VIASM Lectures on

Statistical Machine Learning for High Dimensional Data

John Lafferty and Larry Wasserman

University of Chicago &
Carnegie Mellon University

Outline

1. Regression
   - predicting Y from X
2. Structure and Sparsity
   - finding and using hidden structure
3. Nonparametric Methods
   - using statistical models with weak assumptions
4. Latent Variable Models
   - making use of hidden variables

Lecture 2

Structure and Sparsity

Finding hidden structure in data

Topics

- *Undirected graphical models*
- High dimensional covariance matrices
- Sparse coding

Undirected Graphs

Let $X = (X_1, \ldots, X_p)$. A graph $G = (V, E)$ has vertices V, edges E.
Independence graph has one vertex for each X_j.

means that

$$X \amalg Z \;\Big|\; Y$$

$V = \{X, Y, Z\}$ and $E = \{(X, Y), (Y, Z)\}$.

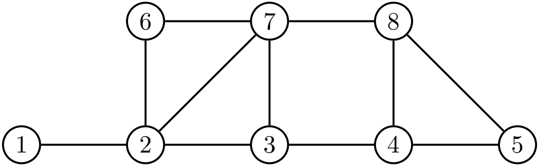
Markov Property

A probability distribution P satisfies the *global Markov property* with respect to a graph G if:

for any disjoint vertex subsets A, B, and C such that C separates A and B,

$$X_A \perp\!\!\!\perp X_B \;\Big|\; X_C.$$

Example



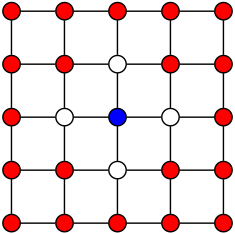
$C = \{3, 7\}$ separates $A = \{1, 2\}$ and $B = \{4, 8\}$. Hence,

$$\{X_1, X_2\} \amalg \{X_4, X_8\} \,\Big|\, \{X_3, X_7\}$$

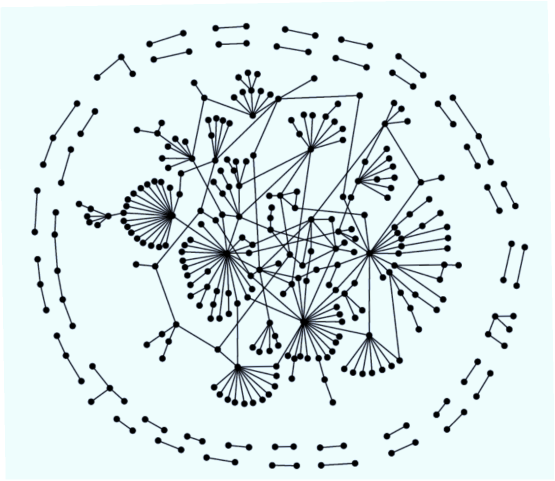
Example

A 2-dimensional grid graph.

The blue node is independent of the red nodes given the white nodes.



Example: Protein networks (Maslov 2002)



Distributions Encoded by a Graph

- $\mathcal{I}(G)$ = all independence statements implied by the graph G.
- $\mathcal{I}(P)$ = all independence statements implied by P.
- $\mathcal{P}(G) = \{P : \mathcal{I}(G) \subseteq \mathcal{I}(P)\}$.
- If $P \in \mathcal{P}(G)$ we say that *P is Markov to G*.
- The graph G represents the class of distributions $\mathcal{P}(G)$.
- Goal: Given $X^1, \ldots, X^n \sim P$ estimate G.

Gaussian Case

- If $X \sim N(\mu, \Sigma)$ then there is no edge between X_i and X_j if and only if
$$\Omega_{ij} = 0$$
where $\Omega = \Sigma^{-1}$.
- Given
$$X^1, \ldots, X^n \sim N(\mu, \Sigma).$$
- For $n > p$, let
$$\widehat{\Omega} = \widehat{\Sigma}^{-1}$$
and test
$$H_0 : \Omega_{ij} = 0 \quad \text{versus} \quad H_1 : \Omega_{ij} \neq 0.$$

Gaussian Case: $p > n$

Two approaches:

- parallel lasso (Meinshausen and Bühlmann)
- graphical lasso (glasso; Banerjee et al, Hastie et al.)

