Statistical Learning with Sparsity

Graphical Models

2019 Wald Lecture III

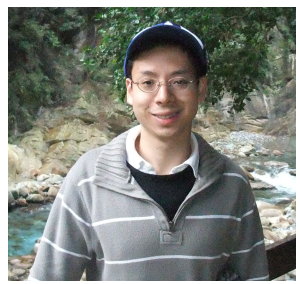
Trevor Hastie
Department of Statistics
Department of Biomedical Data Science
Stanford University



Outline

- What are graphical models?
- Review of Gaussian graphical models and `glasso`

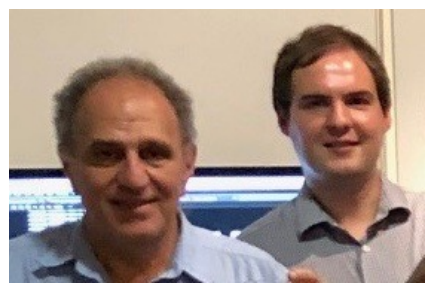


with Friedman, Tibshirani, Mazumder and Lee

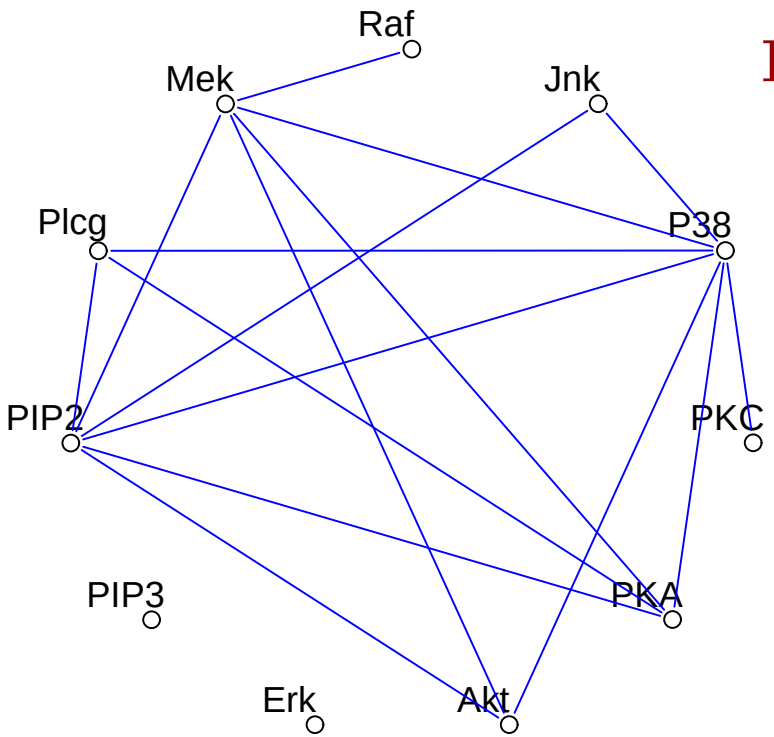
- Anomaly detection and missing-value imputation





with Moehle, Boyd, Candes, Tibshirani, McNeill and the Blackrock AI lab in Palo Alto

Conditional Independence Graphs



Flow Cytometry (Sachs et al, 2003)

11 proteins measured on 7466 cells.

Shown is a fitted *undirected* graphical model or *Markov Network*.

`Raf` and `Jnk` are conditionally independent, given the rest.

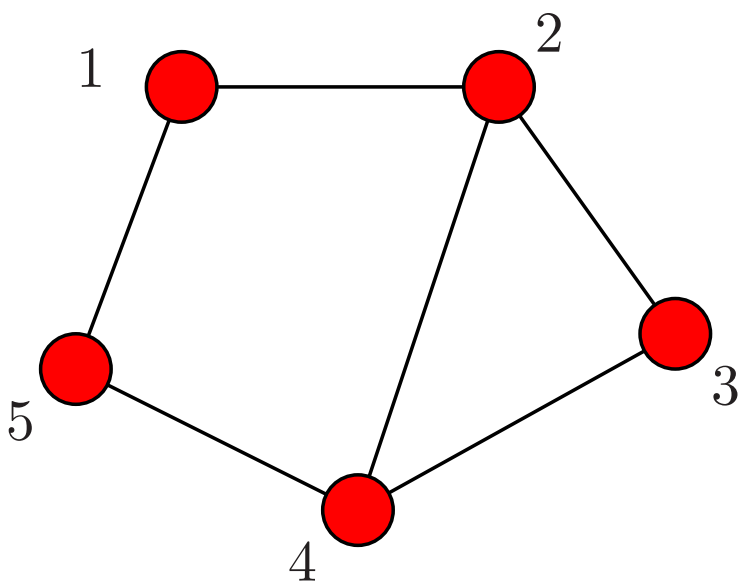
`PIP3` is independent of everything else, as is `Erk`.

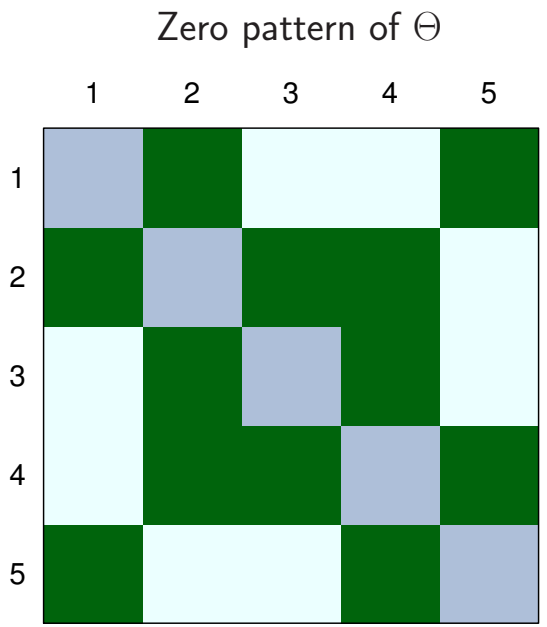
The model was estimated using the *graphical lasso*.

Gaussian Graphical Models

Suppose all the variables $X = X_1, \ldots, X_p$ in a graph are Gaussian, with joint density $X \sim N(0, \mathbf{\Sigma})$.

$\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$ encodes all the conditional dependencies.





Gaussian Graphical Models — in detail

Suppose all the variables $X = X_1, \ldots, X_p$ in a graph are Gaussian, with joint density $X \sim N(0, \mathbf{\Sigma})$
Let $X = (Z, Y)$ where $Z = (X_1, \ldots, X_{p-1})$ and $Y = X_p$. Then with

$$\mathbf{\Sigma} = \begin{pmatrix} \mathbf{\Sigma}_{ZZ} & \boldsymbol{\sigma}_{ZY} \\ \boldsymbol{\sigma}_{ZY}^\top & \sigma_{YY} \end{pmatrix},$$

we can write the conditional distribution of Y given Z (the rest) as

$$Y|Z = z \ \sim N\left(z^\top \mathbf{\Sigma}_{ZZ}^{-1} \boldsymbol{\sigma}_{ZY},\ \sigma_{YY} - \boldsymbol{\sigma}_{ZY}^\top \mathbf{\Sigma}_{ZZ}^{-1} \boldsymbol{\sigma}_{ZY}\right)$$

- The regression coefficients $\boldsymbol{\beta} = \mathbf{\Sigma}_{ZZ}^{-1} \boldsymbol{\sigma}_{ZY}$ determine the conditional (in)dependence structure.
- In particular, if $\beta_j = 0$, then Y and Z_j are conditionally independent, given the rest.

$\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$ and conditional dependence structure

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{ZZ} & \boldsymbol{\sigma}_{ZY} \\ \boldsymbol{\sigma}_{ZY}^\top & \sigma_{YY} \end{bmatrix} \qquad \boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\Theta}_{ZZ} & \boldsymbol{\theta}_{ZY} \\ \boldsymbol{\theta}_{ZY}^\top & \theta_{YY} \end{bmatrix}$$

$\boldsymbol{\Sigma}\boldsymbol{\Theta} = \mathbf{I}$ and part of the partitioned inverse of $\boldsymbol{\Theta}$ looks like

$$\begin{bmatrix} \boldsymbol{\Theta}_{ZZ} & \boldsymbol{\theta}_{ZY} \\ \boldsymbol{\theta}_{ZY}^\top & \theta_{YY} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Theta}_{ZZ} & -\theta_{YY}\boldsymbol{\Sigma}_{ZZ}^{-1}\boldsymbol{\sigma}_{ZY} \\ \boldsymbol{\theta}_{ZY}^\top & (\sigma_{YY} - \boldsymbol{\sigma}_{YZ}\boldsymbol{\Sigma}_{ZZ}^{-1}\boldsymbol{\sigma}_{ZY})^{-1} \end{bmatrix}$$

$\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$ and conditional dependence structure

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{ZZ} & \boldsymbol{\sigma}_{ZY} \\ \boldsymbol{\sigma}_{ZY}^\top & \sigma_{YY} \end{bmatrix} \qquad \boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\Theta}_{ZZ} & \boldsymbol{\theta}_{ZY} \\ \boldsymbol{\theta}_{ZY}^\top & \theta_{YY} \end{bmatrix}$$

