## Case Study 3: fMRI Prediction

# Graphical LASSO


Machine Learning/Statistics for Big Data
CSE599C1/STAT592, University of Washington
Emily Fox


February 26th, 2013 + 28th





# Multivariate Normal Models

- So far, we looked at the univariate multiple regression $y^i \in \mathbb{R}$

$$y^i = \beta_0 + \beta_1 x_1^i + \ldots + \beta_p x_p^i + \epsilon^i \qquad \epsilon^i \sim N(0,\sigma^2)$$
$$= \beta^T x^i + \epsilon^i$$
$$\Rightarrow y^i \sim N(\beta^T x^i, \sigma^2)$$

- If one has a multivariate response $y^i \in \mathbb{R}^d$ ← # of semantic features
  - Assuming independence between dimensions

$$y^i \sim N\left( \overbrace{\begin{bmatrix} - \beta^{(1)T} - \\ - \beta^{(2)T} - \\ \vdots \\ - \beta^{(d)T} - \end{bmatrix}}^{B^T} x^i , \begin{bmatrix} \sigma^2 & & 0 \\ & \ddots & \\ 0 & & \sigma^2 \end{bmatrix} \right)$$

$\beta^{(\ell)}$ are reg. coeff. for the $\ell^{th}$ dim

# Multivariate Normal Models

- If one has a multivariate response $y^i \in \mathbb{R}^d$ "dog" and "furry"
  - Assuming correlation between the output dimensions

$$y^i \sim N(\beta^T x^i, \Sigma)$$

recall : $\mathrm{cov}(y_s, y_t) = \Sigma_{st}$

- Assume linear (or other mean regression) is removed and focus on the correlation structure

$$y^i \sim N(0, \Sigma)$$

sym., pos. def.

- Matrix valued parameter!

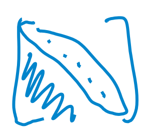
See more of this in Case Study 4




# High-Dimensional Covariance

- What if *d* is large? many semantic features

# params $(\Sigma) = \frac{d(d+1)}{2}$



sym.

Again, consider $d \gg N$, but $O(d^2)$ params to est.

- A few common approaches:
  - Low-rank approximations ✓ last lecture
  - Sparsity assumptions

# Low-Rank Approximations

- In general, assume some matrix parameter

$$\underset{d\times m}{\Theta} = \underset{d\times k}{A}\,\underset{m\times k}{B'} \qquad k << d,m$$

will see this in case study 4

- Here, $\Sigma$ must be a symmetric, positive definite matrix

square

$$\underset{d\times d}{\Sigma} = \underset{d\times k}{\Lambda}\Lambda^T + \Sigma_0 \qquad \Sigma_0 = \begin{bmatrix} \sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_d^2 \end{bmatrix}$$

sym. + square

pos. def.

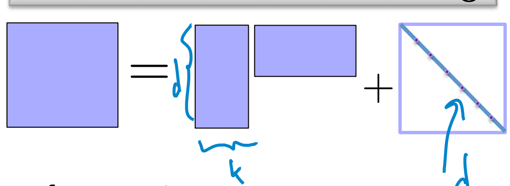



# Low-Rank Approximations

- In pictures…

$$\Sigma_0 = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_d^2)$$

$$\underset{d\times d}{\Sigma} = \underset{d\times k}{\Lambda}\Lambda' + \Sigma_0 \qquad k << d$$




- Number of parameters:

$$d\cdot k + d = d(k+1)$$

sig. reduction in param. for $k << d$

# Latent Factor Models

- Low-rank approximation arises from a latent factor model

$y^i = \Lambda \eta^i + \epsilon^i$ $\qquad$ $\eta^i \overset{iid}{\sim} N_k(0, I)$

"obs" $\qquad$ "factor loadings" $\qquad$ "latent factors" $\qquad$ $\epsilon^i \overset{iid}{\sim} N_d(0, \Sigma_0)$ ← diag

