ETH zürich

Graphs and Model Selection

Muriel Egli

Contents

Basics of Graphical Models

- A graph $G = (V, E)$ consists of a set of vertices V and a set of edges E
- We focus exclusively on undirected graphs
- We can associate a collectioin of random variables $X = (X_1, X_2, \ldots, X_p)$ with the vertex set $V = \{1, 2 \ldots, p\}$ of some underlying graph
- Idea: see the structure of the underlying graph as a visual representation of the joint distribution of the random variables

Factorization Property

- Let $\mathcal{C}$ be the set of all cliques in the graph G
- For a clique $C \in \mathcal{C}$ a compatibility function ψ_C is a function of the subvector $x_C := (x_s, s \in C)$ taking positive real values
- Given a collection of compatibility functions we say that a probability distribution P factorizes over G if and only if

$$P(x_1, \ldots, x_p) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

  where $Z = \sum_{x \in \chi^p} \prod_{c \in \mathcal{C}} \psi_C(x_C)$ ensures that P is properly normalized
- Such a factorization can lead to savings in storage and computation if the clique sizes are not too large

Factorization Property

- Example:

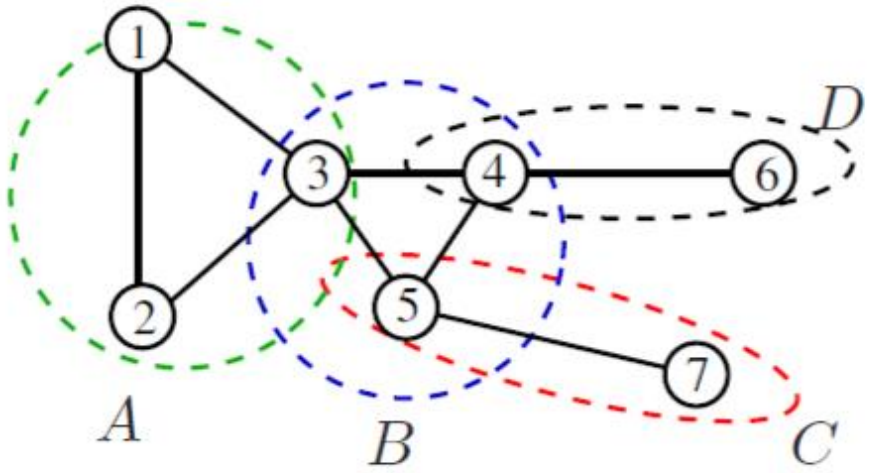


Figure: Trevor Hastie, Robert Tibshirani and Martin Wainwright (2015). Statistical Learning with Sparsity: The Lasso and Generalizations, p242.

- P factorizes over this graph if it has the form

$$P(x_1, \ldots, x_7) = \frac{1}{Z}\psi_A(x_1, x_2, x_3)\psi_B(x_3, x_4, x_5)\psi_D(x_4, x_6)\psi_C(x_5, x_7)$$

For some choice of compatility functions $\{\psi_A, \psi_B, \psi_C, \psi_D\}$

Markov Property

- Let S denote a cut set disconnectiong the graph into components A and B
- We say that a random vector X is Markov with respect to the graph G if

$$X_A \perp\!\!\!\perp X_B \mid X_S \text{ for all cut sets } S \subset V$$

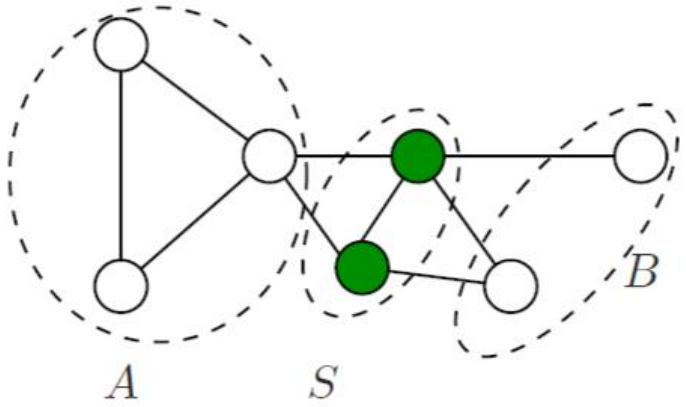


Figure: Trevor Hastie, Robert Tibshirani and Martin Wainwright (2015). Statistical Learning with Sparsity: The Lasso and Generalizations, p242.

