 100$, and increases as p increases. With the mean area under the ROC curve for $\hat{E}_1$ taking the lower value, $\hat{E}_2$ taking the middle value and $\hat{E}_3$ taking the higher value as p increases. Since this difference in the prediction rates of the three different edge sets becomes more apparent as p increases, this assumption may be proved if we tried this comparison for larger values of p and observing whether the trend continues or not. However, due to the computational time for the prediction rates of each edge set for larger values of p, this was not deemed possible.



(a) $p = 30$ (b) $p = 60$ (c) $p = 90$

Figure 9: Mean and standard error for different edge sets, $p = 30, 60, 90$



(a) $p = 100, n = 30$ (b) $p = 100, n = 30$

Figure 10: High dimensional case for node-wise lasso

2.9 The High Dimensional Case

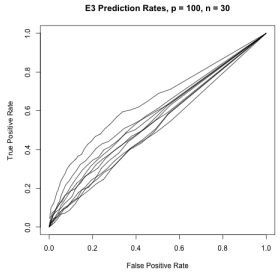
We decided to investigate the behaviour of the two methods for $p > n$.

We applied both node-wise lasso and graphical lasso for $p = 100$ and $n = 30$, and plotted the ROC curves as defined above for the prediction rates of $\hat{E_1}$, $\hat{E_2}$ and $\hat{E_3}$ for 10 repetitions over a grid of $\lambda \in [0, 1]$. Here we chose to do only 10 repetitions as the computational time was too slow to run for 50.

Figure 10 shows that the node-wise lasso approach does not predict the true edge set well using either E_1 or E_2 when $p > n$. In particular, although it predicts the TPR and FPR to be 1, in both cases, for $\lambda = 0$, as λ approaches 0 both the true positive and false positive rates are ~ 0.2 for E_1, and ~ 0.4 for E_2. This indicates that E_1 performs worse than E_2. The lack of variability in the true positive and false positive rates beyond these values is the reason for the straight lines in Figure 10 connecting the point (1,1) to the rest of the plot.

Figure 11 shows that the graphical lasso approach does not predict the true edge set particularly well when $p > n$. Similar to node wise lasso, it predicts the TPR and FPR to be 1, in both cases, for $\lambda = 0$ but as λ approaches 0 both the true positive and false positive rates stop increasing. However we can see in Figure 11 that using graphical lasso it would be possible to obtain a true positive rate of ~ 0.6 and a false positive rate of ~ 0.4, which is an improvement over node-wise lasso thus we conclude that graphical lasso obtains better results for $p > n$.

We would like to investigate this further using a wider range of p and n with $p > n$ and determine if there is a ratio of $p : n$ at which graphical lasso begins to perform better.



(a) $p = 100, n = 30$

Figure 11: High dimensional case for graphical lasso

2.10 Final Discussion

In the simulation, we considered that the true value of δ could fall into the range [0,10]. We started off considering a narrower range of possible δ values, [0, 5], but realised that for larger values of p this led to very few valid values being used in the simulation. As previously discussed we wanted δ to vary freely so as to have minimal impact on the results and allow us to focus on the effect of p and n. Were we to repeat the simulation we would like to either, more carefully consider the range of δ and its effect on the simulation, or else widen the range of δ and perform more iterations, computational time prevented us from doing the latter.

From the results we have gained, it is hard to spot a trend when only considering three p values and three values of n. However, computational time and power prevented us from being able to consider a larger range of p and a larger range of n values. This would have created more combinations, and more results to compare and contrast the two approaches.

As a starting point for this simulation, we considered small values of p, for example $p = 3$. We soon realised that because of the values and the sampling probabilities we used to generate $\boldsymbol{B}$, there was a high probability of generating the identity matrix.

References

[1] Fernandes, K., Vinagre, P. and Cortez, P. (2015). ***A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News***. Proceedings of the 17th EPIA 2015 - Portuguese Conference on Artificial Intelligence, September, Coimbra, Portugal.