- Proof:

$$Cov(y^i; \Lambda, \Sigma_0) = E[(y^i - \cancel{E[y]}^{\,0})(y - \cancel{E[y]}^{\,0})^T] = E[yy^T]$$

$$= E[(\Lambda\eta + \epsilon)(\Lambda\eta + \epsilon)^T] = \Lambda E[\eta\eta^T]\Lambda^T + 2E[\eta^T]\Lambda^T \cancel{E[\epsilon]}^{\,0} + E[\epsilon\epsilon^T]$$

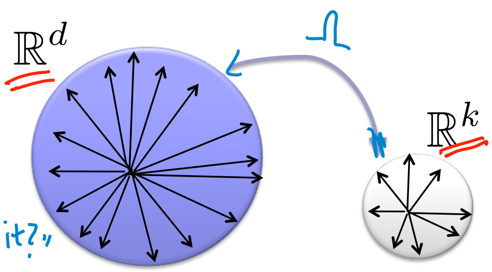
$$= \Lambda I \Lambda^T + \Sigma_0 \quad \blacksquare$$

# Lower-dim Embeddings

Very cool! Very efficient

**Sharing information in**
***low-dim subspace***

obs. $y^i$




$\eta^i$ latent factor

"can you hold it?" and "is it bigger than a bread box?" } redundant ← latent factor "is it big?"

## Sparsity Assumptions

- What if we assume $\Sigma$ is sparse?

$(i \neq j)$ $\Sigma_{ij} = 0 \Rightarrow y_i \perp\!\!\!\perp y_j$

$\mathrm{Cov}(y_i, y_j) = 0$

Could assume $\Sigma$ sparse to reduce # params, but each 0 encodes an indep. assumption ... often too strong

- More often, we can reasonably make statements about *conditional independence*

"cat" $\perp\!\!\!\perp$ "dog" | "animal", "furry", "pet"...




## Information Form

- Motivations for considering "information form" of multivariate normal
  - Easier to read off conditional densities
  - Has log-linear form in terms of "information parameters"

$y \sim N(\mu, \Sigma)$

$$\frac{1}{\sqrt{2\pi|\Sigma|}} e^{-\frac{1}{2}\underline{(y-\mu)^T \Sigma^{-1}(y-\mu)}}$$

$y^T\Sigma^{-1}y$ ✓
$-2y^T\Sigma^{-1}\mu$ ✗
$+\underbrace{\mu^T\Sigma^{-1}\mu}_{\text{const. wrt } y}$

$$\Updownarrow \quad \begin{array}{l} \Omega = \Sigma^{-1} \\ \eta = \Sigma^{-1}\mu \end{array}$$

$$\propto e^{\eta^T y - \frac{1}{2} y^T \Omega y}$$

$y \sim N^{-1}(\eta, \Omega)$

# Conditional Densities

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_d \end{bmatrix} \quad \leftarrow A, \; \bar{A}$$

- Assume a model with

$$y \sim N^{-1}(\eta, \Omega)$$

and divide the dimensions into two sets $A, \bar{A}$

- Then,

$$\begin{bmatrix} y_A \\ y_{\bar{A}} \end{bmatrix} \sim N^{-1}\left( \begin{bmatrix} \eta_A \\ \eta_{\bar{A}} \end{bmatrix}, \begin{bmatrix} \Omega_{AA} & \Omega_{A\bar{A}} \\ \Omega_{\bar{A}A} & \Omega_{\bar{A}\bar{A}} \end{bmatrix} \right)$$

submatrix of $\Omega$ with row indices in $A$ and col. indices in $\bar{A}$

$$p(y_A \mid y_{\bar{A}}) = N^{-1}(\eta_A - \Omega_{A\bar{A}} y_{\bar{A}}, \underline{\underline{\Omega_{AA}}})$$



# Conditional Densities

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_s \\ \vdots \\ y_t \\ \vdots \\ y_d \end{bmatrix} \quad A = \{s,t\}$$

