Large Scale Structure Learning of Conditional Gaussian Graphical Models

Manzil Zaheer, Carlos Ramírez, Soumya Batra {MANZIL, CARLOSRC, SOUMYABATRA}@CMU.EDU
Carnegie Mellon University, Pittsburgh, PA 15213, USA

Advised by: Mladen Kolar MLADEN.KOLAR@CHICAGOBOOTH.EDU
Booth School of Business, Chicago, IL 60637, USA

1. Introduction

Gaussian graphical models (GGMs) are commonly used to represent relationships between objects in complex systems. They can be thought as a class of undirected graphical models in which a set of random variables follow a multivariate Gaussian distribution where nodes correspond to random variables in the set while edges describe the conditional independence relationships among them. An edge between two nodes is absent if and only if the two random variables that are represented by those nodes are independent conditional on all other variables. Provided the normality assumption, when two random variables are independent conditional on all other variables the corresponding entry in the precision matrix would be zero. Given some observations of the set of random variables we then can infer conditional independence relationships between various random variables in the set by just identifying zeros in the precision matrix.

Complex systems where uncovering such relationships among variables is of particular interest are modern financial markets where firms have become increasingly linked to each other through a complex and usually opaque network of relationships.[1] From a economic point of view, if investors take into account these relationships, prices of both linked firms should adjust when news about one of the linked firms is released into the market (as long as those news represent a change in the fundamental value of one of the linked firms). To understand prices behavior then it may be important to uncover and understand the nature of such (possibly time-varying) relationships.

[1]For instance, a firm may find itself in the roles of customer, supplier, partner or competitor all with the same firm at the same time but in different markets. However, many of these relationships may not be so concrete. For example, two firms may be temporarily related by their exposure to the same portfolio of assets or by something more subtle such as social connections between their top executives.

2. Background and Related Work

2.1. Background

Since firms' relationships may be important to understand prices behavior obtaining a better representation of the underlying business network is of key importance. Two of the main problems in determining such business network is that information about firms connections is typically opaque and that those relationships tend to be complex. The goal of this project is to uncover the time-varying structure that characterize the U.S. business network in an economic and meaningful manner. To do so, we use GGMs to represent the U.S. business network. Using financial data such as market prices and returns we aim to uncover the time-varying pattern of zeros on the precision matrix of a GGM. In particular, we develop a novel method for learning the fixed structure of a graphical model where we allow the parameters of the model to change (similar to (Kolar et al., 2010)). Then, we plan to use the inferred network and compare it with the network we get from customer-supplier data in which two firms are connected in a given year as long as one of them represents more than 10% of the total sales of the other company for that year.

2.2. Related work

For sake of exposition we divide this section into two subsections. In the first subsection we review some of the main methods used for estimating the network structure in GGMs. In the second subsection we present some recent research in finance that tries to uncover the relation between business connections and asset prices.

2.2.1. Estimating the network structure in GGMs

As we previously mentioned the inverse covariance matrix, known as precision matrix, is useful for identifying relationships between complex objects. (Meinshausen & Bühlmann, 2006) describes an alternative to standard covariance selection for sparse high-dimensional graphs –

neighborhood selection with a Lasso– and shows how it is a computationally more attractive alternative. The method optimizes a convex function, applied consecutively to each node in the graph. The choice of the penalty function is crucial to the convergence of the algorithm, especially in high dimensions. The neighborhood is defined by the nonzero coefficient estimates of the l_1-penalized regression

$$\widehat{ne}_a^{\lambda} = \{b \in \tau(n) : \hat{\theta}_b^{a,\lambda} \neq 0\} \tag{1}$$

where λ is the penalty parameter. A useful choice for λ is suggested as the prediction-oracle value:

$$\lambda_{\text{oracle}} = \arg\min_{\lambda} E(X_a - \sum_{k \in \tau} \hat{\theta}_k^{a,\lambda} X_k)^2 \tag{2}$$

This minimizes the predictive risk among all Lasso estimates. An estimate for λ_{oracle} is obtained by the cross-validated choice $\lambda_c v$. This approach is consistent under the following assumptions: availability of independent observations from the model, high dimensionality, non singularity, sparsity, bounds on magnitude of partial correlations and neighborhood stability.

(Yuan & Lin, 2007) uses penalized likelihood methods for estimating the concentration matrix in GGMs. The matrix, thus estimated is positive definite, which leads to simultaneous model selection and estimation. A BIC-type criterion is used for selecting the tuning parameter in this case. A Lasso penalty is employed on the off-diagonal elements of the concentration matrix as well.

(Kolar et al., 2010) describes two different techniques for estimating time-varying networks, which build on Lasso regularized logistic regression which is formalized as a convex optimization problem and solved using generic solvers. The key point of this paper is to reverse engineer networks that are latent, and topologically evolving over time, over a series of nodal attributes. An important assumption taken throughout the paper is of independent observations at different points of time. Tuning parameters selection remains a problem, and so is nonparametric estimation of change points.

(Wahlberg et al., 2012) presents an alternating augmented Lagrangian method for convex optimization problems where the sum consists of two terms : the first term is separable in variable blocks, and the second term is separable in differences between consecutive variable blocks. ADMM is applied to l_1 mean filtering and l_1 variance filtering. This method is around $10,000$ times faster compared to generic optimization solvers SDPT3.