Hammersley-Clifford Theorem

- This is the fundamental theorem of random fields and gives necessary and sufficient conditions under which a strictly positive probability distribution can be represented as a Markov network
- **Theorem:** For a strictly positive probability distribution P of a random vector X the two characterizations are equivalent; the distribution of X factorizes according to the graph G if and only if it is Markov with respect to G.

Gaussian Graphical Model

- Given a p dimensional Gaussian distribution with mean vector μ and covariance matrix $\mathbf{\Sigma}$:

$$P_{\mu,\mathbf{\Sigma}}(x) = \frac{1}{(2\pi)^{\frac{p}{2}} \det(\mathbf{\Sigma})^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^T \mathbf{\Sigma}^{-1}(x-\mu)}$$

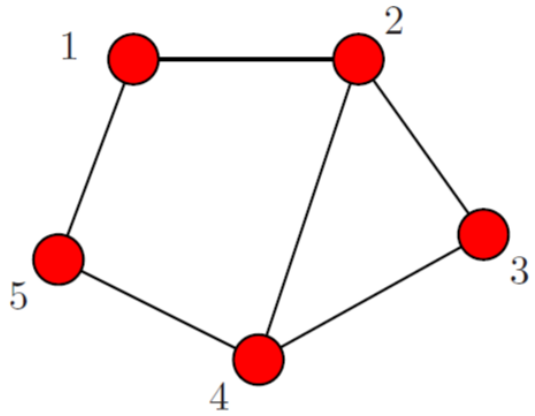
- This can equivalently be formulated as

$$P_{\gamma,\mathbf{\Theta}}(x) = \exp\{\sum_{s=1}^{p} \gamma_s x_s - \frac{1}{2} \sum_{s,t=1}^{p} \boldsymbol{\theta}_{st} x_s x_t - A(\mathbf{\Theta})\}$$

where $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$ the precision matrix, $\gamma = \mathbf{\Theta}\mu$ and $A(\mathbf{\Theta}) = -\frac{1}{2} \log \det(\mathbf{\Theta}/(2\pi))$

Gaussian Graphical Model

- This new representation allows us to discuss factorization properties in terms of the sparsity pattern of $\boldsymbol{\Theta}$
- If X factorizes according to the graph G then for $(s,t) \notin E$ we must have that $\boldsymbol{\theta}_{st} = 0$
- Example:



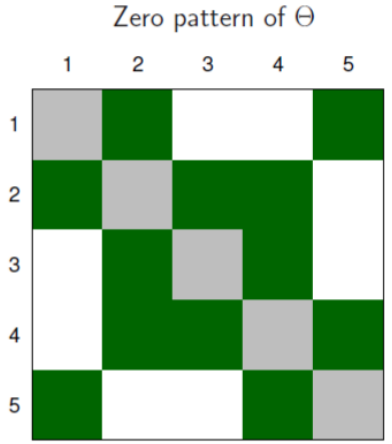


Figure: Trevor Hastie, Robert Tibshirani and Martin Wainwright (2015). Statistical Learning with Sparsity: The Lasso and Generalizations, p246.

Graph Selection

Problem: Given a collection of samples from a graphical model, where the underlying graph structure is unknown. How can we find the correct graph with high probability?