- Let $A = \{s, t\}$ $\quad \bar{A} =$ everything else

$y_s, y_t \qquad y_{\setminus st}$

$$p(y_A \mid y_{\bar{A}}) = \mathcal{N}^{-1}(\eta_A - \Omega_{A\bar{A}} y_{\bar{A}}, \Omega_{AA})$$

$$\begin{bmatrix} \Omega_{ss} & \Omega_{st} \\ \Omega_{ts} & \Omega_{tt} \end{bmatrix}$$

what if $\Omega_{st} = 0$ ? $\Rightarrow \begin{bmatrix} \Omega_{ss} & 0 \\ 0 & \Omega_{tt} \end{bmatrix}$

$$\text{cov}(y_s, y_t \mid y_{\setminus st}) = \Omega_{AA}^{-1} = \begin{bmatrix} \Omega_{ss}^{-1} & 0 \\ 0 & \Omega_{tt}^{-1} \end{bmatrix}$$

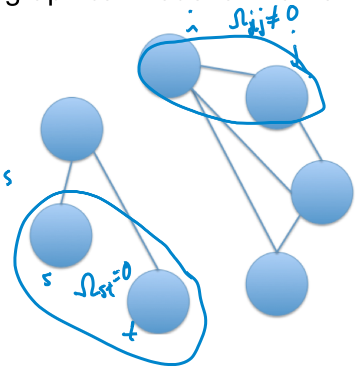
$$\Leftrightarrow \boxed{y_s \perp\!\!\!\perp y_t \mid y_{\setminus st} \iff \Omega_{st} = 0}$$

- Therefore,

# Connection with Graphical Models

- Undirected graphical model or Markov random field (MRF)

No edge (s,t) $\Rightarrow$ $y_s \perp\!\!\!\perp y_t \mid y_{\setminus s,t}$




In Gaussian graphical model case,

$\Omega_{st} = 0$ defines the edge set

In particular

$E = \{(s,t) : \Omega_{st} \neq 0\}$

$$p(y \mid \eta, \Omega) \propto \prod_t \psi_t(y_t) \prod_{(s,t)\in E} \psi_{st}(y_s, y_t)$$

node potentials　　edge potentials

$$\psi_t(y_t) \propto e^{\eta_t y_t}$$

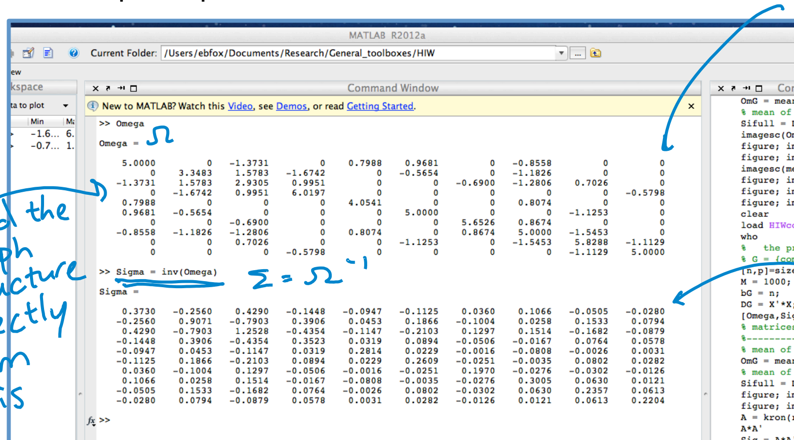
$$\psi_{st}(y_s, y_t) \propto e^{-\frac{1}{2} y_s \Omega_{st} y_t}$$