(Friedman et al., 2008) applies lasso penalty to the inverse covariance matrix using a coordinate descent procedure. (Friedman et al., 2008) also provides a conceptual link between the exact problem and approximation suggested by (Meinshausen & Bühlmann, 2006).

2.2.2. Asset pricing and business networks

One of the main areas in financial economics is asset pricing. Asset pricing theory basically studies the value of assets with a stream of uncertain payoffs. Since low prices imply high rates of returns one can think of asset pricing theories as explanations of why some assets have higher returns than others. Provided that higher returns are compensation for holding higher risk, understanding why some assets are riskier than others is of key importance in asset pricing.

One of the main notions in asset pricing is aggregate risk, which is the risk that cannot be eliminated through a diversified portfolio. By holding a diversified portfolio investors losses in one particular asset are compensated with gains in other assets within the same portfolio. Thus, the exposure of an asset to aggregate risk determines how risky the asset is, and thus its expected return since in principle all other risks can be eliminated through diversification.

Then understanding the sources of aggregate risk is of key importance. In a recent theory, (Acemoglu et al., 2012) posses the idea that shocks affecting particular companies (or industry sectors) may be at the core of the origin of aggregate shocks, provided that some of those companies may be well connected with many others in the economy. Therefore, shocks affecting a well connected company, say company $\mathcal{A}$, may not be eliminated through diversification since those shocks not only affect company $\mathcal{A}$ but also the companies connected to company $\mathcal{A}$. Therefore, shocks originated at the micro level may spread out and become aggregate shocks.

Following the above idea (Buraschi & Porchia, 2012) and (Ahern, 2013) explore whether the importance of an industry sector determines the returns of the companies within that sector. Both studies implicitly model the aggregate economy as a network of related industries and proxy the importance of a sector with centrality measures from graph theory. The main finding in both papers is that industries that are more central in the network of intersectoral trade earn higher stock returns than industries that are less central since stocks in more central industries have greater market risk because they have greater exposure to sectoral shocks that transmit from one industry to another through intersectoral trade. Their empirical evidence then suggests that sectoral shocks that contribute to aggregate risk are more likely to pass through central industries than peripheral industries.

Using customer supplier data, (Cohen & Frazzini, 2008) and (Wu & J.R.Birge, 2014) provide evidence that not only intersectoral linkages matter but also economic linkages between particular companies matter, such as customer-supplier connections. For instance, (Cohen & Frazzini,

2008) provide evidence that well connected firms tend to earn higher returns than less connected ones. On the other hand, (Wu & J.R.Birge, 2014) provides evidence that manufacturing firms that are more central in the network earn lower returns, while logistics firms that are more central in the network earn higher returns. They argue that centrality and multiplicity of suppliers have different risk implications for firms operating in different industries. Their idea is that more central firms in manufacturing choose their suppliers to operationally hedge shocks transmitted from other firms and earn lower returns due to lower aggregate risk. On the contrary, more central firms in logistics are shock aggregators, earning higher returns due to their exposure to greater aggregate risk.

3. Problem Abstraction and Methodology

The problem described above can be abstracted away and formulate as follows. Let $\{x_i, z_i\}_{i\in[n]}$ be an independent sample from a joint probability distribution (X, Z) over $\mathbb{R}^p \times [0,1]$.[2] We assume that the conditional distribution of X given $Z = z$ is given as

$$X \mid Z = z \sim \mathcal{N}(\mu(z), \Sigma(z)). \tag{3}$$

Let $p(z)$ be the density function of Z. We assume that the density is well behaved as specified later, however, we do not pose any specific distributional assumptions. Furthermore, we do not specify parametric relationships for the conditional mean or variance as functions of $Z = z$.

Our goal is to learn the conditional independence relationships among components of vector X given $Z = z$. Let $\Omega(z) = \Sigma(z)^{-1} = (\omega_{ab}(z))_{a,b\in[p]\times[p]}$ be the inverse conditional covariance matrix. The pattern of non-zero elements of this matrix encodes the conditional independencies between the components of the vector X. In particular

$$X_a \perp X_b \mid X_{-ab}, Z = z \quad \Leftrightarrow \quad \omega_{ab}(z) = 0 \tag{4}$$

where $X_{-ab} = (X_c \mid c \in [p]\backslash\{a, b\})$. Denote the set of conditional independencies given $Z = z$ as

$$S(z) = \{(a, b) \mid \omega_{ab}(z) \neq 0\}. \tag{5}$$

Using $S(z)$, we define the set

$$\begin{aligned} S &= \cup_{z\in[0,1]} S(z) \\ &= \{(a, b) \mid \omega_{ab}(z) \neq 0 \text{ for some } z \in [0, 1]\}. \end{aligned} \tag{6}$$

Let $\bar{S}$ be the complement of S, which denotes pairs of components of X that are conditionally independent irrespective of the value of Z.