Graph Selection for Gaussian Graphical Models

- Suppose $\mathbf{X}$ represents samples from a zero-mean multivariate Gaussian distribution with unknown precision matrix $\mathbf{\Theta}$
- One can show that the log-likelihood of this distribution takes the form

$$\mathcal{L}(\mathbf{\Theta}, \mathbf{X}) = \frac{1}{N} \sum_{i=1}^{N} \log P_{\mathbf{\Theta}}(x_i) = \log\det \mathbf{\Theta} - \operatorname{trace}(\mathbf{S\Theta})$$

where $\mathbf{S} = \frac{1}{N} \sum_{i=1}^{N} x_i x_i^T$ the empirical covariance matrix and

$$\log\det(\mathbf{\Theta}) = \begin{cases} \sum_{j=1}^{p} \log(\lambda_j(\mathbf{\Theta})) & \text{if } \mathbf{\Theta} \succ 0 \\ -\infty & \text{otherwise} \end{cases}$$

where $\lambda_j(\mathbf{\Theta})$ is the j-th eigenvalue of $\mathbf{\Theta}$

Graph Selection for Gaussian Graphical Models

- This function is strictly concave, so that if the maximum is achieved it must be unique and defines the maximum likelihood estimate $\hat{\mathbf{\Theta}}$
- If we let N go to infinity $\hat{\mathbf{\Theta}}$ converges to the true precision matrix
- But if $N < p$ no maximum likelihood estimator exists and we need to consider suitably constrained or regularized forms
- If we are seeking Graphical models based on sparse graphs we could consider the following convex optimization problem

$$\hat{\mathbf{\Theta}} \in \arg\max_{\mathbf{\Theta} \succeq 0} \{\log\det \mathbf{\Theta} - \mathrm{trace}(\mathbf{S}\mathbf{\Theta}) - \lambda \rho_1(\mathbf{\Theta})\}$$

where $\rho_1(\mathbf{\Theta}) = \sum_{s \neq t} |\boldsymbol{\theta}_{st}|$

Graphical Lasso Algorithm

- The subgradient equation corresponding to this problem is given by

$$\mathbf{\Theta}^{-1} - \mathbf{S} - \lambda \mathbf{\Psi} = \mathbf{0}$$

where $\mathbf{\Psi}$ has diagonal entries 0, $\psi_{jk} = \text{sign}(\theta_{jk})$ if $\theta_{jk} \neq 0$ and $\psi_{jk} \in [-1, 1]$ if $\theta_{jk} = 0$

- To solve this problem via blockwise coordinate descent we partition the matrices into one column versus the rest: $\mathbf{\Theta} = \begin{bmatrix} \mathbf{\Theta}_{11} & \boldsymbol{\theta}_{12} \\ \boldsymbol{\theta}_{12}^T & \theta_{22} \end{bmatrix}$, $\mathbf{S} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{s}_{12} \\ \mathbf{s}_{12}^T & s_{22} \end{bmatrix}$

Graphical Lasso Algorithm

- Then $\mathbf{W} = \mathbf{\Theta}^{-1} = \begin{bmatrix} \mathbf{W}_{11} & \mathbf{w}_{12} \\ \mathbf{w}_{21} & w_{22} \end{bmatrix} = \begin{bmatrix} (\mathbf{\Theta}_{11} - \frac{\boldsymbol{\theta}_{12}\boldsymbol{\theta}_{21}}{\theta_{22}})^{-1} & -\mathbf{W}_{11}\frac{\boldsymbol{\theta}_{12}}{\theta_{22}} \\ . & . \end{bmatrix}$
- So for the last column of our subgradient equation we get:

$$\mathbf{w}_{12} - \mathbf{s}_{12} + \lambda\boldsymbol{\psi}_{12} = \mathbf{W}_{11}\boldsymbol{\beta} - \mathbf{s}_{12} + \lambda\boldsymbol{\psi}_{12} = 0$$

  where $\boldsymbol{\beta} = -\boldsymbol{\theta}_{12}/\theta_{22}$

- It can be seen that this is equivalent to a modified version of the estimating equations for a lasso regression