$\boldsymbol{\Sigma}\boldsymbol{\Theta} = \mathbf{I}$ and part of the partitioned inverse of $\boldsymbol{\Theta}$ looks like

$$\begin{bmatrix} \boldsymbol{\Theta}_{ZZ} & \boldsymbol{\theta}_{ZY} \\ \boldsymbol{\theta}_{ZY}^\top & \theta_{YY} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Theta}_{ZZ} & -\theta_{YY}\boldsymbol{\Sigma}_{ZZ}^{-1}\boldsymbol{\sigma}_{ZY} \\ \boldsymbol{\theta}_{ZY}^\top & (\sigma_{YY} - \boldsymbol{\sigma}_{YZ}\boldsymbol{\Sigma}_{ZZ}^{-1}\boldsymbol{\sigma}_{ZY})^{-1} \end{bmatrix}$$

$$\begin{aligned} \boldsymbol{\theta}_{ZY} &= -\theta_{YY} \cdot \boldsymbol{\beta}_{Y|Z} \\ 1/\theta_{YY} &= \sigma_{YY} - \boldsymbol{\sigma}_{ZY}^\top \boldsymbol{\Sigma}_{ZZ}^{-1} \boldsymbol{\sigma}_{ZY} \end{aligned}$$

$\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$ and conditional dependence structure

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{ZZ} & \boldsymbol{\sigma}_{ZY} \\ \boldsymbol{\sigma}_{ZY}^\top & \sigma_{YY} \end{bmatrix} \qquad \boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\Theta}_{ZZ} & \boldsymbol{\theta}_{ZY} \\ \boldsymbol{\theta}_{ZY}^\top & \theta_{YY} \end{bmatrix}$$

$\boldsymbol{\Sigma}\boldsymbol{\Theta} = \mathbf{I}$ and part of the partitioned inverse of $\boldsymbol{\Theta}$ looks like

$$\begin{bmatrix} \boldsymbol{\Theta}_{ZZ} & \boldsymbol{\theta}_{ZY} \\ \boldsymbol{\theta}_{ZY}^\top & \theta_{YY} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Theta}_{ZZ} & -\theta_{YY}\boldsymbol{\Sigma}_{ZZ}^{-1}\boldsymbol{\sigma}_{ZY} \\ \boldsymbol{\theta}_{ZY}^\top & (\sigma_{YY} - \boldsymbol{\sigma}_{YZ}\boldsymbol{\Sigma}_{ZZ}^{-1}\boldsymbol{\sigma}_{ZY})^{-1} \end{bmatrix}$$

$$\begin{aligned} \boldsymbol{\theta}_{ZY} &= -\theta_{YY} \cdot \boldsymbol{\beta}_{Y|Z} \\ 1/\theta_{YY} &= \sigma_{YY} - \boldsymbol{\sigma}_{ZY}^\top \boldsymbol{\Sigma}_{ZZ}^{-1} \boldsymbol{\sigma}_{ZY} \end{aligned}$$

Hence $\boldsymbol{\Theta}$ contains *all* the conditional dependence information for the multivariate Gaussian model.
In particular, any $\theta_{ij} = 0$ implies conditional independence of X_i and X_j, given the rest.

Estimating $\boldsymbol{\Theta}$ by Gaussian Maximum Likelihood

Given a sample $x_i,\ i = 1, \ldots, N$ we can write down the Gaussian log-likelihood of the data (ignoring constants):

$$\ell(\boldsymbol{\Sigma}; \{x_i\}) = -\log\det(\boldsymbol{\Sigma}) - \frac{1}{N}\sum_{i=1}^{N} x_i^\top \boldsymbol{\Sigma}^{-1} x_i$$

Setting $\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N} x_i x_i^\top$, we get (up to constants)

$$\ell(\boldsymbol{\Theta}; \mathbf{S}) = \log\ \det\boldsymbol{\Theta} - \mathrm{trace}(\mathbf{S}\boldsymbol{\Theta})$$

and (by some miracle!)

$$\frac{d\ell(\boldsymbol{\Theta}; \mathbf{S})}{d\boldsymbol{\Theta}} = \boldsymbol{\Theta}^{-1} - \mathbf{S}.$$

Hence $\widehat{\boldsymbol{\Theta}} = \mathbf{S}^{-1}$ and of course $\widehat{\boldsymbol{\Sigma}} = \mathbf{S}$.

Solving for $\widehat{\Theta}$ through regression

Since

$$\begin{aligned} \boldsymbol{\theta}_{ZY} &= -\theta_{YY} \cdot \boldsymbol{\Sigma}_{ZZ}^{-1} \boldsymbol{\sigma}_{ZY} \\ &= -\theta_{YY} \cdot \boldsymbol{\beta}_{Y|Z}. \\ 1/\theta_{YY} &= \sigma_{YY} - \boldsymbol{\sigma}_{ZY}^{\top} \boldsymbol{\Sigma}_{ZZ}^{-1} \boldsymbol{\sigma}_{ZY} \end{aligned}$$

we can solve for $\widehat{\boldsymbol{\Theta}} = \mathbf{S}^{-1}$ one column at a time via linear regressions, replacing population moments $\boldsymbol{\Sigma}$ by their sample equivalents $\mathbf{S}$:

- $\hat{\boldsymbol{\theta}}_{ZY}$ is (up to scalar $-\theta_{YY}$) the regression coefficient of the y_i on the z_i;
- $\hat{\theta}_{YY}$ is the inverse of the mean-squared residual.

Do this p times, treating each variable as *response*.

Summary so far

- For Gaussian variables, conditional independence is captured in the regression coefficients.
- The precision matrix $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$ explicitly represents all these regression coefficients.
- Any off-diagonal zeroes in $\mathbf{\Theta}$ means conditional independence.
- We can estimate/compute $\widehat{\Theta} = \mathbf{S}^{-1}$ using N data vectors x_i via p linear regressions.

Summary so far

- For Gaussian variables, conditional independence is captured in the regression coefficients.
- The precision matrix $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$ explicitly represents all these regression coefficients.
- Any off-diagonal zeroes in $\mathbf{\Theta}$ means conditional independence.
- We can estimate/compute $\widehat{\Theta} = \mathbf{S}^{-1}$ using N data vectors x_i via p linear regressions.
- *Natural to look for zeros through sparse regression and variable selection procedures!*

Structure inference through regression

- Fit regressions of each variable X_j on the rest.
- Do variable selection to decide which coefficients should be zero.
- Meinshausen and Bühlmann (2006) use *lasso* regressions to achieve this.

Structure inference through regression

- Fit regressions of each variable X_j on the rest.
- Do variable selection to decide which coefficients should be zero.
- Meinshausen and Bühlmann (2006) use *lasso* regressions to achieve this.

Problem:

- in Gaussian model, if X_j is conditionally independent of X_i, given the rest, then $\beta_{ji} = 0$.
- But then X_i is conditionally independent of X_j, given the rest, and $\beta_{ij} = 0$ as well.
- Adaptive regression methods don't automatically honor this symmetry.

Solving for $\boldsymbol{\Theta}$ when zero structure is known

Suppose we know what edges are missing. We add Lagrange terms to the log-likelihood corresponding to the missing edges

$$\max_{\Theta} \left[\log\det \boldsymbol{\Theta} - \text{trace}(\mathbf{S}\boldsymbol{\Theta}) - \sum_{(j,k)\notin E} \gamma_{jk}\theta_{jk} \right]$$

Score equations are now: $\boldsymbol{\Theta}^{-1} - \mathbf{S} - \boldsymbol{\Gamma} = \mathbf{0}$

$\boldsymbol{\Gamma}$ is a matrix of Lagrange parameters with nonzero values for all pairs with edges absent.
Can solve by regression as before, except now iteration is needed. Let $\mathbf{W} = \boldsymbol{\Theta}^{-1}$ represent the current estimates, with

$$\begin{pmatrix} \mathbf{W}_{11} & \boldsymbol{w}_{12} \\ \boldsymbol{w}_{12}^{\top} & w_{22} \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{\Theta}_{11} & \boldsymbol{\theta}_{12} \\ \boldsymbol{\theta}_{12}^{\top} & \theta_{22} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0}^{\top} & 1 \end{pmatrix}$$

Graphical Regression Algorithm

Use block coordinate descent. The top-right column of score equation $\mathbf{\Theta}^{-1} - \mathbf{S} - \mathbf{\Gamma} = \mathbf{0}$ says

$$\boldsymbol{w}_{12} - \mathbf{s}_{12} - \boldsymbol{\gamma}_{12} = 0.$$

Manipulating $\boldsymbol{\theta}_{12} = -\mathbf{W}_{11}^{-1}\boldsymbol{w}_{12} \cdot \theta_{22}$ we get

$$\mathbf{W}_{11}\boldsymbol{\beta} - \mathbf{s}_{12} - \boldsymbol{\gamma}_{12} = 0$$

These can be seen as score equations for a constrained version of our previous regression, where we use the current estimate of $\mathbf{W}_{11}$ rather than $\mathbf{S}_{11}$ for the predictor covariance matrix.

