CSES Working Paper Series

Elastic Net Regularized Network Estimation for Communities in a Tech Economy

Sirui Wang
Research Assistant
Center for the Study of Economy & Society
Cornell University

February 21, 2016
Elastic Net Regularized Network Estimation for Communities in a Tech Economy

Sirui Wang, sw494@cornell.edu

February 21, 2016

1. INTRODUCTION

The technology sector of New York City has rapidly grown in the past decade, drawing upon well-established firms in the area, as well as new talent from across the world. With the rise of social media platforms, we are able to study the structure of this emerging community by analyzing data from a large online group organized around tech-based startups. In accordance with the tendency for division of labor and specialization in modern societies[1], we study the apparent specialization of knowledge in the recent tech startup economy in New York City. This report aims to study the structure of the New York tech community by analyzing internet networking data from the large online tech-oriented group. Members of the online group can indicate personal interest topics on their profiles, as well join other special-interests groups, both of which provides a proxy of specialized knowledge that each member of the community possesses. We model the structure of specialized interests in a graphical network (a Markov Random Field) and estimate the interdependence of each interest topic or overlap group using node-wise elastic net regularized logistic regression. The result is a network that encodes the interdependence relationships between various specialized knowledge proxies (member interests or overlap groups) present in the current New York City tech-startup community.

Section 2 of this report will provide a general description of the data prepared to for the graphical model. Section 3 describes the context for the LASSO technique and its extension to the elastic net algorithm along with implementation used in this analysis. Section 4 shows the results of the methods in Section 3 performed on the data in Section 2. Section 5 provides a brief discussion on the implications of the results as well as areas of improvement.

2. DESCRIPTION OF DATA

All data collected for this project were accessed through the public API of the online platform that serves as the primary webpage for the large online New York tech group. Individuals can affiliate themselves with the group by creating a profile page and joining the group online. The group page can serve as a platform on which people publicize ideas and information as well as network with one another. Because of the nature of the technology startup business, mass circulation of information as well as networking is crucial to the growth of the community, which may be a reason that the online group has accumulated a large membership. We will assume that the members of this group provide a good representation of the members in overall technology community in New York City in the context of overall research project.

Associated with member profiles are lists of each members’ self-selected interests as well other groups the member has joined on the networking site. Both the interests and the groups can provide insight on the role the member plays in the overall community. The rest of the report on methodology will focus on member interests, but the analysis can also easily be applied to members’ groups.

As new members join the online group, they are given the opportunity to select topics that of interest to them; such topics include anything from “Cooking” to “Professional Networking” to “Web Development”. As of April 2015, the online group has a total membership of around 44,000 members. Of the 44,000 members, about 32,000 members had listed at least one topic of interest on his or her profile. Between the 32,000 members, they had listed around 15,000 unique interest topics.

In the following analysis, each member is considered as a data point, leading to \( n = 32,000 \) observations, and the interest topics are features describing each data, leading to 15,000 features if every interest topic is considered. Each data point is characterized by whether or not the member has indicated an interest topic. The elements that characterize the observations are hence binary values with -1 representing that the interest does not appear in the member’s profile, and a 1 representing that the interest does appear. By
this setup, the data set can be represented by a $32,000 \times 15,000$ matrix whose entries are either -1 or 1. A simple representation of the data in matrix form is given below:

Suppose Member A lists {“Software”, “Data”, “C++”, “Poker”}, Member B lists {“Technology”, “C++”, “Software”}, Member C lists {“Web Design”, “Poker”, “Music”} and Member D lists {“Poker”, “Music”, “Jazz”}. Then the data matrix representing this set of members and interests can be given as:

\[
X = \begin{bmatrix}
\text{Software} & \text{Data} & C++ & \text{Technology} & \text{Web Design} & \text{Poker} & \text{Music} & \text{Jazz} \\
A & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 \\
B & 1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 \\
C & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\
D & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 \\
\end{bmatrix}
\]

3. ESTIMATION OF THE GRAPHICAL MODEL

3.1 The Markov Random Field

Suppose our $n \times p$ data $X$ is set up as above with $X = [x_1, x_2, \ldots, x_p]$ where $x_i$ are the columns of the data matrix, each representing a unique interest topic for $i = 1, 2, \ldots, p$.

Let $X_i$ be a Bernoulli random variable, corresponding to the i-th interest topic, taking on the values 1 or -1, each with certain probability.