Our goal is to estimate S, for which it suffices to find $\Omega(\cdot)$. Suppose that our data set consists of n time instances $\{z_1, \cdots, z_n\}$. At each z_i, we observe n_i instances of data vector x_{ij}. Motivated by the graphical lasso procedure of (Friedman et al., 2008), we propose the following optimization problem for learning the structure of a graphical model which allows the parameters of the model to change:

$$\begin{aligned} \min_{\Omega(\cdot)\in\mathcal{F}} \Bigg\{ &\sum_{i\in[n]} \left(\mathsf{tr}(C_i\Omega(z_i)) - \log|\Omega(z_i)| + \mu\|\Omega(z_i)\|_1\right) \\ &+\lambda \mathsf{pen}\left(\{\Omega(z_i)\}_{i\in[n]}\right)\Bigg\} \\ \text{with } \mathcal{F} &= \{\Omega(\cdot) \mid \forall i, \Omega(z_i) \succ 0\} \end{aligned} \tag{7}$$

where $C_i = \sum_{j=1}^{n_i} \frac{(x_{ij}x'_{ij})}{n_i}$ and $\mathsf{pen}\left(\{\Omega(z_i)\}_{i\in[n]}\right)$ is a penalty function that controls the complexity of the fitted model. Also the penalty function should encourage the precision matrix to be smooth functions of time.

3.1. Setting the Problem

We select the penalty function as $\mathsf{pen}\left(\{\Omega(z_i)\}_{i\in[n]}\right) = \sum\|\Omega(z_{i+1}) - \Omega(z_i)\|_F$. The reason for selecting this penalty will become clearer as we rewrite the optimization problem as:

$$\begin{aligned} \min_{\Omega(\cdot)\in\mathcal{F}} \Bigg\{ &\sum_{i=1}^{n} \left(\mathsf{tr}(C_i\Omega(z_i)) - \log|\Omega(z_i)| + \mu\|\Omega(z_i)\|_1\right) \\ &+\lambda \sum_{i=1}^{n-1} \left(\sqrt{\sum_{a,b}(\omega_{ab}(z_{i+1}) - \omega_{ab}(z_i))^2}\right)\Bigg\} \\ \text{with } \mathcal{F} &= \{\Omega(\cdot) \mid \forall i, \Omega(z_i) \succ 0\} \end{aligned} \tag{8}$$

One can immediately observe that this choice of penalty function would encourage successive values of $\omega_{ab}(z_i)\omega_{ab}(z_{i+1})$ to be similar. At the same time, L_2 nature would ensure smoothness. Thus we can say that the penalty function has been carefully designed so as perform model selection and control the smoothness of the estimator, i.e. we basically impose that precision matrices need to be smooth functions of time. We believe however that our choice of penalty function keeps the problem tractable without losing recovery power.

An alternative approach to the problem would be to extend the approach of (Boyd et al., 2011). Again, one would develop an appropriate penalty that would fix the graph structure, but allow for change in the parameters.

Coming back to our choice of penalty function, to handle large databases we need to make the optimization problem scalable. First step towards parallelizing would be to

[2] We use $[n]$ to denote the set $\{1, \ldots, n\}$.

rewrite the optimization problem as:

$$\begin{aligned}\min_{\Omega(\cdot)\in\mathcal{F},R(\cdot)} \Bigg\{ &\sum_{i=1}^{n} \left(\mathsf{tr}(C_i\Omega(z_i)) - \log|\Omega(z_i)| + \mu\|\Omega(z_i)\|_1\right) \\ &+\lambda\sum_{i=1}^{n-1}\|R_i\|_F \Bigg\} \\ \text{s.t. } & R(z_i) = \Omega(z_{i+1}) - \Omega(z_i) \\ \text{with } & \mathcal{F} = \{\Omega(\cdot) \mid \forall i, \Omega(z_i) \succ 0\}\end{aligned} \tag{9}$$

Then introducing the constraint set $\mathcal{C} = \{(\Omega(\cdot), R(\cdot)) : R(z_i) = \Omega(z_{i+1}) - \Omega(z_i), \forall i\}$ and associated indicator function $I_{\mathcal{C}(\cdot,\cdot)}$, we get the function in standard form to apply ADMM as:

$$\begin{aligned}\min_{\Omega(\cdot)\in\mathcal{F},R(\cdot),W(\cdot),S(\cdot)} \Bigg\{ &\sum_{i=1}^{n} \left(\mathsf{tr}(C_i\Omega(z_i)) - \log|\Omega(z_i)| + \mu\|\Omega(z_i)\|_1\right) \\ &+\lambda\sum_{i=1}^{n-1}\|R_i\|_F + I_{\mathcal{C}}(W,S) \Bigg\} \\ \text{s.t. } & \Omega(z_i) = W(z_i) \\ & R(z_i) = S(z_i) \\ \text{with } & \mathcal{F} = \{\Omega(\cdot) \mid \forall i, \Omega(z_i) \succ 0\}\end{aligned} \tag{10}$$

We explore in the next section our proposed implementation of the above problem using ADMM, a distributed optimization problem solving strategy.

3.2. ADMM steps

We derive an ADMM method to solve the problem described above. For simplicity define $\Omega(z_i) = \Omega_i$ and $R_i = \Omega_i - \Omega_{i-1}$. For the optimization problem in (10), we can write the augmented Lagrangian for this problem as:

$$\begin{aligned}&L_\rho(\Omega, R, W, S, U, T) \\ &= \sum_{i=1}^{n} \left(\mathsf{tr}(C_i\Omega(z_i)) - \log|\Omega(z_i)| + \mu\|\Omega(z_i)\|_1\right) \\ &\quad + \lambda\sum_{i=1}^{n-1}\|R_i\|_F + I_{\mathcal{C}}(W,S) \\ &\frac{\rho}{2}\|\Omega - W + U\|_F^2 + \frac{\rho}{2}\|R - S + T\|_F^2\end{aligned} \tag{11}$$

where U, T are scaled dual variables associated with the constraints $\Omega_i = W_i$ and $R_i = T_i$ respectively. Now we highlight only the flow and main steps involved. In each iteration k of ADMM, we perform the following three steps.