Solving the system $\boldsymbol{\Theta}^{-1} - \mathbf{S} - \boldsymbol{\Gamma} = \mathbf{0}$

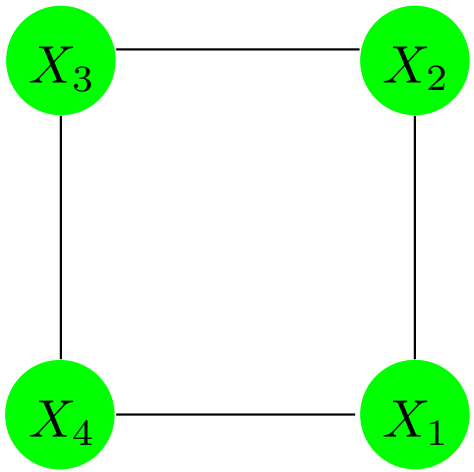
Right column: $\mathbf{W}_{11}\boldsymbol{\beta} - \mathbf{s}_{12} - \boldsymbol{\gamma}_{12} = 0$

- We confine ourselves to the sub-system obtained by omitting the variables constrained to be zero:

$$\mathbf{W}^*_{11}\boldsymbol{\beta}^* - \mathbf{s}^*_{12} = 0$$

- We then fill in $\hat{\boldsymbol{\beta}}$ with $\hat{\boldsymbol{\beta}}^*$ (and zeros), replace $\boldsymbol{w}_{12} \leftarrow \mathbf{W}_{11}\hat{\boldsymbol{\beta}}$, and proceed to the next column.
- As we cycle around the columns, the $\mathbf{W}$ matrix changes, as do the regressions, until the system converges.
- We retain all the $\hat{\boldsymbol{\beta}}$s (for each column) in a matrix $\mathbf{B}$.
- Only at convergence do we need to estimate the $\hat{\theta}_{22}^{-1} = s_{22} - \boldsymbol{w}_{12}^\top\hat{\boldsymbol{\beta}}$ for each column, to recover the entire matrix $\widehat{\boldsymbol{\Theta}}$.

Simple four-variable example with known structure



$$\mathbf{S} = \begin{pmatrix} 10 & 1 & 2 & 4 \\ 1 & 10 & 2 & 1 \\ 2 & 2 & 10 & 3 \\ 4 & 1 & 3 & 10 \end{pmatrix}$$

$$\widehat{\mathbf{\Sigma}} = \begin{pmatrix} 10.00 & 1.00 & 1.31 & 4.00 \\ 1.00 & 10.00 & 2.00 & 0.87 \\ 1.31 & 2.00 & 10.00 & 3.00 \\ 4.00 & 0.87 & 3.00 & 10.00 \end{pmatrix}, \quad \widehat{\mathbf{\Theta}} = \begin{pmatrix} 0.12 & -0.01 & 0.00 & -0.05 \\ -0.01 & 0.11 & -0.02 & 0.00 \\ 0.00 & -0.02 & 0.11 & -0.03 \\ -0.05 & 0.00 & -0.03 & 0.13 \end{pmatrix}$$

Graphical Lasso

Use *lasso* regularized log-likelihood to *learn* sparsity pattern:

$$\max_{\Theta} [\log\det \mathbf{\Theta} - \text{trace}(\mathbf{S\Theta}) - \lambda \cdot \|\mathbf{\Theta}\|_1]$$

with score equations $\mathbf{\Theta}^{-1} - \mathbf{S} - \lambda \cdot \text{Sign}(\mathbf{\Theta}) = \mathbf{0}$.

Graphical Lasso

Use *lasso* regularized log-likelihood to *learn* sparsity pattern:

$$\max_{\Theta} [\log\det \mathbf{\Theta} - \text{trace}(\mathbf{S\Theta}) - \lambda \cdot \|\mathbf{\Theta}\|_1]$$

with score equations $\mathbf{\Theta}^{-1} - \mathbf{S} - \lambda \cdot \text{Sign}(\mathbf{\Theta}) = \mathbf{0}$.

Solving column-wise leads as before to

$$\mathbf{W}_{11}\boldsymbol{\beta} - \mathbf{s}_{12} + \lambda \cdot \text{Sign}(\boldsymbol{\beta}) = 0$$

Compare with solution to lasso problem

$$\min_{\beta} \tfrac{1}{2}\|\mathbf{y} - \mathbf{Z}\boldsymbol{\beta})\|_2^2 + \lambda \cdot \|\beta\|_1$$

with solution

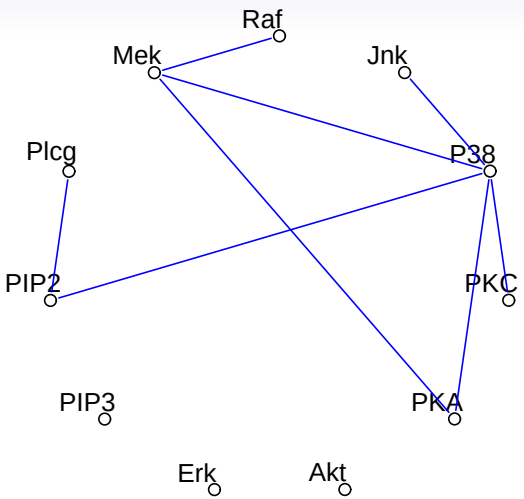
$$\mathbf{Z}^\top\mathbf{Z}\boldsymbol{\beta} - \mathbf{Z}^\top\mathbf{y} + \lambda \cdot \text{Sign}(\boldsymbol{\beta}) = 0$$

This leads to the *graphical lasso* algorithm.

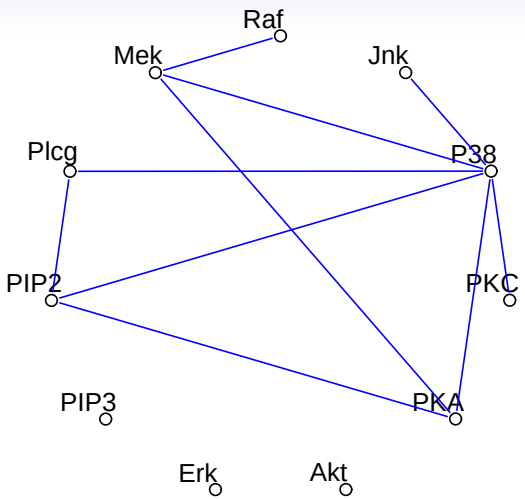
Graphical Lasso Algorithm

1. Initialize $\mathbf{W} = \mathbf{S} + \lambda\mathbf{I}$. The diagonal of $\mathbf{W}$ remains 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 jth row and column, and part 2: the jth row and column.
   (b) Solve the estimating equations $\mathbf{W}_{11}\boldsymbol{\beta} - \mathbf{s}_{12} + \lambda \cdot \text{Sign}(\boldsymbol{\beta}) = 0$ using cyclical coordinate-descent.
   (c) Update $\boldsymbol{w}_{12} = \mathbf{W}_{11}\hat{\boldsymbol{\beta}}$
3. In the final cycle (for each j) solve for $\hat{\boldsymbol{\theta}}_{12} = -\hat{\boldsymbol{\beta}} \cdot \hat{\theta}_{22}$, with $1/\hat{\theta}_{22} = w_{22} - \boldsymbol{w}_{12}^{\top}\hat{\boldsymbol{\beta}}$.

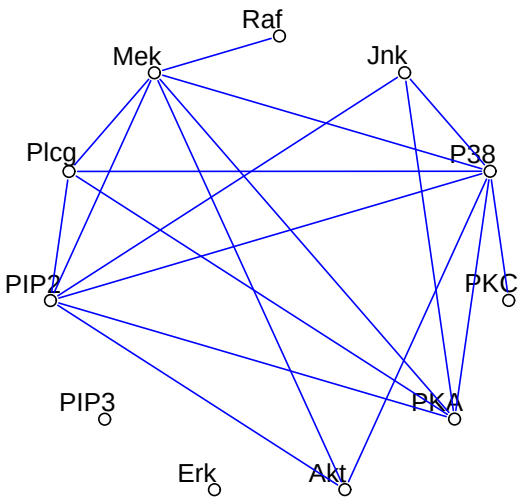
$\lambda = 36$



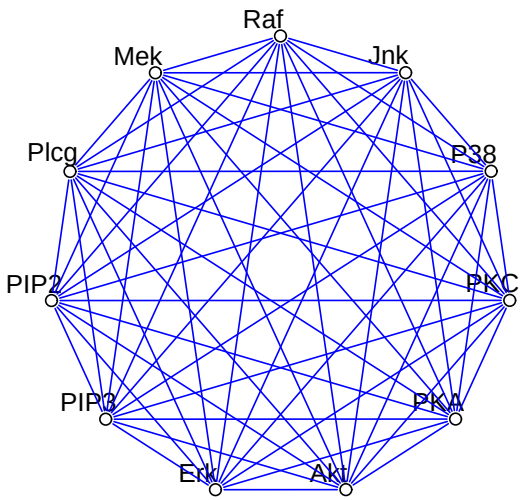
$\lambda = 27$



$\lambda = 7$



$\lambda = 0$



Fit using the `glasso` package in R (on CRAN).