Then let $M = (X_1, X_2, \ldots, X_p)$ be a $p$-dimensional random vector with probability distribution that is the joint distribution of the random variables $X_1, X_2, \ldots, X_p$. We can think of each member in our sample as a realization of the random vector $M$, giving realizations $m^{(1)}, m^{(2)}, \ldots, m^{(n)}$.

In our sample of over 30,000 members, we can estimate the probability distribution along with covariance structures of the random vector $M$, and represent it in a Markov Random Field, a graph $G(V, E)$ satisfying the following properties:

1. $|V| = p$ with each $v_i \in V$ corresponding to the random variable $X_i$ (i.e. Each vertex in the graph represents a unique interest topic).

2. $P(X_i = k | \{X_j : j \neq i\}) = P(X_i = k | \{Y : (Y, X_i) \in E\})$ (i.e. An edge connects two vertices, each representing an interest topic, if and only if the presence of one topic affects the likeliness of the other topic appearing in the same memeber profile)

From the above assertions, the Markov Random Field associated with the graph $G$ over the random vector $M$ is the family of distributions of $M$ that factorize as

\[
P(m) \propto \exp \left( \sum_{(X_s, X_t) \in E} \phi_{st}(x_s, x_t) \right)
\]

where $m = (x_1, x_2, \ldots, x_p)$ and for each edge $(X_s, X_t) \in E$, $\phi_{st}$ is a mapping from pairs $(x_s, x_t) \in \{-1, 1\} \times \{-1, 1\}$ to $\mathbb{R}$.\[^2\]

For the case of when every $X_i$ takes on only values of 1 and -1, the distribution takes form

\[
P_{\theta^*}(m) = \frac{1}{Z(\theta^*)} \exp \left( \sum_{(X_s, X_t) \in E} \theta^*_{st} x_s x_t \right)
\]

where $Z(\theta^*)$ is the partition function that ensures the distribution sums to one and $\theta^*$ is a $\binom{p}{2}$-dimensional parameter vector with $\theta^*_{st} \in \mathbb{R}$ encoding the relationship between the random variables $X_s$ and $X_t$.\[^2\]
We aim to recover, in this Markov Random Field, signed edges $E^* := \{ \text{sign}(\theta_{st}^*) \mid (X_s, X_t) \in E \}$, a $(|E|)$-dimensional vector where each element corresponds to a pair of interest topics and is +1 when the two topics have positive co-dependence, -1 when they have negative co-dependence, and 0 when they have no co-dependence (i.e. no edge is present between their corresponding vertices in the network).

Thus, in this problem, estimating the network structure that captures the interdependence of members’ interest topics is a problem of recovering the signed edge set $E^*$ of the Markov Random Field through estimation of the parameter vector $\theta^*$.

3.2 The LASSO and Elastic Net Regularizations

One way of estimating the signed edge set $E^*$ is by considering the neighborhood set of every vertex in the graph $G$ representation of the Markov Random Field. In other words, we estimate the structure of the entire network by estimating for each interest topic, which other topics have significant co-dependence relationships with it in the members’ interest data.

In this context, we consider the conditional probability distribution of the random variable $X_i$, associated with the $i$-th interest topic, given all other variables $X_{-i} := \{X_j : j \neq i\}$.

The conditional probability distribution under the model given in the previous section takes the form

$$P_{\theta^*}(x_i | x_{-i}) = \frac{\exp(2x_i \sum_{t \in V \setminus v_i} \theta_{it}^* x_t)}{\exp(2x_i \sum_{t \in V \setminus v_i} \theta_{it}^* x_t) + 1}$$

This conditional probability is exactly the form for modeling log odds in the case of logistic regression; therefore, from this relation, we can estimate the elements of $\theta^*$ involving $X_i$ by considering $X_i$ as the response variable in a logistic regression with $X_{-i}$ as the explanatory variables,[2]

The edge recovery problem is now reduced to a logistic regression problem following this node-wise neighborhood selection procedure, iterated through all vertices (i.e. we can go through each interest topic to perform logistic regression to predict that topic using all the other topics based on members’ indication of their interests).

3.2.1 LASSO for Logistic Regression

In trying to recover the signed edge set $E^*$, we are determining whether or not there exists significant co-dependence between two interest topics. This translates in the logistic regression setting as a variable selection problem, selecting the appropriate set of covariates that have significant explanatory value for predicting the response. The LASSO technique (Least Absolute Shrinkage and Selection Operator) is one method of performing variable selection on linear models and generalized linear models.