Graphical Lasso Algorithm

- Recall that in the usual regression setup with outcome $\mathbf{y}$ and predictor matrix $\mathbf{Z}$ the lasso minimizes $\frac{1}{N}\|\mathbf{y} - \mathbf{Z}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1$
- This has the subgradient equations $\frac{1}{N}\mathbf{Z}^T\mathbf{Z}\boldsymbol{\beta} - \frac{1}{N}\mathbf{Z}^T\mathbf{y} + \lambda\,\mathrm{sign}(\boldsymbol{\beta}) = \mathbf{0}$
- Comparing to the last column of our subgradient equation shows that $\frac{1}{N}\mathbf{Z}^T\mathbf{y}$ corresponds to $\mathbf{s}_{12}$ and $\frac{1}{N}\mathbf{Z}^T\mathbf{Z}$ corresponds to $\mathbf{W}_{11}$
- Hence we want to minimize $\frac{1}{2}\|\mathbf{W}_{11}^{\frac{1}{2}}\boldsymbol{\beta} - \mathbf{W}_{11}^{-\frac{1}{2}}\mathbf{s}_{12}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1$

Graphical Lasso Algorithm

Algorithm 9.1 GRAPHICAL LASSO.

1. Initialize $\mathbf{W} = \mathbf{S}$. Note that the diagonal of $\mathbf{W}$ is unchanged in what follows.
2. Repeat for $j = 1, 2, \ldots p, 1, 2, \ldots p, \ldots$ until convergence:
   (a) Partition the matrix $\mathbf{W}$ into part 1: all but the j^{th} row and column, and part 2: the j^{th} row and column.
   (b) Solve the estimating equations $\mathbf{W}_{11}\boldsymbol{\beta} - \mathbf{s}_{12} + \lambda \cdot \text{sign}(\boldsymbol{\beta}) = 0$ using a cyclical coordinate-descent algorithm for the modified lasso.
   (c) Update $\mathbf{w}_{12} = \mathbf{W}_{11}\hat{\boldsymbol{\beta}}$
3. In the final cycle (for each j) solve for $\hat{\boldsymbol{\theta}}_{12} = -\hat{\boldsymbol{\beta}} \cdot \hat{\theta}_{22}$, with $1/\hat{\theta}_{22} = w_{22} - \mathbf{w}_{12}^T\hat{\boldsymbol{\beta}}$.

Figure: Trevor Hastie, Robert Tibshirani and Martin Wainwright (2015). Statistical Learning with Sparsity: The Lasso and Generalizations, p250.

Graphical Lasso Algorithm

- Example in R
  - glasso package
  - manual graphical lasso
  - plotting the coefficients for different values of λ

Graphical Lasso Algorithm

- If we repeat the algorithm for a range of different values for λ we can plot the estimates for the entires of the precision matrix against $\rho_1(\boldsymbol{\Theta})$
- **Example:** Here the true precision matrix is

$$\boldsymbol{\Theta} = \begin{bmatrix} 2 & 0.6 & 0 & 0 & 0.5 \\ 0.6 & 2 & -0.4 & 0.3 & 0 \\ 0 & -0.4 & 2 & -0.2 & 0 \\ 0 & 0.3 & -0.2 & 2 & 0 \\ 0.5 & 0 & 0 & 0 & 2 \end{bmatrix}$$

- If we simulate data from the multivariate gaussian with $\boldsymbol{\Theta}$ the true values are achieved at the right side of the plot