Graphical Lasso: some details

- Blockwise coordinate descent on columns/rows of covariance matrix.
- Each block is also solved by coordinate descent, using the *covariance* form of lasso algorithm.
- Done for a decreasing sequence of values for λ. Use cross-validated log-likelihood to select λ.

Graphical Lasso: some details

- Blockwise coordinate descent on columns/rows of covariance matrix.
- Each block is also solved by coordinate descent, using the *covariance* form of lasso algorithm.
- Done for a decreasing sequence of values for λ. Use cross-validated log-likelihood to select λ.
- Graphical lasso formulation due to Yuan and Lin (2007), Banerjee, El Ghaoui and d'Aspremont (2008). `glasso` algorithm due to Friedman, H, and Tibshirani (2008).

Graphical Lasso: some details

- Blockwise coordinate descent on columns/rows of covariance matrix.
- Each block is also solved by coordinate descent, using the *covariance* form of lasso algorithm.
- Done for a decreasing sequence of values for λ. Use cross-validated log-likelihood to select λ.
- Graphical lasso formulation due to Yuan and Lin (2007), Banerjee, El Ghaoui and d'Aspremont (2008). `glasso` algorithm due to Friedman, H, and Tibshirani (2008).

Warning! this may make your head explode....

Graphical Lasso: some details

- Blockwise coordinate descent on columns/rows of covariance matrix.
- Each block is also solved by coordinate descent, using the *covariance* form of lasso algorithm.
- Done for a decreasing sequence of values for λ. Use cross-validated log-likelihood to select λ.
- Graphical lasso formulation due to Yuan and Lin (2007), Banerjee, El Ghaoui and d'Aspremont (2008). `glasso` algorithm due to Friedman, H, and Tibshirani (2008).

Warning! this may make your head explode....

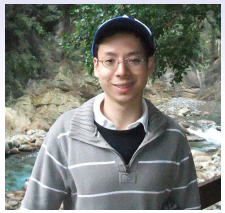
- Mazumder and H (2012) show that `glasso` algorithm solves the *dual* of the lasso objective by block coordinate descent, and each block in turn is solved in dual form (lasso) by coordinate descent.

Qualitative Variables

- With binary variables, second order *Ising* model is popular. Corresponds to first-order interaction models in log-linear models.
- Conditional distributions are logistic regressions.
- Exact maximum-likelihood inference difficult for large p; computations grow exponentially due to computation of partition function.
- Approximations based on lasso-penalized logistic regression in Ravikumar, Wainwright and Lafferty (2008, pub. 2010). Symmetric version in Hoeffling and Tibshirani (2009), using *pseudo-likelihood*

$$\mathrm{PL}(X;\mathbf{\Theta}) = \prod_{j=1}^{p} L(X_j;\mathbf{\Theta}|X_{\setminus j})$$

Mixed variables



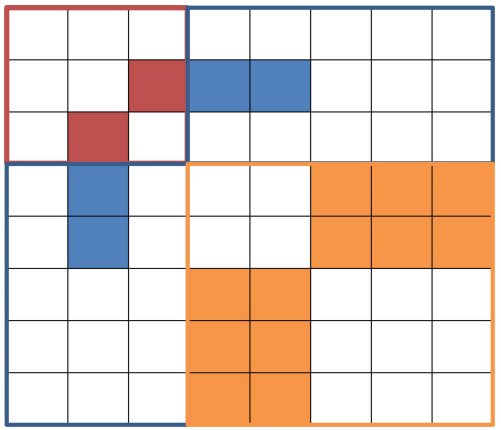
General Markov random field representation, with edge and node potentials.

$$p(x, y; \Theta) \propto \exp\left[\sum_{s=1}^{p}\sum_{t=1}^{p} -\frac{1}{2}\beta_{st}x_s x_t + \sum_{s=1}^{p}\alpha_s x_s + \sum_{s=1}^{p}\sum_{j=1}^{q}\rho_{sj}(y_j)x_s + \sum_{j=1}^{q}\sum_{r=1}^{q}\phi_{rj}(y_r, y_j)\right]$$

- Pseudo likelihood allows simple inference with mixed variables. Conditionals for continuous are Gaussian linear regression models, for categorical are binomial or multinomial logistic regressions.
- Parameters come in symmetric blocks, and the inference should respect this symmetry (next slide)

Group-lasso penalties

Parameters in blocks. Here we have an interaction between a pair of quantitative variables (red), a 2-level qualitative with a quantitative (blue), and an interaction between the 2 level and a 3 level qualitative.



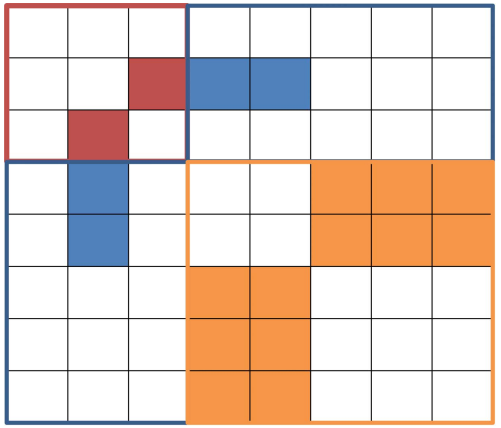
Minimize a negative log-pseudo-likelihood with lasso and group-lasso penalties on parameter blocks.

$$\min_{\Theta} \left[-\ell(\Theta) + \lambda \left(\sum_{s=1}^{p} \sum_{t=1}^{s-1} |\beta_{st}| + \sum_{s=1}^{p} \sum_{j=1}^{q} \|\rho_{sj}\|_2 + \sum_{j=1}^{q} \sum_{r=1}^{j-1} \|\phi_{rj}\|_F \right) \right]$$

Solved using proximal Newton algorithm for a decreasing sequence of values for λ [Lee and H, 2013].

Group-lasso penalties

Parameters in blocks. Here we have an interaction between a pair of quantitative variables (red), a 2-level qualitative with a quantitative (blue), and an interaction between the 2 level and a 3 level qualitative.



Minimize a negative log-pseudo-likelihood with lasso and group-lasso penalties on parameter blocks.

$$\min_{\Theta} \left[-\ell(\Theta) + \lambda \left(\sum_{s=1}^{p} \sum_{t=1}^{s-1} |\beta_{st}| + \sum_{s=1}^{p} \sum_{j=1}^{q} \|\rho_{sj}\|_2 + \sum_{j=1}^{q} \sum_{r=1}^{j-1} \|\phi_{rj}\|_F \right) \right]$$

Solved using proximal Newton algorithm for a decreasing sequence of values for λ [Lee and H, 2013].

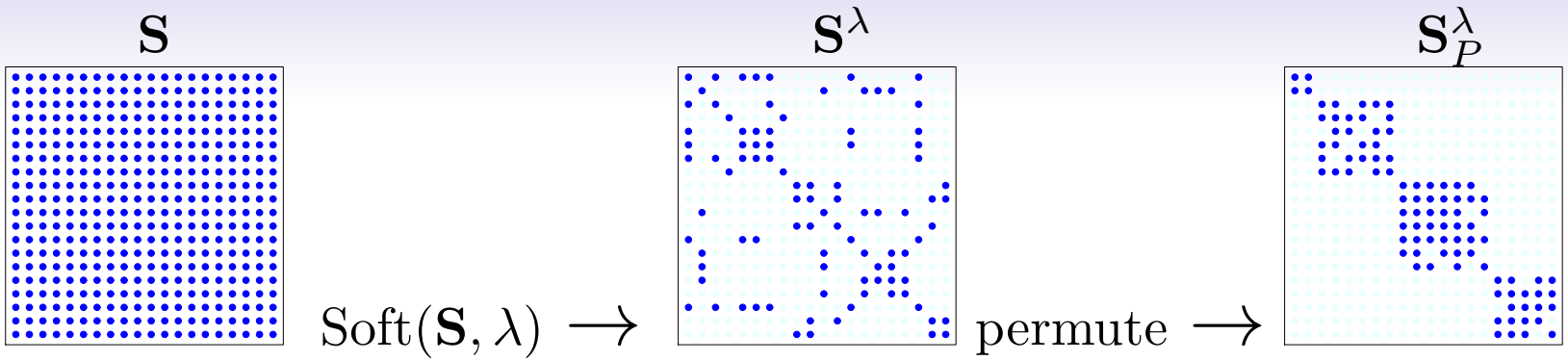
See also Guo, Levina, Michailidis and Zhu (2011, categorical) and Yang, Ravikumar, Allen and Liu (2013, exponential families).