Parallel Lasso:

1. For each $j = 1, \ldots, p$ (in parallel): Regress X_j on all other variables using the lasso.
2. Put an edge between X_i and X_j if each appears in the regression of the other.

Glasso (Graphical Lasso)

The glasso minimizes:

$$-\ell(\Omega) + \lambda \sum_{j \neq k} |\Omega_{jk}|$$

where

$$\ell(\Omega) = \frac{1}{2} \left(\log |\Omega| - \mathrm{tr}(\Omega S) \right)$$

is the Gaussian loglikelihood (maximized over μ).

There is a simple blockwise gradient descent algorithm for minimizing this function. It is very similar to the previous algorithm.

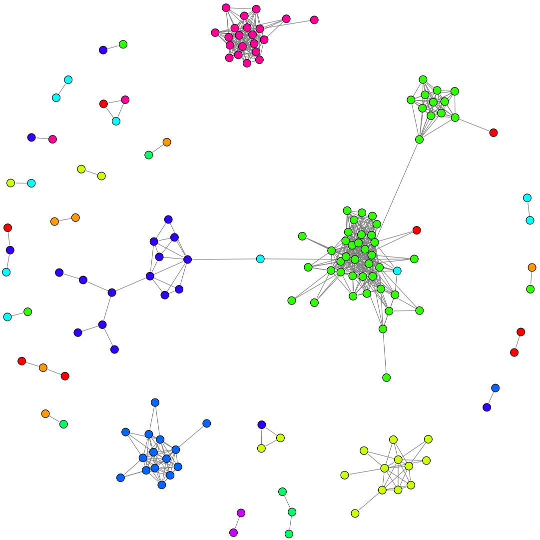
R packages: `glasso` and `huge`

Graphs on the S&P 500

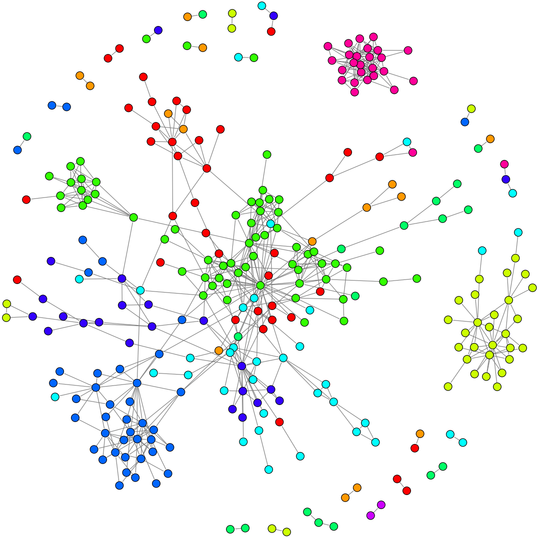
- Data from Yahoo! Finance (`finance.yahoo.com`).
- Daily closing prices for 452 stocks in the S&P 500 between 2003 and 2008 (before onset of the "financial crisis").
- Log returns $X_{tj} = \log\left(S_{t,j}/S_{t-1,j}\right)$.
- Winsorized to trim outliers.
- In following graphs, each node is a stock, and color indicates GICS industry.

```
Consumer Discretionary        Consumer Staples
Energy                        Financials
Health Care                   Industrials
Information Technology        Materials
Telecommunications Services   Utilities
```

S&P 500: Graphical Lasso



S&P 500: Parallel Lasso



Example Neighborhood

Yahoo Inc. (Information Technology):

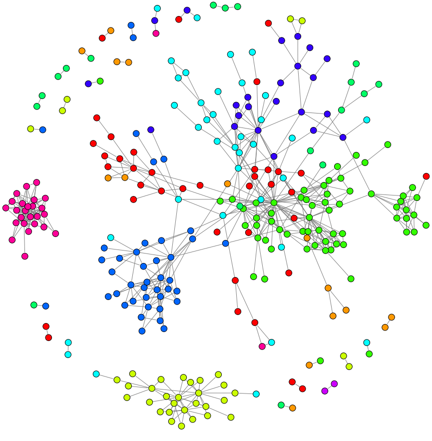
- Amazon.com Inc. (Consumer Discretionary)
- eBay Inc. (Information Technology)
- NetApp (Information Technology)