[2] Grechkin, M., Fazel, M., Witten, D., Lee, S.-I. (2015). ***Pathway Graphical Lasso***. Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence, 2015, p. 2617–2623.

[3] Wang, H. (2013). ***Coordinate Descent Algorithm for Covariance Graphical Lasso***. Stat Comput 6, p. 1–9.

[4] Friedman, J., Hastie, T., Tibshirani, R. (2007). ***Sparse Inverse Covariance Estimation with the Lasso***. Stanford University. p. 1-7.

[5] Strang, G. (2011). ***Symmetric Matrices and Positive Definiteness***. Linear Algebra, MITOpenCourseWare, Massachusetts Institute of Technology.
Available from: *https://ocw.mit.edu/courses/mathematics/18-06sc-linear-algebra-fall-2011/positive-definite-matrices-and-applications/symmetric-matrices-and-positive-definiteness/MIT18_06SCF11_Ses3.1sum.pdf*. Date accessed: 01/12/2017.

A Appendix

A.1 Summary Statistics

	unclass(shares_sum)
Min.	1.00
1st Qu.	946.00
Median	1400.00
Mean	339
3rd Qu.	2800.00
Max.	843300.00

Table 4: Summary Statistics for the response variable (number of shares)

A.2 Appendix: Regression Model: Ridge Regression

We fit a ridge regression model on the training set, and evaluate its MSE on the test set, using $\lambda = 4$. The test MSE, observed, is 117,840,111. However,if we fit a ridge regression model on the training set, with just an intercept, we would have predicted each test observation using the mean. Hence, we get the computed test set MSE of 120,791,031. Therefore, fitting a ridge regression model with $\lambda = 4$, achieves a lower test MSE, than fitting the model with only an intercept.

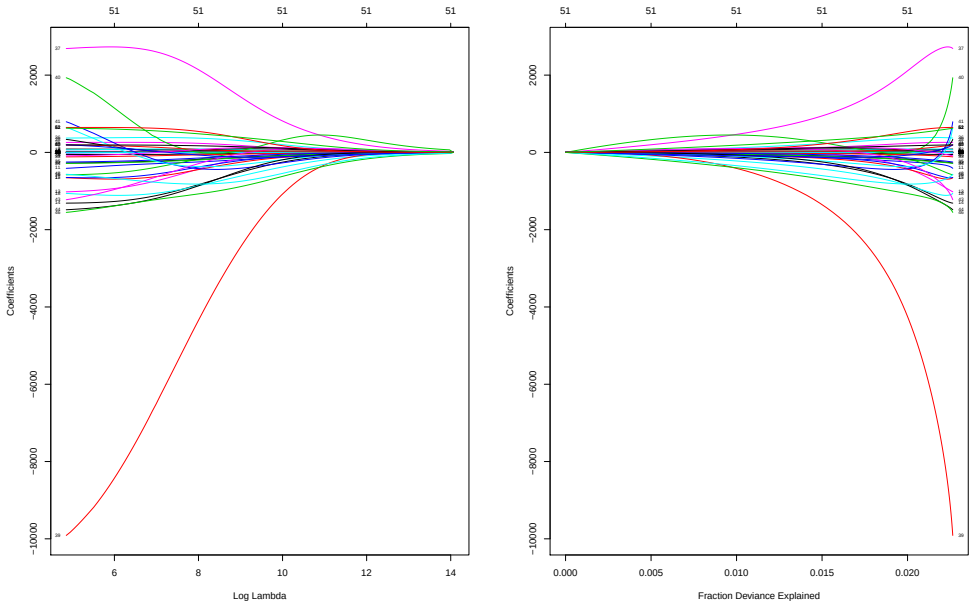


Figure 12: The standardised ridge regression coefficients displayed as a function of the tuning parameter: λ and fractional deviation, respectively