In a simple linear model, the predicted response $\hat{y}_i$ is given by the linear combination

$$\hat{y}_i = \beta_0 + \sum_{j=1}^{p} x_{ij} \beta_j$$

where $X = [x_{ij}]$ is the $N \times p$ design matrix with rows as observations and columns as features, and the $\beta_j$’s are coefficients encoding the predictive effect of the i-th covariate with $\beta_0$ being the overall intercept term.

The usual method to estimate the $\beta_j$’s, or regression coefficients, involve the least squares minimization problem of

$$\min_{\beta_0, \ldots, \beta_p} \left[ \frac{1}{2N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \right]$$

where $\hat{y}_i$ is the predicted response as defined above, and $y_i$ is the observed response for the i-th observation.

Since this is an unconstrained minimization problem, the estimates of $\beta = (\beta_0, \ldots, \beta_p)^T$ could be of a wide range of values. We typically say the i-th covariate is not a significant predictor of the response if $\beta_i$ is “close” to 0, though “closeness” is a relative quantity because of the large variances associated with the
estimates. The LASSO technique automates variable selection in linear models by putting a constraint on the minimization. The problem above (with centered data where $\beta_0 = 0$) under the LASSO would take the form
\[
\min_{\beta \in \mathbb{R}^p} \left[ \frac{1}{2N} \|y - X\beta\|^2_2 + \lambda \|\beta\|_1 \right]
\]
where $\| \cdot \|_1$ is the $l$-1 norm, $\| \cdot \|_2$ the $l$-2 norm, $\lambda > 0$ is the tuning parameter, $y$ is the vector of observed responses, and $X, N, \beta$ are the same as above.\textsuperscript{[3]}

The LASSO technique is a useful tool for model selection because the introduction of the constraint tends the minimization problem to solutions where one or more of the $\beta_i$'s are set exactly to 0.\textsuperscript{[3]} As a result, those covariates whose regression coefficients have been set to zero will be the variables that do not contribute significant predictive value for the response as determined by the LASSO. The threshold that determines the level of “significant predictive value” is determined by the tuning parameter $\lambda$, and the selection of $\lambda$ will be addressed in Section 3.3.

The above translates to the logistic regression and the Markov Random Field edge recovery context by giving the estimation of the elements of $\theta^*$ involving $X_i$, as the following minimization problem:\textsuperscript{[2]}
\[
\min_{\theta_{-i} \in \mathbb{R}^{p-1}} [l(\theta, X) + \lambda \|\theta_{-i}\|_1]
\]
where
\[
l(\theta, X) := -\frac{1}{n} \sum_{j=1}^n \log P_{\theta}(m_{i,j}^1|m_{-i,j}^1)
\]

$X$ is as defined in Section 3.1, $\lambda > 0$ is the tuning parameter for the LASSO, $\| \cdot \|_1$ is the $l$-1 norm, $n$ is the number of members in the sample, $P_{\theta}$ is as defined in Section 3.2, and $m_{i,j}^1$ is 1 if the j-th member in the sample indicated interest topic i in his/her profile and -1 if the j-th member did not indicate the interest, and $m_{-i,j}^1$ is the vector of 1’s and -1’s indicating the presence of all other interests in his/her profile.

### 3.2.2 The Elastic Net

One limitation of the LASSO technique is in its performance to select consistent variables in the case of multiple highly correlated covariates. It is natural to assume that there exists certain interest topics in the New York tech economy that are highly correlated with other interest topics, giving rise to multicollinearity in the node-wise logistic regression problem. In our network estimate, it will be informative to see all of these correlated variables instead of just one or two of them.

The elastic net makes use of both $l$-1 and $l$-2 norms in the parameter estimation minimization problem in linear models, effectively extending both the LASSO technique ($l$-1 penalty) and ridge regression ($l$-2 penalty). Under the elastic net, the estimation of the elements of $\theta^*$ involving $X_i$ takes the following form:\textsuperscript{[4]}
\[
\min_{\theta_{-i} \in \mathbb{R}^{p-1}} \{ l(\theta, X) + \lambda \|\theta_{-i}\|_1 + \frac{1}{2}(1 - \alpha)\|\theta_{-i}\|_2^2 \}
\]
where $l(\theta, X)$ and $\lambda$ are defined as in the previous section, and $\alpha \in [0, 1]$ assigns weights to the LASSO and ridge regression components of the elastic net.