Example Neighborhood

Target Corp. (Consumer Discretionary):

- Big Lots, Inc. (Consumer Discretionary)
- Costco Co. (Consumer Staples)
- Family Dollar Stores (Consumer Discretionary)
- Kohl’s Corp. (Consumer Discretionary)
- Lowe’s Cos. (Consumer Discretionary)
- Macy’s Inc. (Consumer Discretionary)
- Wal-Mart Stores (Consumer Staples)

Parallel vs. Graphical



Choosing λ

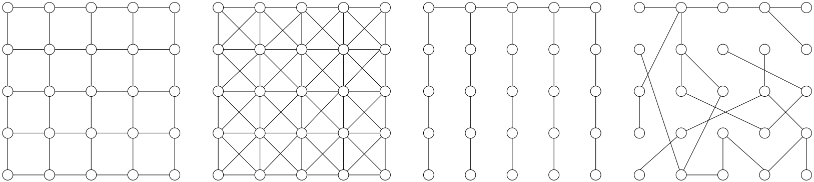
Can use:

1. Cross-validation
2. BIC = log-likelihood - $(p/2)\log n$
3. AIC = log-likelihood - p

where p = number of parameters.

Discrete Graphical Models

Let $G = (V, E)$ be an undirected graph on $m = |V|$ vertices



- (Hammersley, Clifford) A positive distribution p over random variables $Z_1, \ldots, Z_n$ that satisfies the Markov properties of graph G can be represented as

$$p(Z) \propto \prod_{c \in \mathcal{C}} \psi_c(Z_c)$$

where $\mathcal{C}$ is the set of cliques in the graph G.

Discrete Graphical Models

- Positive distributions can be represented by an exponential family,

$$p(Z; \beta^*) \propto \exp\left(\sum_{c \in \mathcal{C}} \beta^*_c \phi_c(Z_c)\right)$$

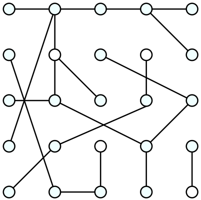
- Special case: Ising Model (binary Gaussian)

$$p(Z; \beta^*) \propto \exp\left(\sum_{i \in V} \beta^*_i Z_i + \sum_{(i,j) \in E} \beta^*_{ij} Z_i Z_j\right).$$

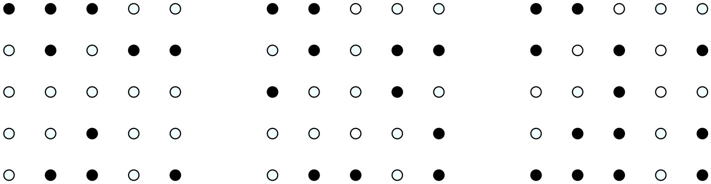
  Here, the set of cliques $\mathcal{C} = \{V \cup E\}$, and the potential functions are $\{Z_i, \, i \in V\} \cup \{Z_i Z_j, \, (i,j) \in E\}$.

Graph Estimation

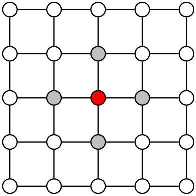
- Given n i.i.d. samples from an Ising distribution, $\{Z^s, s = 1, \ldots, n\}$, identify underlying graph structure.



- Multiple examples are observed:



Local Distributions



- Consider Ising model $p(Z; \beta^*) \propto \exp\left\{\sum_{(i,j)\in E} \beta^*_{ij} Z_i Z_j\right\}$.
- Conditioned on $(z_2, \ldots, z_p)$, variable $Z_1 \in \{-1, +1\}$ has probability mass function given by a logistic function,