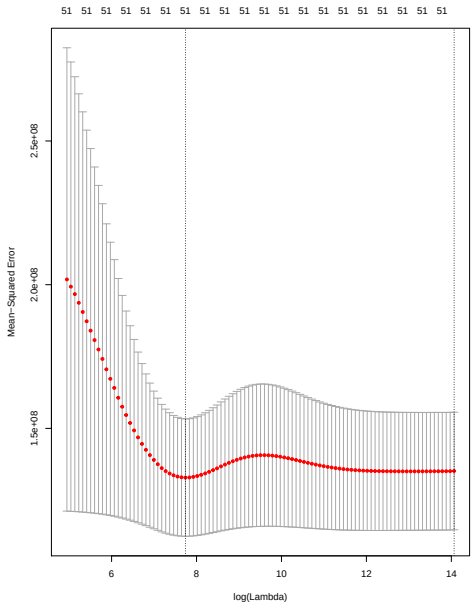


Figure 13: Test error as a function of λ

By performing cross-validation we chose the tuning parameter $\lambda = 1966.168$. Then the MSE associated with this value of $\lambda = 118123857$. This is an improvement, over the test MSE that we obtained, using $\lambda = 4$.

We retted the ridge regression model on the full data set, using the value of λ chosen by cross-validation ($\lambda = 1966.168$). As expected, none of the coefficients are zero - ridge regression does not perform variable selection.

A.3 Appendix: Regression Model: Lasso Model

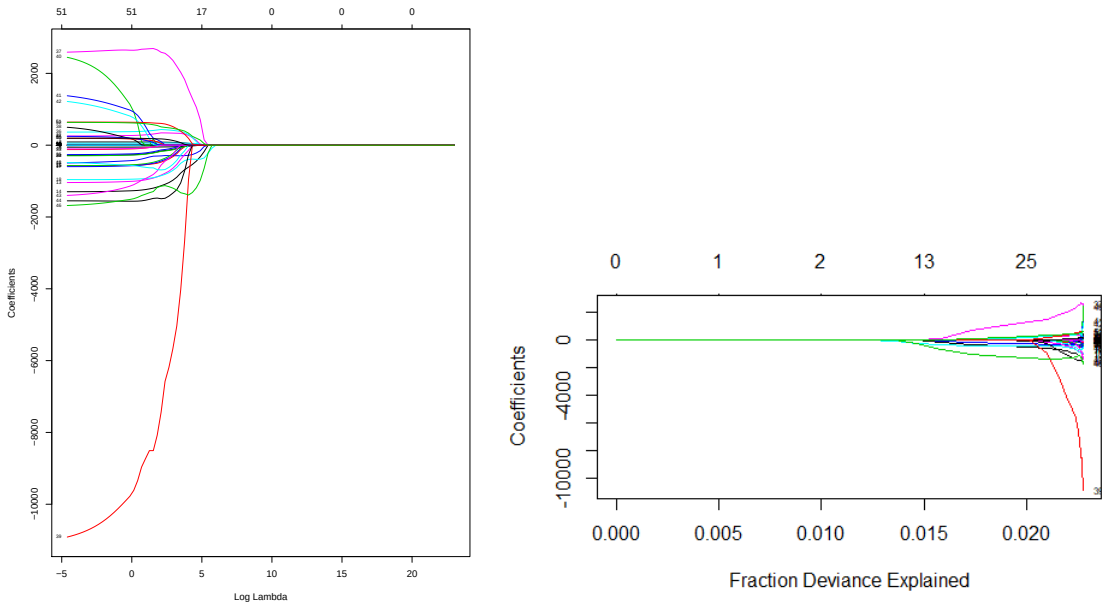


Figure 14: The standardised lasso coefficients displayed as a function of the tuning parameter: λ and fractional deviation, respectively

A.4 Appendix: Logistic Regression

	FALSE	TRUE
FALSE	533	205
TRUE	4455	4451

Table 5: The confusion matrix for the logistic regression model

A.5 Appendix: Linear Discriminant Analysis (LDA)

	FALSE	TRUE
FALSE	3280	1785
TRUE	1708	2871

Table 6: The confusion matrix for the LDA model

A.6 Appendix: Quadratic Discriminant Analysis (QDA)

	FALSE	TRUE
FALSE	4771	4075
TRUE	217	581

Table 7: The confusion matrix for the QDA model

A.7 Appendix: KNN

For the training data, we plot the misclassification error rate vs $1/k$ for KNN.