The elastic net preserves the variable selection property, tending certain $\theta^*$ parameter estimates to 0, through its $l$-1 penalty component, but also encourages the regression coefficients of correlated covariates to converge to each other, through its $l$-2 penalty component. This way, groups of highly correlated variables will tend to either all have significant (nonzero) estimated coefficients (i.e. all selected at once), or all have zero coefficients if the group as a whole has no explanatory value (i.e. none are selected).

### 3.3 Selection of $\lambda$ and Algorithm Implementation

The elastic net penalty depends on two pre-specified parameters that affect the outcome of the variable selection process. For our analysis, we use $\alpha = 0.5$ to give equal weight to the LASSO penalty and the ridge penalty. This is a reasonable compromise as there is no evidence that suggests favoring one penalty over the other would produce overall better interpretable results.
The choice of $\lambda$ is perhaps less intuitive; $\lambda$ controls the strength of the penalty, essentially what is the threshold of co-dependence above which two interest topics will be considered ‘dependent’ and have edge drawn between the two?

In this following implementation, we determine $\lambda$ by testing a range of $\lambda$ values through 5-fold cross validation. The idea is to first divide the sample observations randomly into 5 sub-samples, or folds, and run the constrained logistic regression 5 times for each proposed $\lambda$ value, leaving out a different fold each time. For each time the regression is run, we determine the accuracy of the predictions according to the logistic regression model against the data in the fold that was left out in that iteration. The accuracy would be determined by the proportion of the data in the left-out fold that is correctly predicted by the estimated model of that iteration. Thus, we can take the average and the variance of the accuracy measures across 5 folds for each proposed $\lambda$ value, and use it to compare the average accuracy across all proposed $\lambda$ values. Ideally, we would take the $\lambda$ value that gives the highest accuracy in prediction.

With the elastic net, there exists efficient algorithms that carry out the constrained minimization problem. In our analyses, we use the glmnet package in R. The implementation of the above mentioned Markov Random Field signed edge recovery through elastic net regularized logistic regression takes the following steps:

1. Arrange the collected data of members’ interest topics (or groups) into the data matrix $X$ of 1’s and -1’s like shown in Section 2.
2. Divide the rows of the matrix randomly into 5 folds to perform 5-fold cross validation.
3. For each fold, take the first column of the $X$ data matrix to use as the observed response vector, and take all other columns of the $X$ data matrix as the covariates.
4. Perform logistic regression with elastic net regularization using a range of proposed $\lambda$ values, $\alpha = 0.5$ and the explanatory and response variables from Step 3. Take the mean and variance of the accuracy for each proposed $\lambda$ across the 5 folds. Select the smallest lambda that has an accuracy within 1 standard deviation of the minimum accuracy. (This is more conservative than selecting the $\lambda$ with the highest accuracy).
5. There is sufficient co-dependence between an explanatory variable and the response variable if the regression coefficient for the explanatory variable is nonzero. In the network representation, this would result in an edge drawn between the two interest topics.
6. Repeat Steps 2 through 5, using a different column of the $X$ data matrix to use as the response variable in each iteration.
7. The result is a collection where every interest topic is related to a group of other interest topics. This can be represented as a network, or an estimated graph for the Markov Random Field.

Note: Because of the cross-validation procedure, there is necessary randomness introduced to estimate the best value of $\lambda$ for each logistic regression (i.e. different runs of the same algorithm may yield different estimations of the final network). Furthermore, a $k$-fold cross-validation technique should only run if the response variable has at least one success in each of the $k$-folds. So for 5-fold cross validation, we cannot effectively estimate the co-dependence relationships of a member interest that appears fewer than 5 times in our sample. Ideally, the interest would have to appear in much more than 5 member profiles for reliable results using 5-fold cross validation.

4. RESULTS

This section will display some of the networks obtained using the algorithm described above. Because the data is high dimensional (number of observations are on the order of $10^4$, and number of interests/groups are typically on the order of $10^3 \sim 10^4$), we may want to be conservative in drawing edges in the network estimate. Because of the nature of the algorithm, the edges are not necessarily symmetric, even though the idea of co-dependence is intuitive symmetric (i.e. interest A is co-dependent with interest B; therefore, interest B is naturally co-dependent with interest A). In the interest of noise reduction, an edge is drawn
in the following networks if and only if the two interests (or groups) mutually predict each other in their respective regressions.