- Step 1: Since the objective function is separable in Ω_i and R_i, the first step of the ADMM algorithm consists of $2n-1$ separate minimizations.

  (a) The first n minimizations correspond to

  $$\begin{aligned}\Omega_i^{k+1} := \arg\min_{\Omega_i \succ 0} \; \Big\{&\mathsf{tr}(C_i\Omega_i) - \log|\Omega_i| \\ &+\frac{\rho}{2}\|\Omega_i - W_i^k + U_i^k\|_2^2\Big\}\end{aligned}$$

  with $i \in [n]$ which can be solved analytically, as follows:

  (a.1) Compute the eigenvalue decomposition of

  $$\rho(W_i^k - U_i^k) - C_i = Q\Lambda Q'$$

  where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_p)$

  (a.2) Let

  $$\mu_j := \frac{\lambda_j + \sqrt{\lambda_j^2 + 4\rho}}{2\rho}, \text{ with } j \in [p]$$

  (a.3) Finally, set

  $$\Omega_i^{k+1} = Q\ \text{diag}(\mu_1, \cdots, \mu_p)\ Q'$$

  (b) The last $n-1$ minimizations correspond to

  $$\begin{aligned}R_i^{k+1} := \arg\min_{R_i} \; \Big\{&\lambda\|R_i\|_F + \cdots \\ &+\frac{\rho}{2}\|R_i - S_i^k + T_i^k\|_2^2\Big\}\end{aligned}$$

  with $i \in [n-1]$ which simplifies to

  $$R_i^{k+1} = \mathcal{S}_{\frac{\lambda}{\rho}}\left(S_i^k - T_i^k\right)$$

  where

  $$\mathcal{S}_{\frac{\lambda}{\rho}}\left(S_i^k - T_i^k\right) = \left(1 - \frac{\lambda}{\rho}\frac{1}{\|(S_i^k - T_i^k)\|_F}\right)(S_i^k - T_i^k)$$

  and $\mathcal{S}_{\frac{\lambda}{\rho}}(0) = 0$.

- Step 2: In the second step we project $(\Omega^{k+1} + U^k, R^{k+1} + T^k)$ onto the constraint set $\mathcal{C}$, i.e.

  $$(W^{k+1}, S^{k+1}) := \Pi_{\mathcal{C}}(\Omega^{k+1} + U^k, R^{k+1} + T^k)$$

  (a) Let $Q_{k+1} = \Omega^{k+1} + U^k$ and $V_{k+1} = R^{k+1} + T^k$

  (b) The projection can then be performed by solving the following minimization problem:

  $$\begin{aligned}&\text{minimize } \|W^{k+1} - Q^{k+1}\|_F{}^2 + \|S^{k+1} - V^{k+1}\|_F^2 \\ &\text{subject to } S = DW,\end{aligned}$$

  where D is forward difference operator with $D \in \mathbb{R}^{(n-1)p\times np}$ and again just to be clear $W = \{W_1; \cdots; W_n\}$, $S = \{S_1; \cdots; S_{n-1}\}$.

(c) We solve the optimality condition:

$$(I + D'D)W = Q + D'V$$

for W and then using $S = DW$, obtain S as well. We obtain $E = (I + D'D)$. Note that E will be fixed for a given set of data and hence, can be precomputed and saved in memory for use in projection function. Also, let $B = Q + D'V$. Finally, steps for projection can be summarized as:

(c.1) We get B as:

$$\begin{aligned} & B = Q + D'V \\ \implies & B_1 = Q_1 - V_1, \\ & B_n = Q_n + V_{n-1}, \\ & B_i = Q_i + (V_{i-1} - V_i), i \in [n-1] \end{aligned}$$

(c.2) W can be obtained as described above. However, we need to take sparsity constraint into account as well. We do that by using a soft threshold of $\frac{\mu}{\rho}$ on w, based on the proposition 1 of (Friedman et al., 2007). The result basically implies that, we should first obtain a solution assuming $mu = 0$ and then obtain the solution for $\mu \neq 0$ by simple thresholding on the previous result. Hence we get W in two steps as:

(c.2.1)

$$\begin{aligned} (I + D'D)W &= \underbrace{Q + D'V}_{=B} \\ \text{or} \quad EW &= B \\ \text{or} \quad W &= E^{-1}B \end{aligned}$$

Notice that E is a very well conditioned matrix, so inverting it is not an issue. Nevertheless we can solve the linear system by Cholesky factorization technique and infact its factors are available in closed form, e.g. see (Wahlberg et al., 2012).

(c.2.2)

$$W = \mathcal{S}_{\frac{\mu}{\rho}}(W)$$

where $\mathcal{S}$ is the element-wise soft thresholding operator.