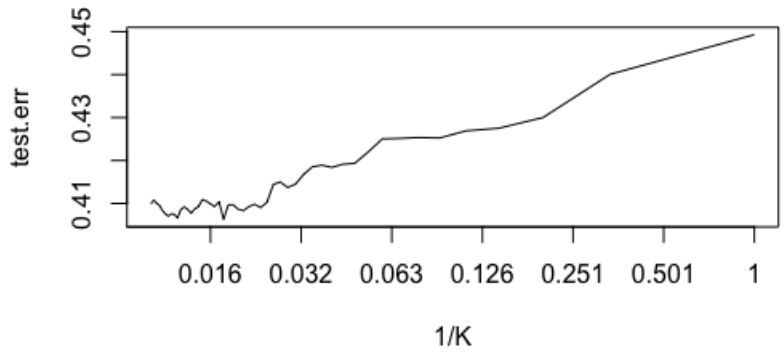


Figure 15: $1/K$ vs test error rate

The optimal K that minimises the test misclassification error rate is 57.

	FALSE	TRUE
FALSE	3197	2126
TRUE	1791	2530

Table 8: The confusion matrix for the KNN model

A.8 Appendix:Tree Model

	FALSE	TRUE
FALSE	2643	1245
TRUE	2345	3411

Table 9: The confusion matrix for the unpruned tree model

	FALSE	TRUE
FALSE	2889	1473
TRUE	2099	3183

Table 10: The confusion matrix for the pruned tree model

A.9 Appendix: Random Forest (Popular/Unpopular)

	FALSE	TRUE
FALSE	3285	1566
TRUE	1703	3090

Table 11: The confusion matrix for the Random Forest model

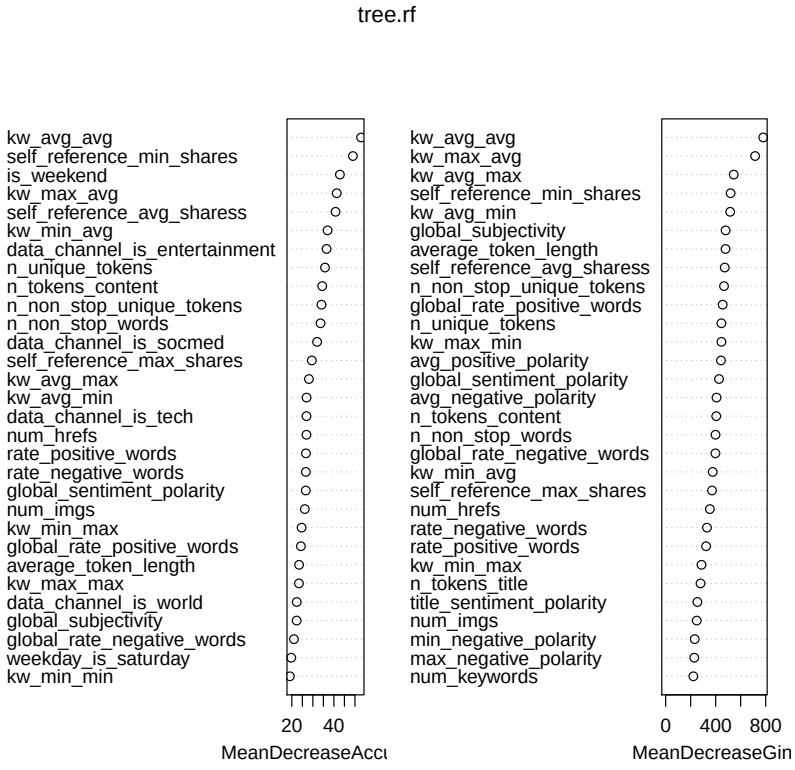


Figure 16: Variable importance plot for Random Forest model of news popularity data