©Emily Fox 2013　　13

# Sparse Precision vs. Covariance

- For a sparse precision matrix, the covariance need not be

0's encode cond. ind. statements




```
>> Omega

Omega =

    5.0000         0   -1.3731         0    0.7988    0.9681         0   -0.8558         0         0
         0    3.3483    1.5783   -1.6742         0   -0.5654         0   -1.1826         0         0
   -1.3731    1.5783    2.9305    0.9951         0         0   -0.6900   -1.2806    0.7026         0
         0   -1.6742    0.9951    6.0197         0         0         0         0         0   -0.5798
    0.7988         0         0         0    4.0541         0         0    0.8074         0         0
    0.9681   -0.5654         0         0         0    5.0000         0         0   -1.1253         0
         0         0   -0.6900         0         0         0    5.6526    0.8674         0         0
   -0.8558   -1.1826   -1.2806         0    0.8074         0    0.8674    5.0000   -1.5453         0
         0         0    0.7026         0         0   -1.1253         0   -1.5453    5.8288   -1.1129
         0         0         0   -0.5798         0         0         0         0   -1.1129    5.0000

>> Sigma = inv(Omega)

Sigma =

    0.3730   -0.2560    0.4290   -0.1448   -0.0947   -0.1125    0.0360    0.1066   -0.0505   -0.0280
   -0.2560    0.9071   -0.7903    0.3906    0.0453    0.1866   -0.1004    0.0258    0.1533    0.0794
    0.4290   -0.7903    1.2528   -0.4354   -0.1147   -0.2103    0.1297    0.1514   -0.1682   -0.0879
   -0.1448    0.3906   -0.4354    0.3523    0.0319    0.0894   -0.0506   -0.0167    0.0764    0.0578
   -0.0947    0.0453   -0.1147    0.0319    0.2814    0.0229   -0.0016   -0.0808   -0.0026    0.0031
   -0.1125    0.1866   -0.2103    0.0894    0.0229    0.2609   -0.0251   -0.0035    0.0802    0.0282
    0.0360   -0.1004    0.1297   -0.0506   -0.0016   -0.0251    0.1970   -0.0276   -0.0302   -0.0126
    0.1066    0.0258    0.1514   -0.0167   -0.0808   -0.0035   -0.0276    0.3005    0.0630    0.0121
   -0.0505    0.1533   -0.1682    0.0764   -0.0026    0.0802   -0.0302    0.0630    0.2357    0.0613
   -0.0280    0.0794   -0.0879    0.0578    0.0031    0.0282   -0.0126    0.0121    0.0613    0.2204

>>
```

$\Omega$

$\Sigma = \Omega^{-1}$

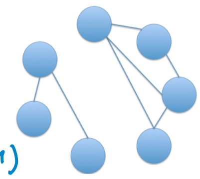
read the graph structure directly from this

does not imply sparsity of cov (ind. assump.)

$\Rightarrow$ y is still fully correlated!

©Emily Fox 2013　　16

# ML Estimation for Given Graph



- Assume a known graph $G = \{V,E\}$
- Rewrite log likelihood: $y^1, \dots, y^N$ N obs.

$$\log p(y|\theta) = \frac{N}{2}\log|\Omega| - \frac{1}{2}\sum_i \underbrace{(y^i-\mu)^T}_{x^T}\underbrace{\Omega}_{A}\underbrace{(y^i-\mu)}_{x}$$

$$= \frac{N}{2}\log|\Omega| - \frac{1}{2}\sum_i tr\left[(y^i-\mu)(y^i-\mu)^T\Omega\right]$$

$$\triangleq \frac{N}{2}\log|\Omega| - \frac{1}{2} tr(S_\mu \Omega)$$

$S_\mu = \sum_i (y^i-\mu)(y^i-\mu)^T$

Trace trick:
$x^TAx = tr(x^TAx)$
$= tr(xx^TA)$

matrix reference manual

$$L(\Omega) = \log|\Omega| - tr(S\Omega)$$

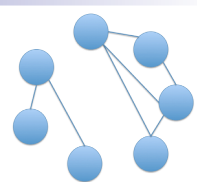
$S = \frac{1}{N}\sum_i (y^i-\mu)(y^i-\mu)^T$

In our case, $\mu = 0$




# ML Estimation for Given Graph

$$L(\Omega) = \log|\Omega| - \mathrm{tr}(S\Omega)$$



- Take gradient:

$$\nabla L(\Omega) = \Omega^{-1} - S$$

s.t. $\Omega_{st} = 0$ if $(s,t) \notin E$ ← linear constraint

$\Omega$ pos. def, sym. matrix

hard!!