(c.3) Finally, we get S as:

$$\begin{aligned} & S = DW \\ \implies & S_i = W_{i+1} - W_i, i \in [n-1] \end{aligned}$$

- Step 3: Finally, we update the dual variables:

$$U_i^{k+1} := U_i^k + (\Omega_i^{k+1} - W_i^{k+1}),\ i \in [n]$$

and

$$T_i^{k+1} := T_i^k + (R_i^{k+1} - S_i^{k+1}),\ i \in [n-1]$$

3.2.1. Stopping criteria

Let $e_p^k = [\Omega^k - W^k; R^k - S^k]$ and $e_d^k = -\rho[W^k - W^{k-1}; S^k - S^{k-1}]$ be the primal and dual residuals at iteration k. We stop the algorithm when both the primal and the dual residuals satisfy $\|e_p^k\|_F \leq \epsilon^{\text{pri}}$, $\|e_d^k\|_F \leq \epsilon^{\text{dual}}$ where $\epsilon^{\text{pri}} > 0$ and $\epsilon^{\text{dual}} > 0$ are the tolerance of the primal and dual problems respectively. These tolerances can be set via an absolute plus relative criterion,

$$\begin{aligned} \epsilon^{\text{pri}} &= p\sqrt{2n-1}\epsilon^{\text{abs}} + \epsilon^{\text{rel}} \max\{\|\Omega^k\|_F + \|R^k\|_F, \|W^k\|_F + \|S^k\|_F\}, \\ \epsilon^{\text{dual}} &= \sqrt{n}\epsilon^{\text{abs}} + \epsilon^{\text{rel}}\rho(\|U^k\|_F + \|T^k\|_F) \end{aligned}$$

where ϵ^{abs} and ϵ^{rel} are absolute and relative tolerances (see (Boyd et al., 2011) for more details).

4. Experiments

Using monthly returns from 1980 to 2004 of about 6,636 U.S. firms we uncover the conditional independence relationships among them. We select those companies since they appear in a database we have access to with information about customer-supplier relationships. To assess whether the inferred time series of conditional independence graphs resemble the time series of graphs we get from the customer-supplier database, we use precision/recall as well as spectral methods for comparing graphs, e.g. cospectrality of graph laplacians.

For illustrative purposes we first describe the customer-supplier database and some of the characteristics of the monthly return data. We then explain how we select the tuning parameters μ (sparsity) and λ (smoothness) and the main features of our results.

4.1. Description of the data: customer-supplier and return data

Under regulation SFAS No. 131 firms are required to disclose financial information for any industry segment that comprised more than 10% of consolidated yearly sales, assets, or profits, and the identity of any customer representing more than 10% of the total reported sales. The sample consists of all firms listed in the CRSP/Compustat database with nonmissing values of book equity (BE) and market equity (ME) at the fiscal year-end for which one can identify the customer as another traded CRSP/Compustat firm. Since prior to 1998, most firms customers were listed as an abbreviation of the customer name we use the customer identity identified by (Cohen & Frazzini, 2008). The final sample includes about 27,000 distinct firm-year relationships, representing a total of about 7,000 unique supplier-customer relationships between 1980 and 2004. Each observation in our dataset indicates both the name of the customer and its supplier, the year in which both companies were linked and the strength of the link (which is repre-

sented by the fraction of sales a given customer represents for a particular supplier).

Since we use monthly returns to uncover time-varying relationships between firms we further clean the customer-supplier database to consider only those firms for which we have nonmissing observations in returns from 1990 to 2000. We obtain that information from Compustat as well. This procedure selects only about 1140 firms for which we compare the output of both graphs.

4.2. Implementation

We implemented the problem as a MapReduce program in Hadoop and ran it across 8 workers. In a single job execution, we used two set of mappers to operate on the two minimization equations, with each individual mapper operating on a single instance of data. Thus, there were a total of n mapper instances to operate on the first set of minimization equations and (n-1) mapper instances to operate on the second set of minimization equations. The two set of mappers were then combined in a single reducer instance which carried the projection operation as well as updated the dual variables. The reducer writes output to two separate output files in the same format as that of input files. The output files are then reused as input for the next job execution. We keep executing new jobs with updated input files until the convergence condition is met.

We can see that the first set of minimization equations are computationally very expensive since each involves computing eigen value decomposition of the combination of the matrices W, U and C. This is precisely where the power of Hadoop is realized as the eigen value decompositions for each instance i is carried by a separate mapper in parallel. We can also see that the projection function is not a computationally intensive task and thus, the results from both mapper sets can be combined in a single reducer instance without losing on execution time. Figure 1 gives an architectural overview of our algorithm.

Following are the implementation details of a single MapReduce job:

1. **Input Files** : We used two input files : one containing W,U,C,Ω matrix values and the other containing R,S,T matrix values.

2. **File Splits** : The 1st input file was split into blocks of size 4n and distributed amongst the first set of mappers and the 2nd input file was split into blocks of size 3n and distributed amongst a second set of mappers. Note that file 1 contained n instances, while file 2 contained (n-1) instances. This ensured that each mapper handled just one instance of the data. We also used a custom record reader to change the output value. Following are the (key,value) mappings:
   (a) (path, file) $\rightarrow$ (i, $[W_i, U_i, C_i]$)
   (b) (path, file) $\rightarrow$ (i, $[R_i, S_i, T_i]$)

3. **Mappers** : Corresponding to the 2 input files, we used 2 mappers that carried the mapping as:
   (a) (i, $[W_i, U_i, C_i]$) $\rightarrow$ (0, $[U_i, C_i, \Omega_i]$)
   (b) (i, $[R_i, S_i, T_i]$) $\rightarrow$ (0, $[R_i, T_i]$)
   The first mapper carried the first n minimizations while the second mapper carried the last n-1 minimizations as given in the above ADMM steps. Also, note that we used the same key 0 for both set of mapper instances so that all the (2n-1) minimization results can be collected in a single reducer instance.