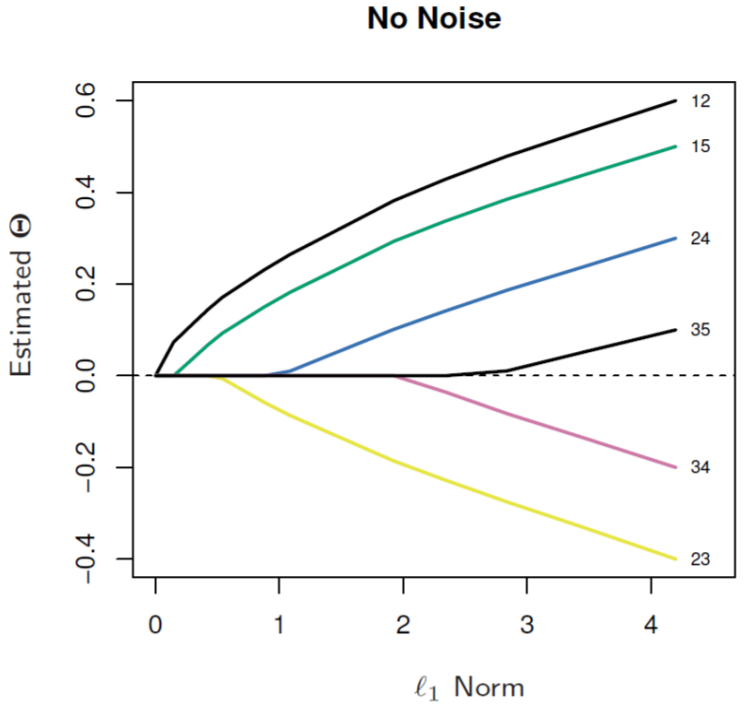


Figure: Trevor Hastie, Robert Tibshirani and Martin Wainwright (2015). Statistical Learning with Sparsity: The Lasso and Generalizations, p251.

Graphical Lasso Algorithm

- However if we add some standard Gaussian noise to each column the true edge set is not recovered for any value of λ

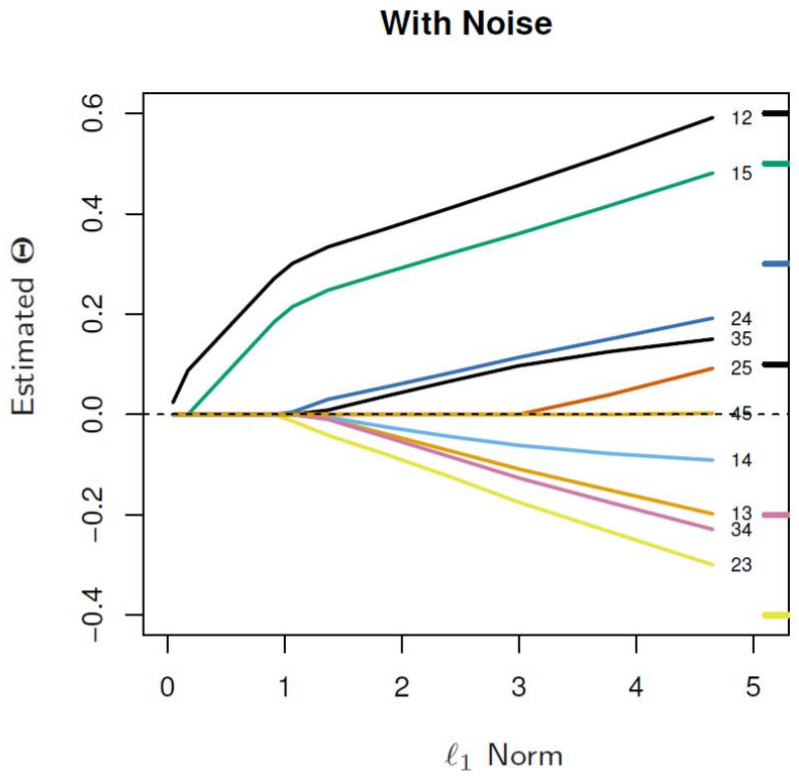


Figure: Trevor Hastie, Robert Tibshirani and Martin Wainwright (2015). Statistical Learning with Sparsity: The Lasso and Generalizations, p251.

