High Dimensional Models

Time-Varying Graphical Lasso

Andrew Boomer & Jacob Pichelmann

Toulouse School of Economics
M2 EEE

March 17, 2021

Overview

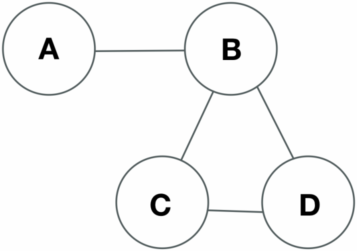
1. Introduction to Graphical Models
   - Important Properties
   - Interpretation
2. Gaussian Graphical Model
3. Time Varying Graphical Lasso (TGLV)
   - Altered Optimisation Problem
   - ADMM
4. Practical Application of TGVL
   - Comparison to Static Graphical Lasso
   - Changing the penalty function

Graphical Models

- Graphical models offer a way to encode conditional dependencies between p random variables $X_1, \cdots, X_p$ by a graph g
- A graph consists of a vertex set $V = \{1, 2, \cdots, p\}$ and an edge set $E \subset V \times V$
- We focus on undirected graphical models, i.e. no distinction between an edge $(s, t) \in E$ and the edge (t, s).

Consider the following example:

Figure: Undirected Graphical Model



Factorization Property

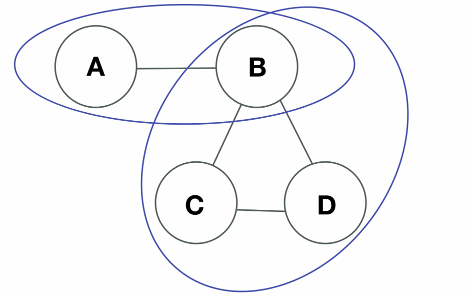
A graph clique $C \subseteq V$ is a fully-connected subset of the vertex set, i.e. $(s,t) \in E \forall s,t \in C$. (?, ?)

$$\mathbb{P}(A,B,C,D) \propto \phi(A,B)\phi(B,C,D)$$

$$\mathbb{P}(X) = \frac{1}{Z}\prod_{c \in C} \phi c(x_c)$$

where $Z = \sum_{x \in X^p} \prod_{c \in C} \phi_c(x_c)$.

Figure: Maximal Cliques

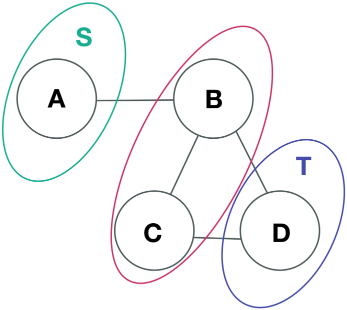


Markov Property

Any two subsets S and T are conditionally independent given a separating subset Y. A random vector X is Markov with respect to g if

$$X_S \perp\!\!\!\perp X_T | X_Y \text{ for all cut sets } S \subset V.$$

Figure: Separating Set: $\{B, C\}$



Equivalence of Properties

- Hammersley-Clifford theorem:

  For any strictly positive distribution the distribution of X factorizes according to the graph g if and only if the random vector X is Markov with respect to the graph. (?, ?)

Gaussian Graphical Model

X follows a Gaussian distribution:

$$X \sim \mathcal{N}(\mu, \Sigma)$$

If Σ is positive definite, distribution has density on $\mathbb{R}^p$

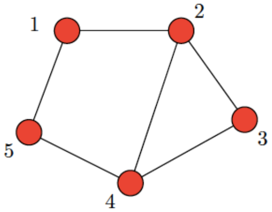
$$f(x \mid \mu, \Sigma) = (2\pi)^{-p/2} (\det \Theta)^{1/2} e^{-(x-\mu)^T \Theta (x-\mu)/2}$$

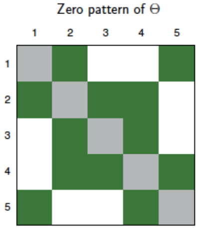
where $\Theta = \Sigma^{-1}$ is the **Precision matrix** of the distribution.

Empirical covariance $S = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)'$

Gaussian Graphical Model

We can represent a multivariate Gaussian distribution as a graphical model. Whenever X factorizes according to the graph g we must have $\Theta_{st} = 0$ for any pair $(s, t) \notin E$. This gives a correspondence between the zero pattern of Θ and the edge structure of g.





Estimating the graph structure $\Leftrightarrow \Theta$

- Suppose X denotes samples from a multivariate Gaussian distribution with $\mu = 0$ and precision matrix $\Theta \in \mathbb{R}^{p \times p}$
- We can write the log-likelihood of the multivariate Gaussian as

$$\mathcal{L}(\Theta; X) = \frac{1}{N} \sum_{i=1}^{N} \log \mathbb{P}_{\Theta}(xi) = \log \det \Theta - \mathrm{trace}(S\Theta)$$

- So why not just estimate by MLE to obtain $\widehat{\Theta}_{ML}$?
  1. A sparse graph increases interpretability, prevents overfitting.
  2. In real world applications often times $p > N$, then MLE solution does not exist.