4. **Reducer** : The output of both mappers was received by a single reducer instance that carried the projection operation on them and produced the following (key,value) mappings while writing the output to two different files:
   (a) $\forall j \in [n], (0, [U_j, C_j, \Omega_j]) + (0, [R_j, T_j]) \rightarrow \forall i \in [n], (i, [W_i, U_i, C_i, \Omega_i])$
   (b) $\forall j \in [n], (0, [U_j, C_j, \Omega_j]) + (0, [R_j, T_j]) \rightarrow \forall i \in [n], (i, [R_i, S_i, T_i])$

5. **Reuse of Output Files** : We purposely created the output files in the same format as that of input files so that we can reuse them across all jobs.

4.3. Selection of tuning parameters

To select the tuning parameters μ and λ in (8) we perform grid search among pairs $(\mu, \lambda) \in [0.1, 1] \times [0.1, 1]$ and select the pair (μ^*, λ^*) which attains either the highest recall or the highest precision. To define recall and precision we use information from the customer-supplier database. We then use edges that appear in graphs from the customer-supplier database as the edges we would like to retrieved from our implementation. In particular, for each pair (μ, λ) we compute the recall and precision from 1990 to 2000 as follows:

$$|\text{recall}| = \frac{|\text{Edges identified}| \cap |\text{True Edges}|}{|\text{True Edges}|}$$

$$|\text{precision}| = \frac{|\text{Edges identified}| \cap |\text{True Edges}|}{|\text{Edges identified}|}$$

4.4. Results

First of all we test our approach and implementation on a synthetic data set. We generate an artificial normally distributed data for $p = 10, n = 4, n_i = 3$. Next we run the

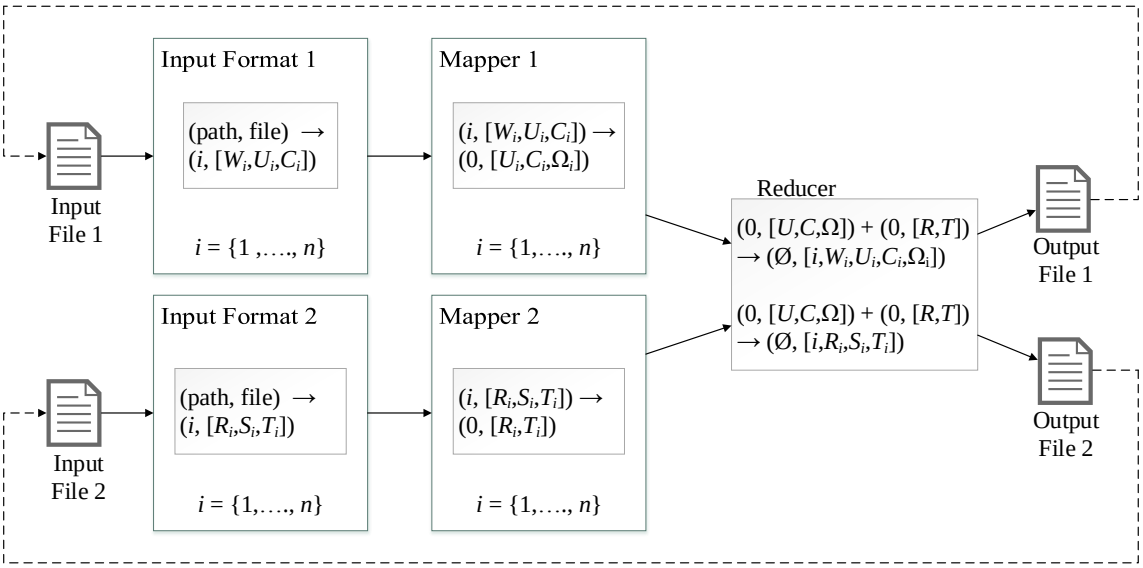

Figure 1. Architectural overview of our MapReduce implementation

proposed method on this synthetic dataset and plot the precision and recall for the grid $(\mu, \lambda) \in [0.1, 1] \times [0.1, 1]$ in Figures (2) and (3) respectively. As we achieve precision around 0.5 for $p > n$ case, we can say that the proposed method performed fairly well.

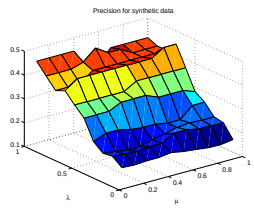

Figure 2. Precision using synthetic data as function of λ and μ respectively (from left to right)

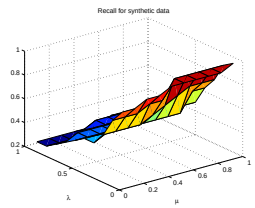

Figure 3. Recall using synthetic data as function of λ and μ respectively (from left to right)

Now moving onto real financial data, Figures (4) and (5) show the precision and recall as functions of tuning parameters λ (smoothness) and μ (sparsity) respectively. Notwithstanding both figures exhibit a nonmonotonic behavior with respect to λ the overall level of recall and precision in both figures is small. Thus, using only monthly return data does not help uncovering customer-supplier linkages among the firms in our database.

However, our implementation seems to uncover clusters of firms that represent different industries. A similar result was obtained by (Lafferty et al., 2012), while trying to infer graphical models from returns data. Therefore, even when customer-supplier relationships may not be identified from using return data other economical relationships seem to be identified. Since shocks may affect firms within the same industry similarly, returns of firms in the same industry tend to co-move.