Theoretical Guarantees for Graphical Lasso

- Plot of the operator norm $\|\hat{\mathbf{\Theta}} - \mathbf{\Theta}\|_2$ versus the sample size N for three different graph sizes where $\lambda_N = 2\sqrt{\frac{\log p}{N}}$ was used as the regularization parameter
- We see that larger graphs require more samples for a consistent estimation

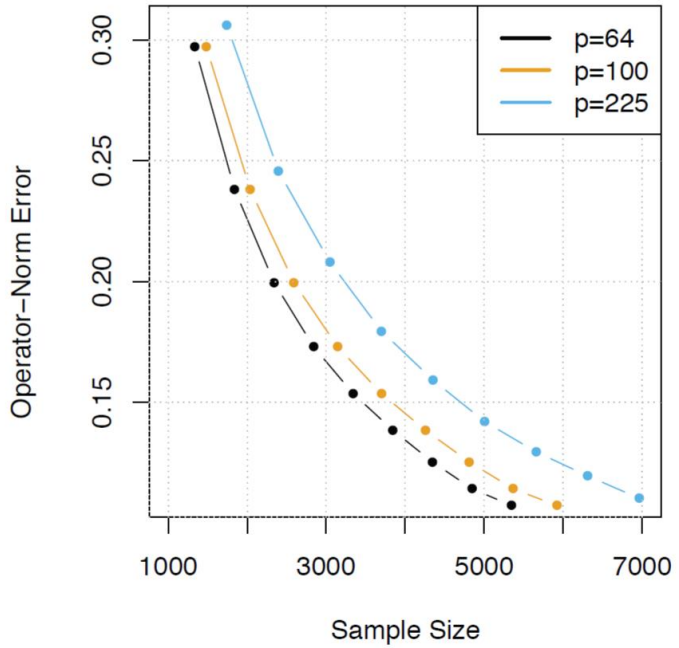


Figure: Trevor Hastie, Robert Tibshirani and Martin Wainwright (2015). Statistical Learning with Sparsity: The Lasso and Generalizations, p252.

Neighborhood Selection

- High-dimensional Graphs and Variable Selection with the Lasso; Meinshausen and Buhlmann (2006)
- It is an alternative method for graph selection that is computationally efficient and consistent for high dimensional graphs
- For a random vector $X = (X_1, \ldots, X_p)$ consider the conditional distribution of X_s given the random vector $X_{\setminus\{s\}} = (X_1, \ldots, X_{s-1}, X_{s+1}, \ldots, X_p)$
- By the properties of a graphical model the only relevant variables are those in the neighborhood set $\mathcal{N}(s)$

Neighborhood Selection for Gaussians

- In the case of a multivarite Gaussian the conditional distribution of X_s given $X_{\setminus\{s\}}$ is

$$X_s = X_{\setminus\{s\}}\beta^s + W_{\setminus\{s\}}$$

  where $W_{\setminus\{s\}}$ corresponds to a prediction error independent of $X_{\setminus\{s\}}$

- The key property is that the regression vector β^s satisfies $\mathrm{supp}(\beta^s) = \mathcal{N}(s)$

- It is natural to estimate β via the lasso

Neighborhood Selection

Algorithm 9.2 NEIGHBORHOOD-BASED GRAPH SELECTION FOR GAUSSIAN GRAPHICAL MODELS.

1. For each vertex $s = 1, 2, \ldots, p$:
   (a) Apply the lasso to solve the neighborhood prediction problem:

$$\widehat{\beta}^s \in \underset{\beta^s \in \mathbb{R}^{p-1}}{\arg\min} \left\{ \frac{1}{2N} \sum_{i=1}^{N} (x_{is} - x_{i,V\setminus\{s\}}^T \beta^s)^2 + \lambda \|\beta^s\|_1 \right\}. \tag{9.25}$$

   (b) Compute the estimate $\widehat{\mathcal{N}}(s) = \text{supp}(\widehat{\beta}^s)$ of the neighborhood set $\mathcal{N}(s)$.
2. Combine the neighborhood estimates $\{\widehat{\mathcal{N}}(s), s \in V\}$ via the AND or OR rule to form a graph estimate $\widehat{G} = (V, \widehat{E})$.

Figure: Trevor Hastie, Robert Tibshirani and Martin Wainwright (2015). Statistical Learning with Sparsity: The Lasso and Generalizations, p256.

Neighborhood Selection

- Example in R unsing the glasso package setting approx=TRUE

Questions?