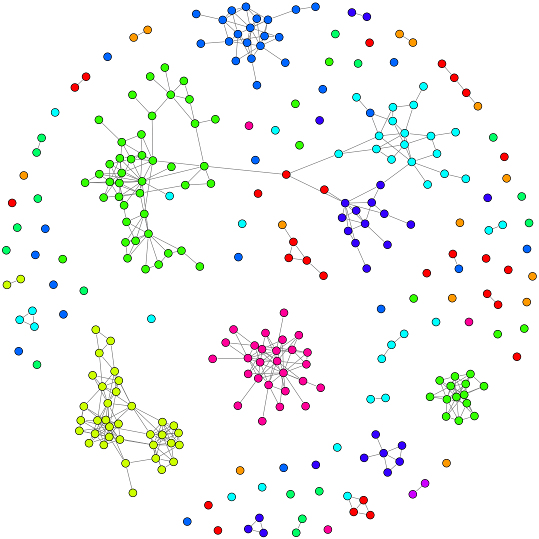
$$\mathbb{P}(Z_1 = 1 \mid z_2, \ldots, z_p) \quad = \quad \frac{1}{1 + \exp\left(\sum_{j\in\mathcal{N}(1)} \beta^*_{1j} z_j\right)}.$$

Parallel Logistic Regressions

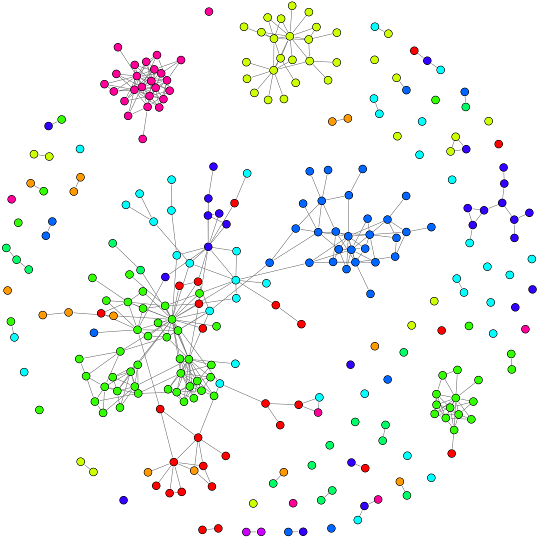
Approach of Ravikumar, Wainwright and Lafferty (Ann. Stat., 2010):

- Inspired by Meinshausen & Bühlmann (2006) for Gaussian case
- Recovering graph structure equivalent to recovering neighborhood structure $\mathcal{N}(i)$ for every $i \in V$
- *Strategy:* perform ℓ_1 regularized logistic regression of each node Z_i on $Z_{\setminus i} = \{Z_j,\ j \neq i\}$ to estimate $\widehat{\mathcal{N}}(i)$.
- Error probability $\mathbb{P}\left(\widehat{\mathcal{N}}(i) \neq \mathcal{N}(i)\right)$ must decay exponentially fast.

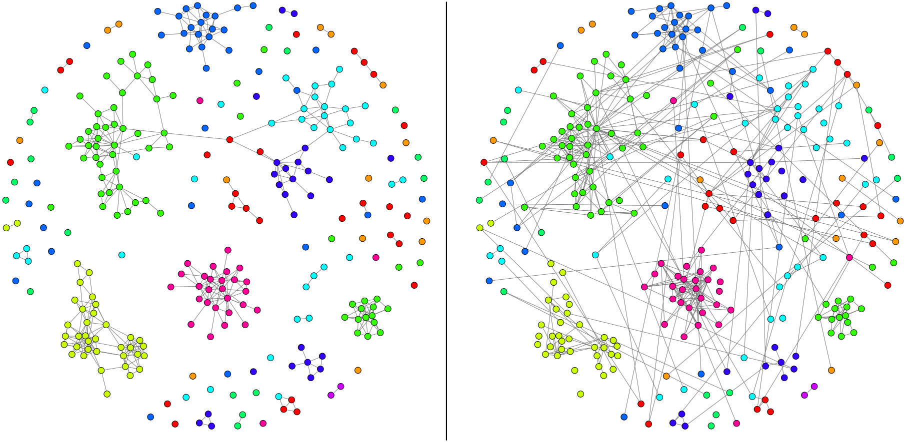
S&P 500: Ising Model (Price up or down?)