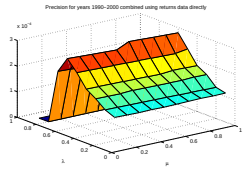

Figure 4. Precision using monthly returns data as function of λ and μ respectively (from left to right)

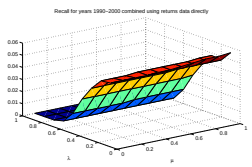

Figure 5. Recall using monthly returns data as function of λ and μ respectively (from left to right)

To tackle the identification problem of customer-supplier linkages we further explore the nature of the difference among returns of different companies. In empirical asset pricing the cross section of expected returns is typically characterized using the Fama and French factors (see (Fama & French)). In particular, (Fama & French) pose the idea that the expected excess return of a given company j at period t, namely $E[r_{jt} - r_{ft}]$, follows

$$E[r_{jt} - r_{ft}] = \alpha + \beta_m(r_{mt} - r_{ft}) + \cdots + \beta_s \text{SMB}_t + \beta_h \text{HML}_t \quad (12)$$

where r_{ft} represents the risk free rate at period t, r_{mt} the return of the market portfolio at period t, SMB_t the histori-

cal excess return of a portfolio that buy small cap firms and sell big cap firms and HML_t the historical return of a portfolio that buys high book to market ratio firms and sell low book to market ratio firms.

Hoping to improve our previous results and to incorporate the above findings we run (12) for each firm. We compute the residuals of such regressions per each month. We then try to uncover customer-supplier relationships from the information that is not spanned by the three Fama and French factors. Figures (6) and (7) show the precision and recall as functions of tuning parameters λ and μ respectively once we use residual of the Fama and French regressions instead of monthly returns.

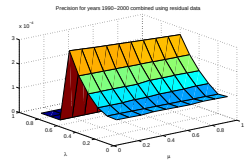

Figure 6. Precision using residuals from Fama and French regressions as function of λ and μ respectively (from left to right)

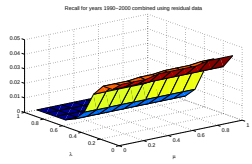

Figure 7. Recall using residuals from Fama and French regressions as function of λ and μ respectively (from left to right)

Even when precision and recall may not be monotonic functions of λ –as we see in the above figures– the overall precision and recall does not improve if we use residuals rather than returns. In fact, once we correct for the existence of the three Fama and French factors it becomes hard to understand the nature of the inferred relationships among firms.

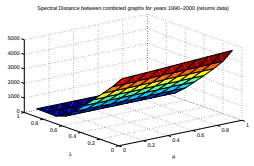

Figure 8. Spectral distance using monthly returns data as function of λ and μ respectively (from left to right)

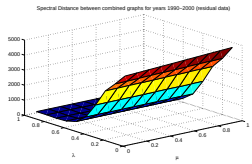

Figure 9. Spectral distance using residuals from Fama and French regressions as function of λ and μ respectively (from left to right)

Even if the proposed method could not recover the desired consumer supplier relationships, we next try to determine if the structure of true consumer-supplier graph and recovered graph are similar or not. For this purpose, we resort to spectral methods. In figures (8) and (9) we show the spectral distance between the customer-supplier graphs and the ones we obtain with our method, which also indicates that

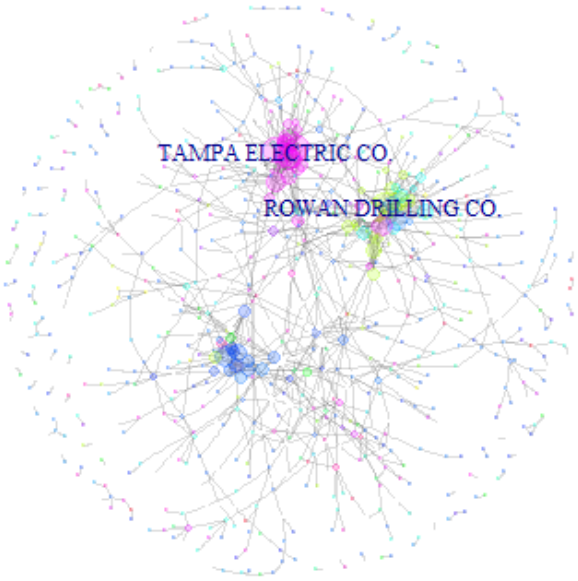

Figure 10. Conditional independence graph from returns data in 1991. Each node represents a firm. An edge between two companies exists as long as both companies are conditionally dependent provided the existence of all other companies in the economy. Colors represent firms' industries which are determined by SIC codes.

no such structural similarity could be found for any value of the tuning parameters.

To give a glimpse of the structure of the business network we recover with our method figures 10 and 11 show the conditional independence graph and the customer-supplier network in 2000. For computing the conditional independence graph we select the pair (λ, μ) to maximize precision. In both graphs companies are colored based on their industry classification code (SIC). In both graphs each node represents one firm in our customer-suppliers database and the size of the node is given by its degree plus one. To infer the conditional independence graph we simply use the time series of firms' monthly returns.