- Many approaches to solving:
  - Barrier method – add penalty if $\Omega$ leaves the positive definite cone (Dahl et al. 2008)
  - Coordinate descent method (cf., Hastie et al. 2009)
  - …

# ML Estimation for Given Graph

- Can show that the optimal solution satisfies

$\hat{\Sigma}^{ML,G}_{st} = S_{st}$ if $(s,t) \in E$ match to sample cov.
if $s=t$

$\Omega_{st} = 0$ if $(s,t) \notin E$

- Example:

adj. matrix
$1 =$ edge

$$G = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \qquad S = \begin{pmatrix} 10 & 1 & 5 & 4 \\ 1 & 10 & 2 & 6 \\ 5 & 2 & 10 & 3 \\ 4 & 6 & 3 & 10 \end{pmatrix}$$

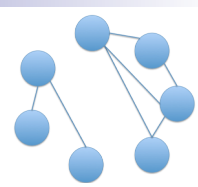
$$\Omega = \begin{pmatrix} \cdot & \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot \\ \cdot & 0 & \cdot & \cdot \end{pmatrix} \qquad \hat{\Sigma}^{ML,G} = \begin{pmatrix} 10 & 1 & 1.31 & 4 \\ 1 & 10 & 2 & 0.87 \\ 1.31 & 2 & 10 & 3 \\ 4 & 0.87 & 3 & 10 \end{pmatrix}$$



# Estimating Graph Structure

want to max

- To learn the structure of the Gaussian graphical model, we want to trade off fit and sparsity
  - Measure of fit: log likelihood
    $\log|\Omega| - tr(S\Omega) + const.$
  - Encouraging sparsity: $\Omega_{st} = 0 \Rightarrow$ no edge "sparsity"
    $\|\Omega\|_1 = \sum_{s,t} |\Omega_{st}|$ want to min



- Overall objective = "graphical LASSO" or "Glasso"

$$F(\Omega) = -\log|\Omega| + tr(S\Omega) + \lambda \|\Omega\|_1$$

Just as in LASSO, but w/ a matrix parameter and s.t. $\Omega \succ 0$

# Solving the Graphical LASSO

- Objective is convex, but non-smooth as in LASSO ... subgrad.
- Also, positive definite constraint!
- There are many approaches to optimizing the objective
  - Most common = coordinate descent akin to shooting algorithm (Friedman et al. 2008)
- Some issues…
  - Ballpark: several minutes for a 1000-variable problem
  - Algorithms scale as *O(d^3)*

Lots of recent literature on this...




# Faster Computations

From Daniela Witten's talk at JSM 2012:

1. The $j$th variable is unconnected from all others in the graphical lasso solution if and only if $|S_{ij}| \leq \lambda$ for all $i = 1, \ldots, j-1, j+1, \ldots, p$.
   sample cov is small relative to chosen penalty
2. Let $\mathbf{A}$ denote the $p \times p$ matrix whose elements take the form $A_{ii} = 1,\ A_{ij} = 1_{|S_{ij}| > \lambda}$. Then the connected components of $\mathbf{A}$ are the same as the connected components of the graphical lasso solution.

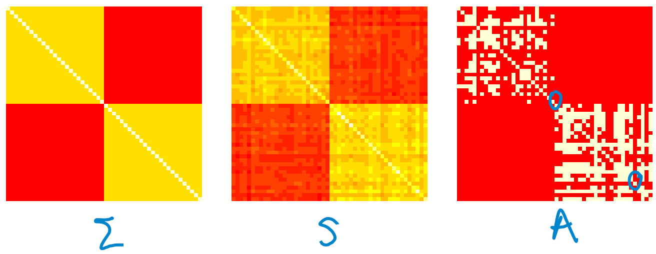
ind. on the thresholded values

We can obtain the *exact* right answer by solving the graphical lasso on each connected component separately!

Citations: Witten et al. JCGS 2011, Mazumder and Hastie JMLR 2012

# Covariance Screening for Glasso

From Daniela Witten's talk at JSM 2012:




- The solution to the graphical lasso problem with $\lambda = 0.7$ has five connected components (why 5?!)
- Perform graphical lasso on each component separately!
- Reduction in computational time: From $O(50^3)$ to $O(24^3)$.