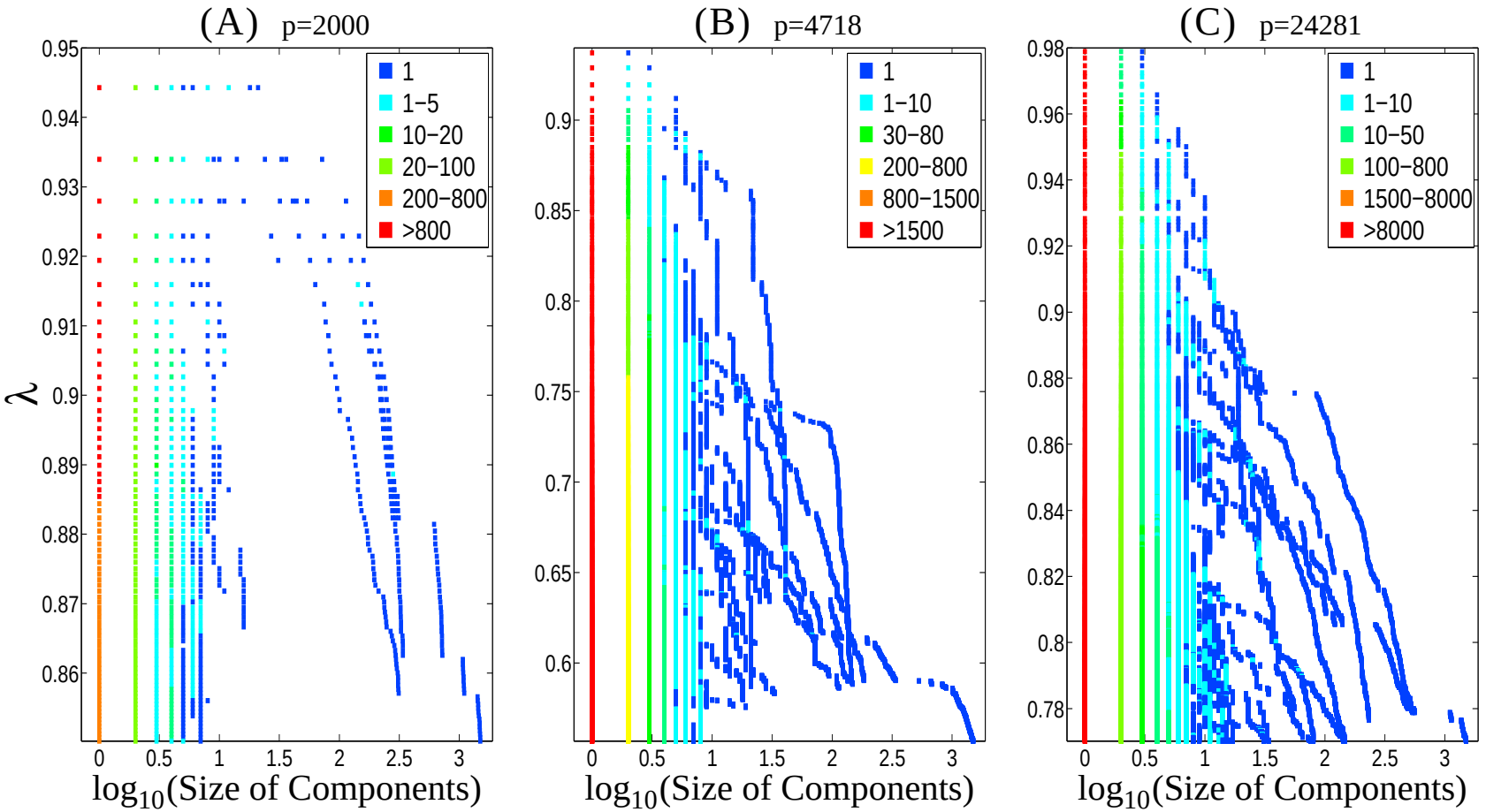
Large scale graphical lasso



- The cost of `glasso` is $O(np^2 + p^{3+\Delta})$ where $\Delta \in [0, 1]$; prohibitive for genomic scale p.
- For many of these problems, $n \ll p$, so we can only fit very sparse solutions anyway.
- Simple idea:
  - Compute $\mathbf{S}$ and soft-threshold: $\mathrm{S}_{ij}^{\lambda} = \mathrm{sign}(\mathrm{S}_{ij})(|\mathrm{S}_{ij}| - \lambda)_+$.
  - Reorder rows and columns to achieve block-diagonal pattern [Tarjan, 1972]
  - Run `glasso` on each corresponding block of $\mathbf{S}$ with parameter λ, and then reconstruct.
  - Solution solves original glasso problem!

Witten, Friedman and Simon, 2011; Mazumder and H, 2012.

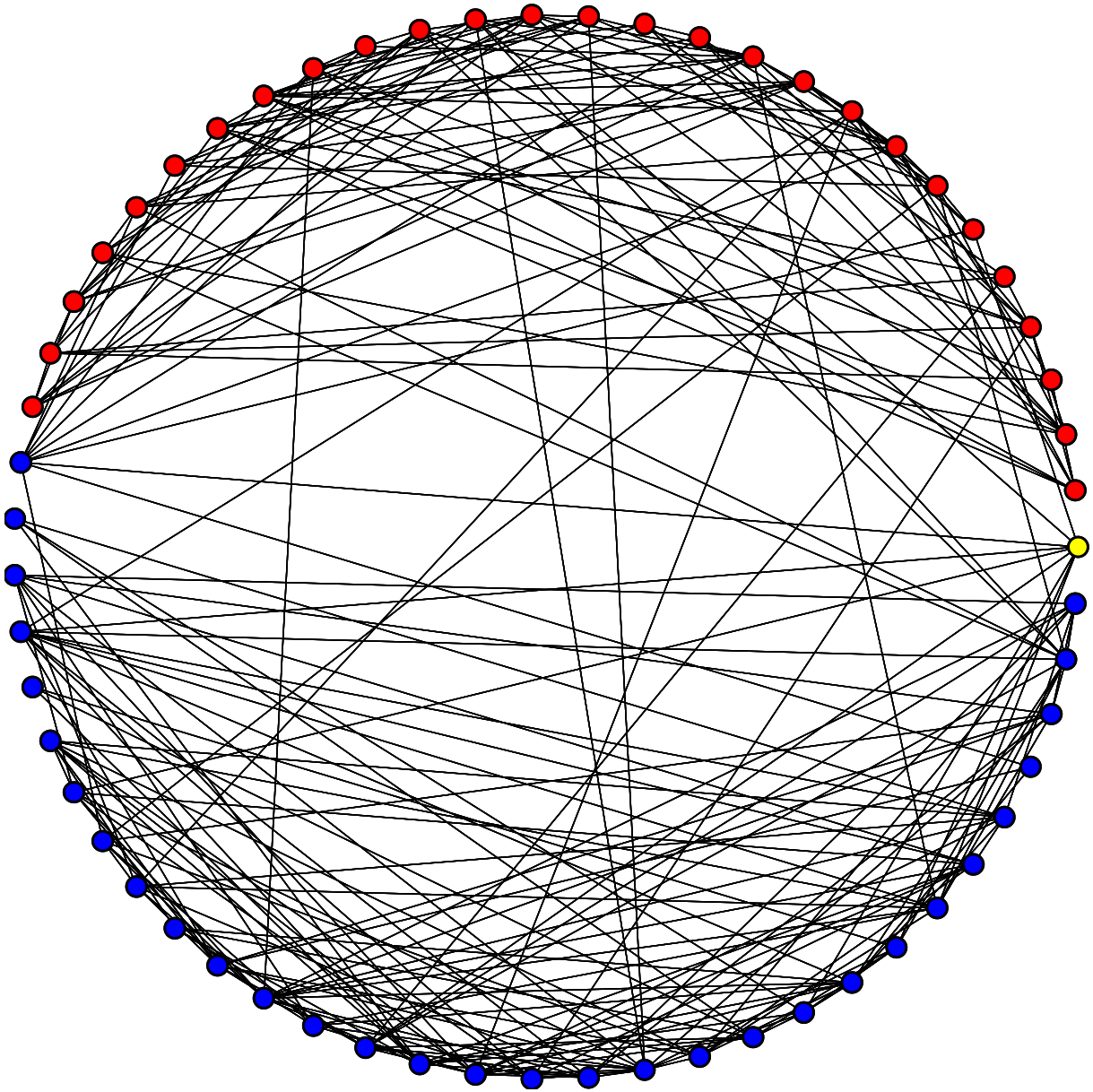

$\mathbf{S}$
$\mathbf{S}^\lambda$
$\mathbf{S}^\lambda_P$
$\text{Soft}(\mathbf{S}, \lambda) \rightarrow$
permute $\rightarrow$


(A) p=2000
(B) p=4718
(C) p=24281
1
1−5
10−20
20−100
200−800
>800
1−10
30−80
200−800
800−1500
>1500
10−50
100−800
1500−8000
>8000
λ
$\log_{10}$(Size of Components)

What are graphical models good for?

- Exploratory association analysis. Makes terrific graphs.
- Hypothesis generation in genomics, multiomics
- Applications in many multivariate applications, like discriminant analysis, mixture models, regression, classification, copulas, ...

US Senate 2004–06 voting alignment



Anomaly detection and missing-value imputation

Many organizations process a vector of measurements on a regular basis. Examples include:

- Chemical plants, where the vector is populated every minute with readouts from system processes and monitors.
- Large multi-center companies, with categorized sales figures, advertising budgets, etc which change daily or weekly.
- Financial firms where the daily vector might consist of trade volumes, security returns, bond prices, etc.