5. Conclusion

Conditional Gaussian Graphical Models provide an important tool to uncover relationships among different variables in complex systems. We use such tool to uncover linkages among firms in financial markets and see whether such links can be understood from an economic point of view. Using returns data the use of graphical models seems to uncover industry relationships among firms. However, it seems to not provide further information about the nature of these relationships besides the identification of indus-

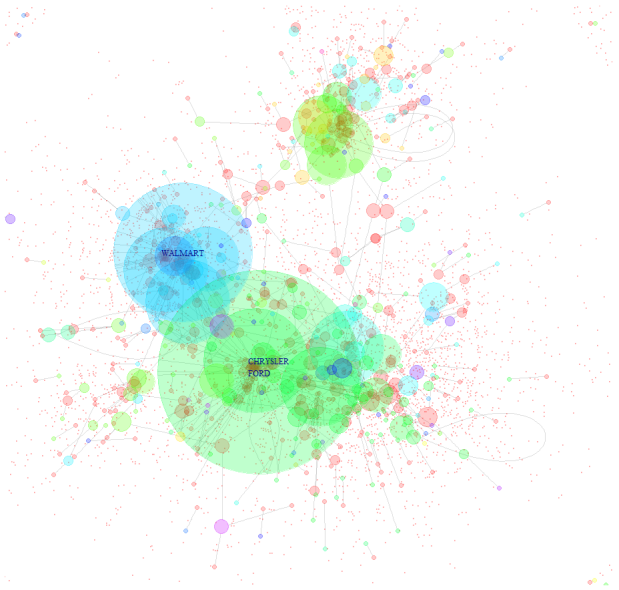

Figure 11. Customer-supplier network in 1991. Each node represents a firm. Each edge represent the existence of a customer-supplier linkage between the two linked companies. Colors represent firms' industries and the size of a firm is determined by its total degree plus 1 (i.e. the number of linkages a firm has plus one). Firms industries are determined by SIC codes.

try clusters. In particular we tried uncovering customer-supplier relations by inferring the patterns of zeros in the time series of precision matrices. We believe we may need to include more information about each particular firm to be able to capture such type of relationships.

5.1. Future Work

In this section we provide some ideas about how we may improve the current paper

- In our current experiments we used a subset of data in which we have complete information about firms returns. However in the complete data set, there are many other firms for which there are missing return entries. Since they have missing entries we do not consider the existence of such firms. To include more firms then one may explore strategies like Kalman smoothing and low rank matrix completion.

- Use more information about each company to have a better description of each of them. We plan to consider data on earnings, cost of capital, profits before taxes and volume of each company. We also plan to control for time trends in data before using such information, e.g. using the Hodrick-Prescott filter (see (Hodrick & Prescott, 1997)). Moreover, to improve the performance of prediction of a firm return based on its linked companies one may want to explore the use of kernel trick.

- Finally, based on the results of the previous step we will assess whether using GGMs (with financial data) may help researchers to tackle the problem of opacity in uncovering business networks.

References

Acemoglu, Daron, Carvalho, Vasco M., Ozdaglar, Asuman, and Tahbaz-Salehi, Alireza. The network origins of aggregate fuctuations,. *Econometrica*, 80:1977–2016, 2012.

Ahern, Kenneth R. Network centrality and the cross section of stock returns,. *University of Southern California Working Paper*, 2013.

Boyd, Stephen P., Parikh, Neal, Chu, Eric, Peleato, Borja, and Eckstein, Jonathan. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011. ISSN 1935-8237. doi: 10.1561/2200000016. URL http://dx.doi.org/10.1561/2200000016.

Buraschi, A. and Porchia, P. Dynamic networks and asset pricing. *Social Science Research Network*, pp. 33, 2012.

Cohen, Lauren and Frazzini, Andrea. Economic links and predictable returns. *Journal of Finance*, 63(4):1977–2011, 2008.

Fama, Eugene and French, Kenneth. The cross-section of expected stock returns.

Friedman, Jerome H., Hastie, Trevor J., Höfling, H., and Tibshirani, Robert J. Pathwise coordinate optimization. *Ann. Appl. Stat.*, 1(2):302–332, 2007. ISSN 1932-6157. doi: 10.1214/07-AOAS131. URL http://dx.doi.org/10.1214/07-AOAS131.

Friedman, Jerome H., Hastie, Trevor J., and Tibshirani, Robert J. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Hodrick, Robert and Prescott, Edward C. Postwar u.s. business cycles: An empirical investigation,. *Journal of Money, Credit, and Banking*, 29:1–16, 1997.

Kolar, Mladen, Song, Le, Ahmed, Amr, and Xing, Eric P. Estimating Time-varying networks. *Ann. Appl. Stat.*, 4 (1):94–123, 2010.

Lafferty, John D., Liu, Han, and Wasserman, Larry A. Sparse nonparametric graphical models. *Statistical Science*, 27:519–537, 2012. doi: 10.1214/12-STS391. URL http://arxiv.org/pdf/1201.0794.pdf.

Meinshausen, Nicolas and Bühlmann, Peter. High dimensional graphs and variable selection with the lasso. *Ann. Stat.*, 34(3):1436–1462, 2006.

Wahlberg, Bo, Boyd, Stephen P., Annergren, Mariette, and Wang, Yang. An ADMM algorithm for a class of total variation regularized estimation problems. *arXiv preprint arXiv:1203.1828*, 2012.

Wu, J. and J.R.Birge. Supply chain network structure and firm returns. *Social Science Research Network*, pp. 51, 2014.

Yuan, M. and Lin, Y. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.