S&P 500: Parallel Lasso

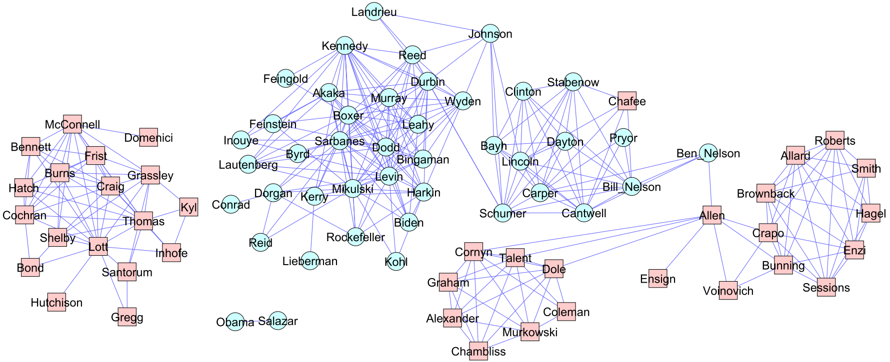


Ising vs. Parallel Lasso



Voting Data

Example of Banerjee, El Gahoui, and d'Asepremont (JMLR, 2008).
Voting records of US Senate, 2006-2008



Statistical Scaling Behavior

Maximum degree d of the p variables. Sample size n must satisfy

$$\text{Ising model:} \quad n \geq d^3 \log p$$

$$\text{Graphical lasso:} \quad n \geq d^2 \log p$$

$$\text{Parallel lasso:} \quad n \geq d \log p$$

$$\text{Lower bound:} \quad n \geq d \log p$$

- Each method makes different *incoherence assumptions*.
- Intuitively, correlations between unrelated variables not too large.

Topics

- Undirected graphical models
- *High dimensional covariance matrices*
- Sparse coding

High Dimensional Covariance Matrices

Let $X = (X_1, \ldots, X_p)$ (for example, p stocks). Suppose we want to estimate Σ, the covariance matrix of X. Here $\Sigma = [\sigma_{jk}]$ where $\sigma_{jk} = \mathrm{Cov}(X_j, X_k)$.

The data are n random vectors $X^1, \ldots, X^n \in \mathbb{R}^p$. Let

$$S = \frac{1}{n}\sum_{i=1}^{n}(X^i - \overline{X})(X^i - \overline{X})^T$$

be the sample covariance matrix, where $\overline{X} = (\overline{X}_1, \ldots, \overline{X}_p)^T$ and

$$\overline{X}_j = \frac{1}{n}\sum_{i=1}^{n} X_j^i$$

is the mean of the j^{th} variable. Let s_{jk} denote the (j, k) element of S.
If $p < n$, then S is a good estimator of Σ.

Bounds on Sample Covariance

Results of Vershynin show that for sub-Gaussian families F

$$\sup_F \|\widehat{\Sigma} - \Sigma\|_2 = O_P\left(\sqrt{\frac{p}{n}}\right)$$

where $S = \widehat{\Sigma} = \frac{1}{n}\sum_{i=1} X_i X_i^T$ is the sample covariance.

What if $p > n$?

If $p > n$ then S is a poor estimator of Σ. But suppose that Σ is *sparse*: most σ_{jk} are small.

Define the *threshold estimator* $\widehat{\Sigma}_t$. The (j, k) element of $\widehat{\Sigma}_t$ is

$$\widehat{\sigma}_{jk} = \begin{cases} s_{jk} & \text{if } |s_{jk}| \geq t \\ 0 & \text{if } |s_{jk}| < t. \end{cases}$$

Bickel and Levina (2008) show that, if Σ is sparse, then $\widehat{\Sigma}_t$ is a good estimator of Σ. (It is not positive-semi-definite (PSD) but can be made PSD by doing a SVD and getting rid of negative singular values.)