We note that the network estimate of the Markov Random Field encodes both positive co-dependence and negative co-dependence. In the context of detecting various sub-communities in the New York City tech culture, it may be more useful to consider only those positive co-dependencies between specialized interests’ (or groups). The following graphs show only those positive co-dependencies (i.e. there is an edge between two nodes only if the appearance of one in a member profile is associated with a higher chance of the other appearing in the same profile).

4.1 Member Interests Network

The above network shows the dependence structure of interest topics for those interests who have been listed by more than 100 members. The colors of the network were determined using the Louvain Method for community detection, and the size of the node represents the degree of the node in the network. There is an edge between two nodes if the interests they represent show a positive relation in the data. Overall, we see some general patterns and concentrations of certain interests focuses: the nodes towards the left of the network are generally social and leisure oriented while the nodes on the right of the network are more technical. The green community in the center corresponds with many business and marketing related interests and can be seen as connecting those social activities to the more technical interests.
4.2 Member Groups Network

The above network shows the dependence structure of groups membership as of January 2012 in the New York tech community. We are able see some clear concentrations of groups that show dependence with each other. In particular, the light blue nodes on the far left of the network correspond with social and cultural focused groups whereas the light green nodes on the far right of the network correspond with programming and computing focused groups. The orange and dark green nodes in the center correspond with business and marketing focused groups.

5. DISCUSSION

We can also consider modifications to the methods above to better study the organization structure of the community through their interests. The current node-wise regression method for estimating the neighborhood of each node individuality takes away the inherent symmetry of co-dependence relationships in the network. An alternative to the node-wise approach is an attempt to directly estimate the conditional correlation structure of the data set. Peng et. al. has addressed this issue in Gaussian graphical models through the Sparse Partial Correlation Estimation (SPACE) technique.\(^6\) The pseudo-likelihood used in SPACE can theoretically be extended to fit the binary data and logistic regression of this context. However, there are not widely available software packages that can as efficiently perform the computations, especially for an adapted version for binary data, as there are for the elastic net; therefore, we are not at this stage ready to fully implement SPACE as a solution for the asymmetry problem.

One possibility of adjusting the current algorithm is to have weighted edges. The current algorithm only detects the presence or absence of an edge (co-dependence between two nodes), as well as the sign of the dependence relation; effectively, every edge in the network carries unit weight. Theoretically, it is possible to also estimate a weight representing the strength of the dependence relation encoded by the edge. This may lead to a more informative network, but would require justification for a weight assignment algorithm along with increased computational complexity for the estimation problem.

While the preliminary network estimations on the interests and groups data set for the online New York City technology based community yields some signs of sub-communities within the interest specialties and group members, there is still much to be improved on the model to better the structure of specialized knowledge within this group.

6. REFERENCES


7. APPENDIX: CODE

All computations for the above mentioned analyses were performed using processor servers from Cornell University’s Institute for Social and Economic Research (CISER). The computing made use of 6 servers running Windows 2008 R2. The software used for computing in this project is RStudio running R version 3.2.2. The network graphics were drawn using Gephi version 0.9.1. The R code for this project is given below:

```r
#ELASTIC NET REGULARIZED NETWORK ESTIMATION FOR SUB-COMMUNITIES IN A TECH ECONOMY
#
#Author: Sirui Wang (sw494@cornell.edu)
#Date: 23 February 2016

library(arules, lib="U:/RPackages")
library(doParallel, lib="U:/RPackages")
library(glmnet)
library(igraph)\noptions(warn=-1)

###For Interests Data

Interests = read.csv("Present Interests.csv") #Reads in Interest data document
Interests = as.matrix(Interests)
reconvertformat = function(input) {
  #To use to format data at a later step
  #Input should be a matrix whose rows are to be each element of the list.
  #The first element of each row will be the name of the entry.
  output.list = vector("list",length(input[,1]))
  names(output.list) = input[,1]
  for(i in 1:length(input[,1])) {
    empties = which(input[i,] == "")
    if(length(empties)!=0){
      min.index=min(empties)-1
    } else {
      min.index=length(input[i,])
    }
    entry = unique(input[i,2:min.index])
    output.list[[i]] = entry
    if(i%%1000==1) {print(i)}
  }
  return(output.list)
}

#Converts Interest document into a binary matrix with each column corresponding to an interest
system.time(Interest.L <- reconvertformat(Interests)) #10s
system.time(Interest.T <- as(Interest.L,"transactions")) #4s
```
# Removes interests that appear in fewer than 100 member profiles
count = colSums(Interest.M)
Interest.M = Interest.M[, which(count >= 100)]
count.members = rowSums(Interest.M)
Interest.M = Interest.M[, which(count.members != 0)]
Interest.M = 2 * (Interest.M - 0.5)