ℓ_1 Norm Regularisation

Sparsity can be achieved by adding a penalty term to the optimisation problem. Using the ℓ_1 norm yields the familiar lasso estimator.

$$\hat{\Theta} = \operatorname{argmin}_{\Theta \geq 0} \left(\operatorname{tr}(S\Theta) - \log \det(\Theta) + \lambda \, ||\Theta||_{\mathrm{od},1} \right)$$

where $||\Theta||_{\mathrm{od},1}$ is the ℓ_1-norm of the off-diagonal entries of Θ.

Challenge: The Network Structure Can Change Over Time

In many real world settings (e.g. financial markets) the structure of the complex system changes over time.

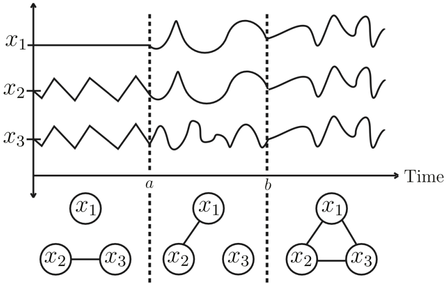


Figure: Example of Changing Network Structure (?, ?)

Solution: Optimization on a Chain Graph (TVGL)

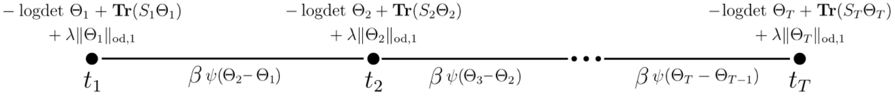


Figure: (?, ?)

The optimization problem becomes

$$\underset{\Theta \in S^p_{++}}{\text{minimize}} \quad \sum_{i=1}^{T} -l_i\left(\Theta_i\right) + \lambda \|\Theta_i\|_{\text{od},1} + \beta \sum_{i=2}^{T} \psi\left(\Theta_i - \Theta_{i-1}\right)$$

where β determines how strongly correlated neighboring covariance estimations should be. A small β will lead to θ's which fluctuate from estimate-to-estimate, whereas large β's lead to smoother estimates over time.

Choice of ψ

- ψ allows to enforce different behaviors in the evolution of the network structure
- Expectations how the underlying network may change over time can be encoded into ψ

Options:

- **Global restructuring** - $\psi(X) = \sum_j \|[X]_j\|_2$
- **Smoothly varying over time** - $\psi(X) = \sum_{i,j} X_{i,j}^2$
- **Perturbed node** - $\psi(X) = \min_{V:V+V^T=X} \sum_j \left\|[V]_j\right\|_2$

Optimization Algorithm: ADMM

- The authors use ADMM (alternating direction method of multipliers) to solve the TVGL optimization problem.
- ADMM is an general optimization technique that can be used on any convex optimization problem.
- ADMM has a couple main advantages compared to standard gradient descent based methods: (1) Can be applied to nonsmooth functions, (2) Can be distributed across multiple independent machines
- To put ADMM into context, we show how it can be used to solve a generic optimization problem

Optimization Algorithm: ADMM

General Example

We can take the generic minimization problem

$$\underset{x}{\operatorname{argmin}}\, f(x) \quad \text{s.t. } x \in \mathcal{C}$$

And separate it into two functions, f and g, where g is the indicator of $\mathcal{C}$

$$\underset{x}{\operatorname{argmin}}\, f(x) + g(z) \quad \text{s.t. } x - z = 0$$

The variable z is known as a consensus variable, and the constraint ensures final convergence between x and z

Optimization Algorithm: ADMM

Proximal Operators/Proximal Gradient Descent

The generality of the ADMM optimization technique relies on the method of proximal gradient descent. Proximal gradient descent makes use of proximal operators, defined as:

$$\mathrm{prox}_{\lambda f}(v) = \underset{x}{\mathrm{argmin}}\left(f(x) + (1/2\lambda)\|x - v\|_2^2\right)$$

The ADMM iteration based update method is:

$$\begin{aligned}
x^{k+1} &:= \underset{x}{\mathrm{argmin}}\left(f(x) + (\rho/2)\left\|x - z^k + u^k\right\|_2^2\right) \\
z^{k+1} &:= \Pi_C\left(x^{k+1} + u^k\right) \\
u^{k+1} &:= u^k + x^{k+1} - z^{k+1}
\end{aligned}$$

Iterations stop when $u^k \to u^{k+1}$ ($x - z = 0$ constraint satisfied)

GlASSO vs. TVGL

Importance of ψ

- Choice of ψ relies on knowledge about network behavior
- No a priori decision possible
- ψ is fixed over time

We illustrate the importance of the choice of ψ by replicating the author's case studies with different penalty functions.

Changing ψ

References