Anomaly detection and missing-value imputation

Many organizations process a vector of measurements on a regular basis. Examples include:

- Chemical plants, where the vector is populated every minute with readouts from system processes and monitors.
- Large multi-center companies, with categorized sales figures, advertising budgets, etc which change daily or weekly.
- Financial firms where the daily vector might consist of trade volumes, security returns, bond prices, etc.

In terms of quality control, at least two questions are of interest:

- Are there any entries in the new vector that are surprising?
- If entries are missing or too surprising, can we make an educated guess as to what their values should be?

Mahalanobis filter

We have *clean* training vectors $x_1, x_2, \ldots, x_N$, with each $x_i \in \mathbb{R}^p$. Assume each of the p elements are standardized to have mean zero and unit variance across the N samples.

Mahalanobis filter

We have *clean* training vectors $x_1, x_2, \ldots, x_N$, with each $x_i \in \mathbb{R}^p$. Assume each of the p elements are standardized to have mean zero and unit variance across the N samples. Let $\mathbf{\Theta}$ be a (sparse) precision matrix estimate, and $\mathbf{\Sigma} = \mathbf{\Theta}^{-1}$.

Mahalanobis filter

We have *clean* training vectors $x_1, x_2, \ldots, x_N$, with each $x_i \in \mathbb{R}^p$. Assume each of the p elements are standardized to have mean zero and unit variance across the N samples. Let $\mathbf{\Theta}$ be a (sparse) precision matrix estimate, and $\mathbf{\Sigma} = \mathbf{\Theta}^{-1}$. Define for an observed vector x (e.g. the new vector)

$$\begin{array}{rcccl} \hat{x}_{\mathcal{A}} & = & E(X_{\mathcal{A}}|x_{\mathcal{R}}) & = & \mathbf{\Sigma}_{\mathcal{A}\mathcal{R}}\mathbf{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1}x_{\mathcal{R}} \\ \mathbf{\Sigma}_{\mathcal{A}|\mathcal{R}} & = & \mathrm{Cov}(X_{\mathcal{A}}|x_{\mathcal{R}}) & = & \mathbf{\Sigma}_{\mathcal{A}\mathcal{A}} - \mathbf{\Sigma}_{\mathcal{A}\mathcal{R}}\mathbf{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1}\mathbf{\Sigma}_{\mathcal{R}\mathcal{A}} \end{array}$$

for a subset $\mathcal{A}$ of potential anomalies, and $\mathcal{R}$ the rest, with $\mathcal{A} \cup \mathcal{R} = \{1, 2, \ldots, p\}$.

Mahalanobis filter

We have *clean* training vectors $x_1, x_2, \ldots, x_N$, with each $x_i \in \mathbb{R}^p$. Assume each of the p elements are standardized to have mean zero and unit variance across the N samples. Let $\mathbf{\Theta}$ be a (sparse) precision matrix estimate, and $\mathbf{\Sigma} = \mathbf{\Theta}^{-1}$. Define for an observed vector x (e.g. the new vector)

$$\begin{array}{rcrcl} \hat{x}_{\mathcal{A}} & = & E(X_{\mathcal{A}}|x_{\mathcal{R}}) & = & \mathbf{\Sigma}_{\mathcal{A}\mathcal{R}}\mathbf{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1}x_{\mathcal{R}} \\ \mathbf{\Sigma}_{\mathcal{A}|\mathcal{R}} & = & \mathrm{Cov}(X_{\mathcal{A}}|x_{\mathcal{R}}) & = & \mathbf{\Sigma}_{\mathcal{A}\mathcal{A}} - \mathbf{\Sigma}_{\mathcal{A}\mathcal{R}}\mathbf{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1}\mathbf{\Sigma}_{\mathcal{R}\mathcal{A}} \end{array}$$

for a subset $\mathcal{A}$ of potential anomalies, and $\mathcal{R}$ the rest, with $\mathcal{A} \cup \mathcal{R} = \{1, 2, \ldots, p\}$.

$$x^\top \mathbf{\Sigma}^{-1} x = x_{\mathcal{R}} \mathbf{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1} x_{\mathcal{R}} + (x_{\mathcal{A}} - \hat{x}_{\mathcal{A}})^\top \mathbf{\Sigma}_{\mathcal{A}|\mathcal{R}}^{-1} (x_{\mathcal{A}} - \hat{x}_{\mathcal{A}})$$

Mahalanobis filter

We have *clean* training vectors $x_1, x_2, \ldots, x_N$, with each $x_i \in \mathbb{R}^p$. Assume each of the p elements are standardized to have mean zero and unit variance across the N samples. Let $\mathbf{\Theta}$ be a (sparse) precision matrix estimate, and $\mathbf{\Sigma} = \mathbf{\Theta}^{-1}$. Define for an observed vector x (e.g. the new vector)

$$\begin{array}{rcrcl} \hat{x}_{\mathcal{A}} & = & E(X_{\mathcal{A}}|x_{\mathcal{R}}) & = & \mathbf{\Sigma}_{\mathcal{A}\mathcal{R}}\mathbf{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1}x_{\mathcal{R}} \\ \mathbf{\Sigma}_{\mathcal{A}|\mathcal{R}} & = & \mathrm{Cov}(X_{\mathcal{A}}|x_{\mathcal{R}}) & = & \mathbf{\Sigma}_{\mathcal{A}\mathcal{A}} - \mathbf{\Sigma}_{\mathcal{A}\mathcal{R}}\mathbf{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1}\mathbf{\Sigma}_{\mathcal{R}\mathcal{A}} \end{array}$$

for a subset $\mathcal{A}$ of potential anomalies, and $\mathcal{R}$ the rest, with $\mathcal{A} \cup \mathcal{R} = \{1, 2, \ldots, p\}$.

$$x^\top \mathbf{\Sigma}^{-1} x = x_{\mathcal{R}} \mathbf{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1} x_{\mathcal{R}} + (x_{\mathcal{A}} - \hat{x}_{\mathcal{A}})^\top \mathbf{\Sigma}_{\mathcal{A}|\mathcal{R}}^{-1} (x_{\mathcal{A}} - \hat{x}_{\mathcal{A}})$$

Look for subset $\mathcal{A}$ such that Mahalanobis term on the right is unduly large!

Optimization view of Mahalanobis filter

- Find set $\mathcal{A}$ with $|\mathcal{A}| = k$ and largest value for green term

$$x^\top \boldsymbol{\Sigma}^{-1} x = x_{\mathcal{R}} \boldsymbol{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1} x_{\mathcal{R}} + (x_{\mathcal{A}} - \hat{x}_{\mathcal{A}})^\top \boldsymbol{\Sigma}_{\mathcal{A}|\mathcal{R}}^{-1} (x_{\mathcal{A}} - \hat{x}_{\mathcal{A}})$$

- Equivalently solve

$$\min_{\nu} (x - \nu)^\top \boldsymbol{\Sigma}^{-1} (x - \nu) \quad \text{s.t. } \|\nu\|_0 \leq k$$

  Solution identifies $\mathcal{R}$ with minimal $x_{\mathcal{R}} \boldsymbol{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1} x_{\mathcal{R}}$

Optimization view of Mahalanobis filter

- Find set $\mathcal{A}$ with $|\mathcal{A}| = k$ and largest value for green term

$$x^\top \mathbf{\Sigma}^{-1} x = x_{\mathcal{R}} \mathbf{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1} x_{\mathcal{R}} + (x_{\mathcal{A}} - \hat{x}_{\mathcal{A}})^\top \mathbf{\Sigma}_{\mathcal{A}|\mathcal{R}}^{-1} (x_{\mathcal{A}} - \hat{x}_{\mathcal{A}})$$

- Equivalently solve

$$\min_{\nu} (x - \nu)^\top \mathbf{\Sigma}^{-1} (x - \nu) \quad \text{s.t. } \|\nu\|_0 \leq k$$

  Solution identifies $\mathcal{R}$ with minimal $x_{\mathcal{R}} \mathbf{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1} x_{\mathcal{R}}$ — *hard!*

- Can convexify

$$\min_{\nu} (x - \nu)^\top \mathbf{\Sigma}^{-1} (x - \nu) \quad \text{s.t. } \|\nu\|_1 \leq k$$

  Nice lasso algorithm emerges

- Or use stepwise (next slide)

