Large Scale Structure Learning of Conditional Gaussian Graphical Models

Manzil Zaheer (manzil@cmu.edu)
Carlos Ramírez (carlosrc@cmu.edu)
Soumya Batra (soumyabatra@cmu.edu)

Advised by Mladen Kolar (mkolar@chicagobooth.edu)

May 7, 2014
Motivation

- GGMs provide useful framework to represent relationships in complex systems.
- GGMs can be thought as undirected graphs of a set of random variables following a multivariate Gaussian distribution where
  - Nodes represent random variables
  - An edge between two nodes is absent if and only if the two r.v.s represented by those nodes are independent conditional on all other variables
- Modern financial markets are complex systems where uncovering relationships among firms may be of particular interest
  - Firms have become increasingly linked to each other through a complex and usually opaque network of relationships
  - Uncovering those relationships may be useful to predict future firm performance and thus firm prices and returns
Customer Supplier Network

Colors represent major industry clusters (first two digits of SIC code)
Problem Abstraction

Assumption

\[ X \mid Z = z \sim \mathcal{N}(\mu(z), \Sigma(z)) \]

- Goal: Learn the conditional independence relationships among components of vector \( X \) given \( Z = z \).
  - Let \( \Omega(z) = \Sigma(z)^{-1} = (\omega_{ab}(z))_{a,b \in [p] \times [p]} \)
  - Pattern of non-zero elements of this matrix encodes the conditional independencies

\[ X_a \perp X_b \mid X_{-ab}, Z = z \iff \omega_{ab}(z) = 0 \]

- 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 \( x_{ij} \)
Optimization Problem

\[
\min_{\Omega(\cdot) \in \mathcal{F}} \left\{ \sum_{i \in [n]} \left( \text{tr}(C_i \Omega(z_i)) - \log |\Omega(z_i)| + \mu \| \Omega(z_i) \|_1 \right) + \lambda \text{pen} \left( \{ \Omega(z_i) \}_{i \in [n]} \right) \right\}
\]

- Desired properties for penalty function
  - Perform model selection and control the smoothness of the estimator \( \omega_{ab}(z) \)
  - Encourage the successive function values \( \omega_{ab}(z_i), \omega_{ab}(z_{i+1}) \) to be close
- Our choice of penalty function:

\[
\text{pen} \left( \{ \Omega(z_i) \}_{i \in [n]} \right) = \sum_{i \in [n]} \left( \sqrt{\sum_{a,b} (\omega_{ab}(z_{i+1}) - \omega_{ab}(z_i))^2} \right)
\]
Parallelizing

Original Optimization Problem

\[
\min_{\Omega} \left\{ \sum_{i=1}^{n} (\text{tr}(C_i \Omega_i - \log |\Omega_i|) + \mu \|\Omega_i\|_1) + \lambda \sum_{i=1}^{n-1} \|\Omega_{i+1} - \Omega_i\|_F \right\}
\]

Rewrite for parallelization

\[
\min_{\Omega, R} \left\{ \sum_{i=1}^{n} (\text{tr}(C_i \Omega_i) - \log |\Omega_i| + \mu \|\Omega_i\|_1) + \lambda \sum_{i=1}^{n-1} \|R_i\|_F \right\}
\]

subject to: \( R_i = \Omega_{i+1} - \Omega_i \)

ADMM Form

Define constraint set \( \mathcal{C} = \{(\Omega, R) : R_i = \Omega_{i+1} - \Omega_i\} \)

\[
\min_{\Omega, R, W, S} \left\{ \sum_{i=1}^{n} (\text{tr}(C_i \Omega_i) - \log |\Omega_i| + \mu \|\Omega_i\|_1) + \lambda \sum_{i=1}^{n-1} \|R_i\|_F + I_{\mathcal{C}}(W, S) \right\}
\]

subject to: \( \Omega_i = W_i \quad R_i = S_i \)
ADMM Steps

1. 2n – 1 Independent Optimizations

\[ \Omega_{i}^{k+1} := \arg \min_{\Omega_i \succ 0} \left\{ \text{tr}(C_i\Omega_i) - \log |\Omega_i| + \frac{\rho}{2} \|\Omega_i - W_i^k + U_i^k\|_2^2 \right\} \]

\[ R_{i}^{k+1} := \arg \min_{R_i} \left\{ \lambda \|R_i\| + \frac{\rho}{2} \|R_i - S_i^k + T_i^k\|_2^2 \right\} \]

2. Projection onto the constraint set

\[ (W, S) := \Pi_C(\Omega_i^{k+1} + U_i^k, R_i^{k+1} + T_i^k) \]

3. Scaled dual variables update

\[ U_{i}^{k+1} := U_i^k + (\Omega_i^{k+1} - W_i^{k+1}) \]

\[ T_{i}^{k+1} := T_i^k + (R_i^{k+1} - S_i^{k+1}) \]
Distributed Implementation: Hadoop

Input Format 1
(path, file) → (i, [W_i, U_i, C_i])
i = \{1, ..., n\}

Mapper 1
(i, [W_i, U_i, C_i]) → (0, [U_i, C_i, \Omega_i])
i = \{1, ..., n\}

Reducer
(0, [U, C, \Omega]) + (0, [R, T]) → (\emptyset, [i, W_i, U_i, C_i, \Omega_i])
(0, [U, C, \Omega]) + (0, [R, T]) → (\emptyset, [i, R_i, S_i, T_i])

Input File 1
Output File 1

Input Format 2
(path, file) → (i, [R_i, S_i, T_i])
i = \{1, ..., n\}

Mapper 2
(i, [R_i, S_i, T_i]) → (0, [R_i, T_i])
i = \{1, ..., n\}

Output File 2
Estimation Quality 😞

Recall for years 1990–2000 combined using returns data directly

Recall for years 1990–2000 combined using residual data

Precision for years 1990–2000 combined using returns data directly

Precision for years 1990–2000 combined using residual data
Estimated graph for 1991
Conclusions

- Conditional GGMs provide an important tool to uncover relationships among different variables in complex systems.
- Using returns data the use of graphical models seems to uncover industry relationships among firms.
- However, conditional GGMs does not provide further information about the nature of such relationships (besides the identification of industry clusters).
  - We may need to include more information about firms to understand better the nature of such relationships.

Future Work

- Handle missing data
- Feature engineering
Thank You