# Create Adjacency Matrix
build.adj = function(Interest.M, start = 1, end = 1){
  adj.mat = matrix(0, nrow = end - start + 1, ncol = ncol(Interest.M))
  colnames(adj.mat) = colnames(Interest.M)
  rownames(adj.mat) = colnames(Interest.M)[start:end]
  count = 1
  for(i in start:end) {
    fit = cv.glmnet(Interest.M[-i], Interest.M[, i], family = "binomial",
                    type.logistic = "modified.Newton", alpha = 0.5, nlambda = 50,
                    maxit = 1000, nfolds = 5)
    pos = which(coef(fit, s = fit$lambda.1se)[, 1] > 0)
    pos[which(pos <= i)] = pos[which(pos <= i)] - 1
    adj.mat[count, pos] = 1
    
    neg = which(coef(fit, s = fit$lambda.1se)[, 1] < 0)
    neg[which(neg <= i)] = neg[which(neg <= i)] - 1
    adj.mat[count, neg] = -1
  }
  print(i)
  count = count + 1
}
return(adj.mat)

# Collect Adjacency Matrices For Interests Data
set.seed(104)
system.time(adj.mat1 <- build.adj(Interest.M, 1, 100))
set.seed(106)
system.time(adj.mat2 <- build.adj(Interest.M, 101, 200))
system.time(adj.mat3 <- build.adj(Interest.M, 201, 300))
set.seed(105)
system.time(adj.mat4 <- build.adj(Interest.M, 301, 400))
system.time(adj.mat5 <- build.adj(Interest.M, 401, 500))
set.seed(103)
system.time(adj.mat6 <- build.adj(Interest.M, 501, ncol(Interest.M)))

### For Groups Data
Groups = read.csv("NY Member Groups RF.csv")

jointime = as.numeric(as.character(Groups[, 6])) / 1000
joindate = as.POSIXct(jointime, origin = "1970-01-01")
joindate = as.Date(joindate)

Groups = Groups[which(joindate < as.Date("2012-01-01")), 1:2] # Cutoff at 2012
system.time(Groups$system.time(Groups$L <- reconvertformatrf(Groups))

system.time(Groups$T <- as(Groups$L,"transactions"))

system.time(Groups$M <- as(Groups$T,"matrix"))

discordant format conversion complete.

mode(Groups$M)="numeric"
count = colSums(Groups$M)
Groups$M = Groups$M[,which(count>=20)]

Groups$M = 2*(Groups$M-0.5)

lambdas = vector("numeric",ncol(Groups$M))

build.adj = function(Groups$M,start=1,end=1){
  adj.mat = matrix(0,nrow=end-start+1,ncol=ncol(Groups$M))
  colnames(adj.mat) = colnames(Groups$M)
  rownames(adj.mat) = colnames(Groups$M)[start:end]
  count=1

  for(i in start:end) {
    fit = cv.glmnet(Groups$M[,i],Groups$M[,i],family="binomial",
        type.logistic="modified.Newton",alpha=0.5,maxit=5000,nfolds=5,parallel=T)
    pos = which(coef(fit,s=fit$lambda.1se)>0)
    pos[which(pos<=i)] = pos[which(pos<=i)]-1
    adj.mat[count,pos] = 1

    neg = which(coef(fit,s=fit$lambda.1se)[,1]<0)
    neg[which(neg<=i)] = neg[which(neg<=i)]-1
    adj.mat[count,neg] = -1

    print(i)
    count=count+1
    lambdas[i] <<- fit$lambda.1se
  }
  return(adj.mat)
}

cl <- makeCluster(5)
registerDoParallel(cl)
set.seed(101)

system.time(adj.mat<-build.adj(Groups$M,1,250))

set.seed(102)

system.time(adj.mat<-build.adj(Groups$M,251,ncol(Groups$M)))