Stepwise anomaly detection

Standard formula for partitioned inverse $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$:

$$\begin{bmatrix} \boldsymbol{\Theta}_{\mathcal{R}\mathcal{R}} & \boldsymbol{\Theta}_{\mathcal{R}\mathcal{A}} \\ \boldsymbol{\Theta}_{\mathcal{A}\mathcal{R}} & \boldsymbol{\Theta}_{\mathcal{A}\mathcal{A}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1} + \boldsymbol{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1}\boldsymbol{\Sigma}_{\mathcal{R}\mathcal{A}}\boldsymbol{\Theta}_{\mathcal{A}\mathcal{A}}\boldsymbol{\Sigma}_{\mathcal{A}\mathcal{R}}\boldsymbol{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1} & -\boldsymbol{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1}\boldsymbol{\Sigma}_{\mathcal{R}\mathcal{A}}\boldsymbol{\Theta}_{\mathcal{A}\mathcal{A}} \\ -\boldsymbol{\Theta}_{\mathcal{A}\mathcal{A}}\boldsymbol{\Sigma}_{\mathcal{A}\mathcal{R}}\boldsymbol{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1} & (\boldsymbol{\Sigma}_{\mathcal{A}\mathcal{A}} - \boldsymbol{\Sigma}_{\mathcal{A}\mathcal{R}}\boldsymbol{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1}\boldsymbol{\Sigma}_{\mathcal{R}\mathcal{A}})^{-1} \end{bmatrix}$$

Let $z = \boldsymbol{\Theta}x$. Then

$$\begin{aligned} z_{\mathcal{A}} &= \boldsymbol{\Theta}_{\mathcal{A}\mathcal{A}}(x_{\mathcal{A}} - \hat{x}_{\mathcal{A}}), \\ z_{\mathcal{A}}^{\top}\boldsymbol{\Theta}_{\mathcal{A}\mathcal{A}}^{-1}z_A &= (x_{\mathcal{A}} - \hat{x}_{\mathcal{A}})^{\top}\boldsymbol{\Sigma}_{\mathcal{A}|\mathcal{R}}^{-1}(x_{\mathcal{A}} - \hat{x}_{\mathcal{A}}). \end{aligned}$$

If $\mathcal{A} = j$ is a singleton, then

$$(x_{\mathcal{A}} - \hat{x}_{\mathcal{A}})^{\top}\boldsymbol{\Sigma}_{\mathcal{A}|\mathcal{R}}^{-1}(x_{\mathcal{A}} - \hat{x}_{\mathcal{A}}) = z_j^2/\theta_{jj}$$

This is computed for all j, and the largest is inspected as potential anomaly.
If rejected, reduce $\boldsymbol{\Sigma}$ to $\boldsymbol{\Sigma}_{\mathcal{R}\mathcal{R}}$, invert and repeat.

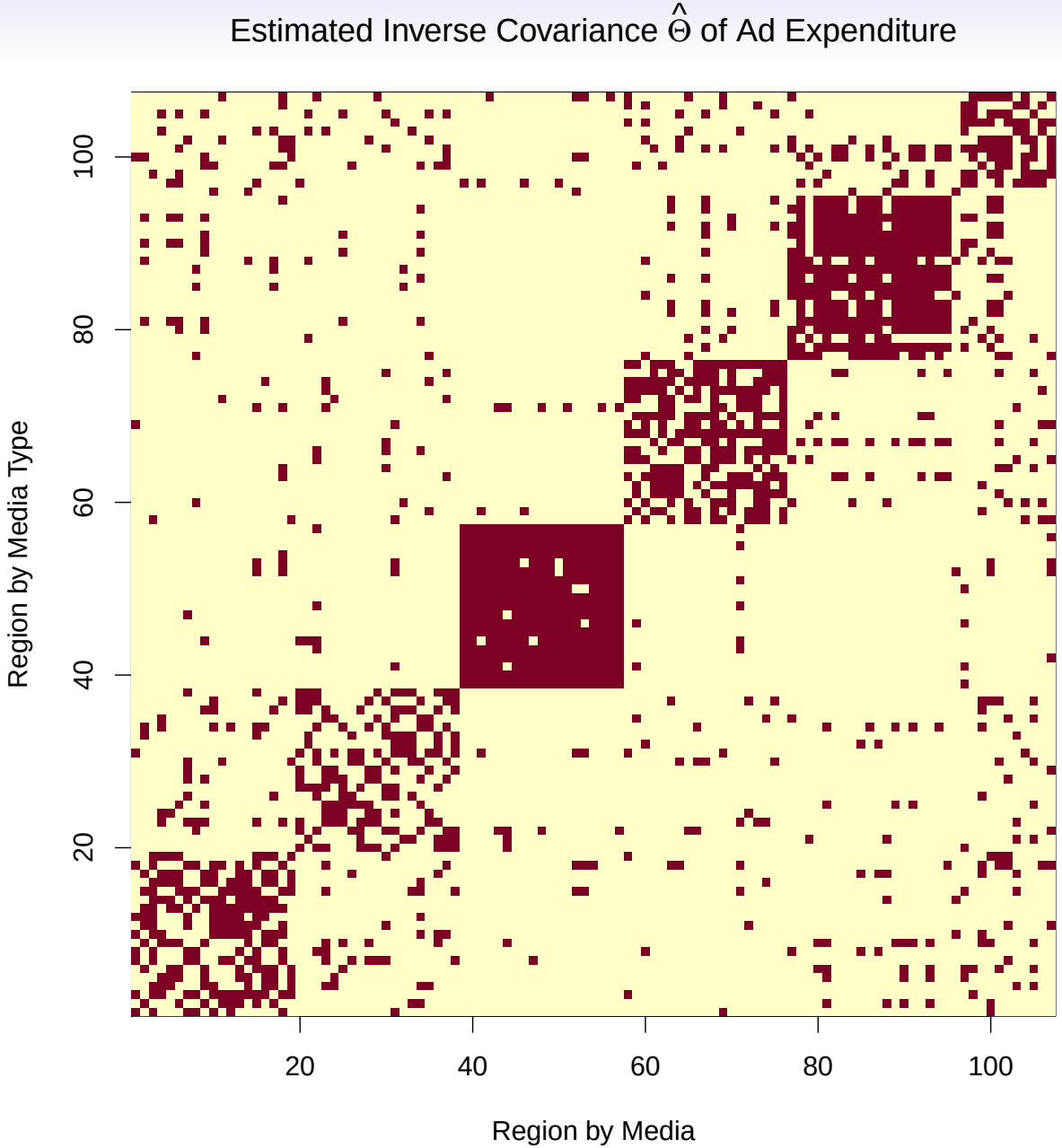
Example: Insurance Expenditure

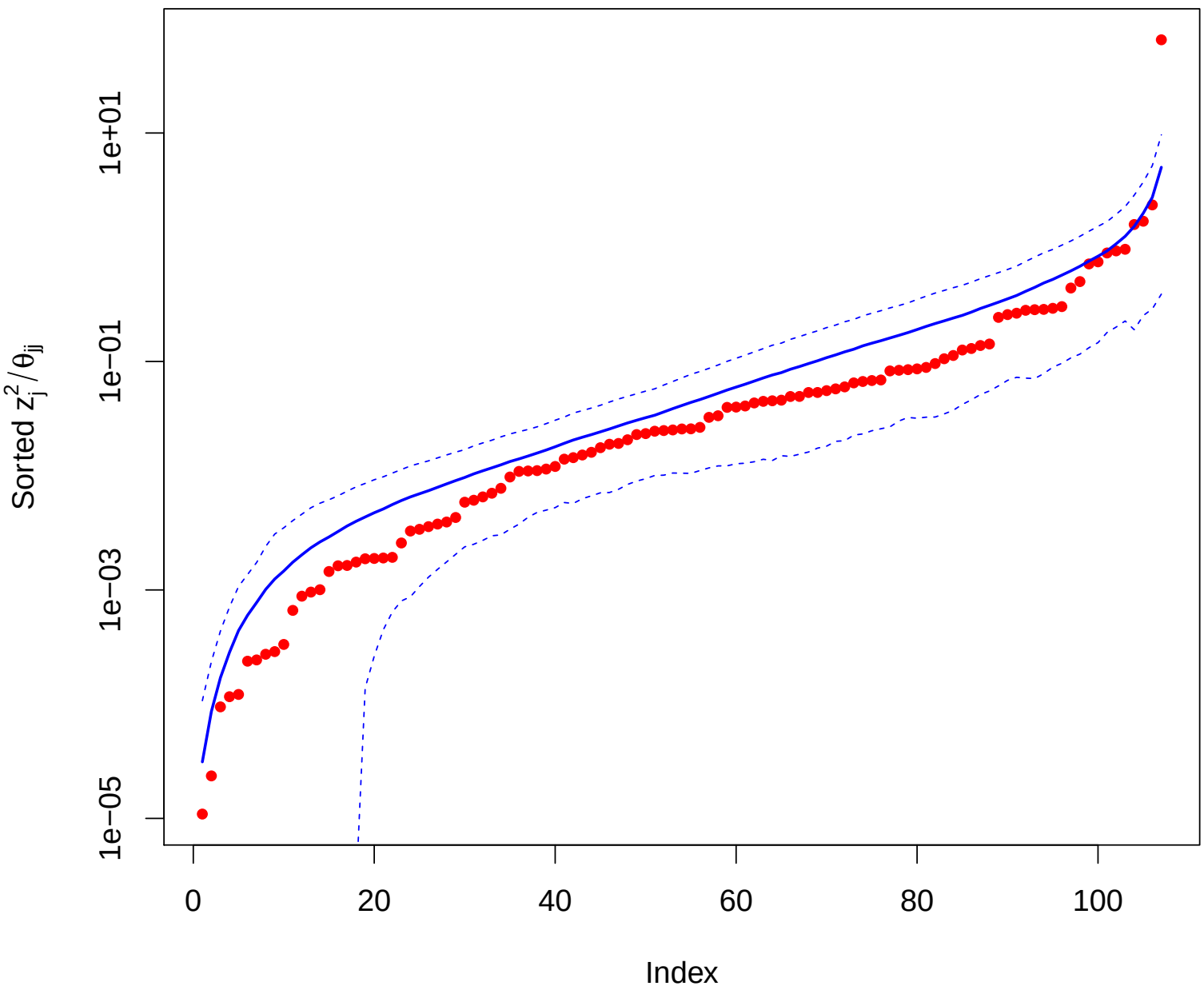
An online auto-insurance company has weekly spend in 8 media types (e.g. *TV, Web, Radio, etc*) and 19 regions (eg *San Francisco-Oakland-San Jose, CA*)

After filtering, left with 107 Region $\times$ Type variables.

216 training weeks to build graphical lasso model (next slide).

λ chosen by CV


Estimated Inverse Covariance $\hat{\Theta}$ of Ad Expenditure
Region by Media Type
20
40
60
80
100
Region by Media
20
40
60
80
100


Mahalanobis Filter
Sorted z_j^2/θ_{jj}
1e−05
1e−03
1e−01
1e+01
0
20
40
60
80
100
Index

Details

- After each step, new $\boldsymbol{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1}$ can be computed by a rank-one update of $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$:

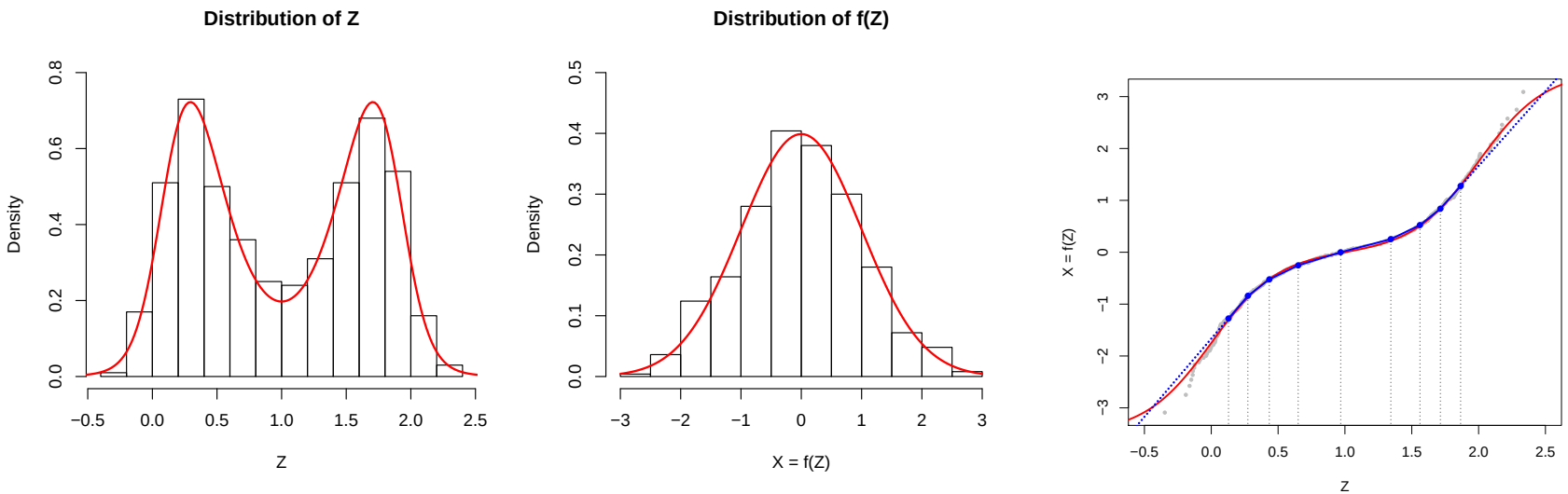
$$\boldsymbol{\Sigma}_{\mathcal{R}\mathcal{R}}^{-1} = \boldsymbol{\Theta}_{\mathcal{R}\mathcal{R}} - \frac{\boldsymbol{\theta}_{\mathcal{R}\mathcal{A}}\boldsymbol{\theta}_{\mathcal{R}\mathcal{A}}^{\top}}{\theta_{\mathcal{A}\mathcal{A}}}$$

Details

- After each step, new $\mathbf{\Sigma}^{-1}_{\mathcal{R}\mathcal{R}}$ can be computed by a rank-one update of $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$:

$$\mathbf{\Sigma}^{-1}_{\mathcal{R}\mathcal{R}} = \mathbf{\Theta}_{\mathcal{R}\mathcal{R}} - \frac{\boldsymbol{\theta}_{\mathcal{R}\mathcal{A}}\boldsymbol{\theta}^{\top}_{\mathcal{R}\mathcal{A}}}{\theta_{\mathcal{A}\mathcal{A}}}$$

- Gaussian? We transform each variable marginally to be approximately Gaussian, hence work with *Gaussian Copula* model (Liu, Han, Yuan, Lafferty, Wasserman AoS 2012)



Details

- In practice we use a *sparse + low-rank* model for $\mathbf{\Theta}$, which allows for latent factors. The neighborhood regressors for a particular variable now include the other variables as well as the $r \ll p$ (estimated) factor scores.
  Extension of factor analysis model for random vector X:

$$X \sim \mathbf{F}s + u, \quad s \sim N(0, \mathbf{I}_r),\ u \sim N(0, \mathbf{D})$$

Details

- In practice we use a *sparse + low-rank* model for $\mathbf{\Theta}$, which allows for latent factors. The neighborhood regressors for a particular variable now include the other variables as well as the $r \ll p$ (estimated) factor scores.
  Extension of factor analysis model for random vector X:

$$X \sim \mathbf{F}s + u, \quad s \sim N(0, \mathbf{I}_r),\ u \sim N(0, \mathbf{D})$$

  - $\mathbf{D}$ diagonal is traditional factor model, with $\mathbf{\Sigma} = \mathbf{F}\mathbf{F}^\top + \mathbf{D}$, and similar form for $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$.

Details

- In practice we use a *sparse + low-rank* model for $\mathbf{\Theta}$, which allows for latent factors. The neighborhood regressors for a particular variable now include the other variables as well as the $r \ll p$ (estimated) factor scores.
  Extension of factor analysis model for random vector X:

$$X \sim \mathbf{F}s + u, \quad s \sim N(0, \mathbf{I}_r),\ u \sim N(0, \mathbf{D})$$

  - $\mathbf{D}$ diagonal is traditional factor model, with $\mathbf{\Sigma} = \mathbf{F}\mathbf{F}^\top + \mathbf{D}$, and similar form for $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$.
  - $\mathbf{D}$ block diagonal with small blocks leads to block-diagonal + low-rank $\mathbf{\Theta}$.

Details

- In practice we use a *sparse + low-rank* model for $\mathbf{\Theta}$, which allows for latent factors. The neighborhood regressors for a particular variable now include the other variables as well as the $r \ll p$ (estimated) factor scores.
  Extension of factor analysis model for random vector X:

$$X \sim \mathbf{F}s + u, \quad s \sim N(0, \mathbf{I}_r),\ u \sim N(0, \mathbf{D})$$

  - $\mathbf{D}$ diagonal is traditional factor model, with $\mathbf{\Sigma} = \mathbf{F}\mathbf{F}^\top + \mathbf{D}$, and similar form for $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$.
  - $\mathbf{D}$ block diagonal with small blocks leads to block-diagonal + low-rank $\mathbf{\Theta}$.
  - $\mathbf{D}$ with sparse inverse leads to sparse + low-rank $\mathbf{\Theta}$.

Details

- In practice we use a *sparse + low-rank* model for $\mathbf{\Theta}$, which allows for latent factors. The neighborhood regressors for a particular variable now include the other variables as well as the $r \ll p$ (estimated) factor scores.
  Extension of factor analysis model for random vector X:

$$X \sim \mathbf{F}s + u, \quad s \sim N(0, \mathbf{I}_r),\ u \sim N(0, \mathbf{D})$$

  - $\mathbf{D}$ diagonal is traditional factor model, with $\mathbf{\Sigma} = \mathbf{F}\mathbf{F}^\top + \mathbf{D}$, and similar form for $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$.
  - $\mathbf{D}$ block diagonal with small blocks leads to block-diagonal + low-rank $\mathbf{\Theta}$.
  - $\mathbf{D}$ with sparse inverse leads to sparse + low-rank $\mathbf{\Theta}$.

  Work in progress!

Summary

- MVN never ceases to amaze!
- Precision matrix $\boldsymbol{\Theta}$ has all regressions encoded.
- Estimating $\boldsymbol{\Theta}$ with sparsity allows us to learn dependence structure in underlying variables.
- Also regularizes both $\widehat{\boldsymbol{\Theta}}$ and $\widehat{\boldsymbol{\Sigma}}$, and allows for estimation when $p > N$.
- Having fun revisiting graphical models with anomaly detection.

Thank you for attending!