Bounds on Thresholded Covariance

Bickel and Levina show that

$$\|\widehat{\Sigma}_t - \Sigma\|_2 = O_P\left(c_0(p)t^{1-q} + c_0(p)t^{-q}\sqrt{\frac{\log p}{n}}\right)$$

for the class of covariance matrices

$$U_q = \left\{\Sigma \ : \ \max_i \sigma_{ii} \leq M, \ \max_i \sum_{j=1}^{p} |\sigma_{ij}|^q \leq c_0(p)\right\}$$

How To Choose the Threshold

1. Split the data into two halves giving sample covariance matrices S_1, S_2.
2. Threshold S_1 to get $\widehat{\Sigma}_{t,1}$.
3. Repeat N times:

$$(\widehat{\Sigma}_{t,1,1}, S_{2,1}), \ldots, (\widehat{\Sigma}_{t,1,s}, S_{2,s}), \ldots, (\widehat{\Sigma}_{t,1,N}, S_{2,N}).$$

4. Let

$$\widehat{R}(t) = \frac{1}{N}\sum_{s=1}^{N}\left\|\widehat{\Sigma}_{t,1,s} - S_{2,s}\right\|_F^2$$

   where $\|A\|_F^2 = \sum_{j,k} A_{jk}^2 = \mathrm{trace}(AA^T)$ is the Frobenius norm.

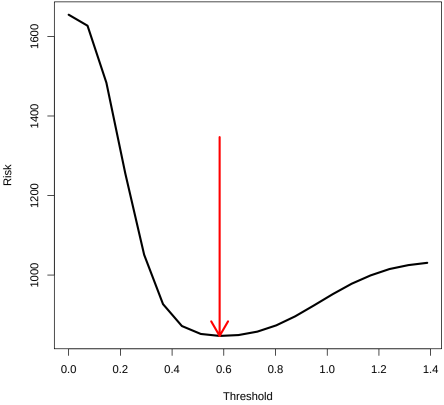
5. Choose t to minimize $\widehat{R}(t)$.

Example

We take $n = 100$, $p = 200$ and

$$X^1, \ldots, X^n \sim N(0, \Sigma)$$

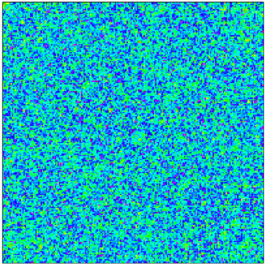
where $\sigma_{jk} = \rho^{|i-j|}$ and $\rho = 0.2$.



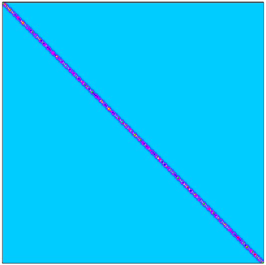
Example

We find that

$$\|\Sigma - S\|_F^2 = 420 \qquad \|\Sigma - \widehat{\Sigma}_t\|_F^2 = 20$$
$$\|\Sigma - S\|_2 = 4.7 \qquad \|\Sigma - \widehat{\Sigma}_t\|_2 = 0.6$$



$\Sigma - S$



$\Sigma - \hat{\Sigma}$

Factor Models

Covariance under a factor model:

$$Y = Bf + \epsilon$$

$Y \in \mathbb{R}^p$, $B \in \mathbb{R}^{p \times k}$, for k *known* factors f_j. So

$$\Sigma = B\,\mathrm{cov}(f)\,B^T + I.$$

Natural estimate is the plugin estimator

$$\widehat{\Sigma}_n = \widehat{B}_n\,\widehat{\mathrm{cov}}(f)\,\widehat{B}_n^T + I.$$

where $\widehat{B}_n$ are estimated regression coefficients. Fan, Fan and Lv (2008) study this in the high dimensional setting.

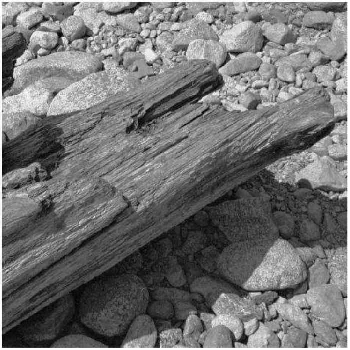
Topics

- Undirected graphical models
- High dimensional covariance matrices
- *Sparse coding*

Sparse Coding

Motivation: understand neural coding (Olshausen and Field, 1996).

original image



sparse representation

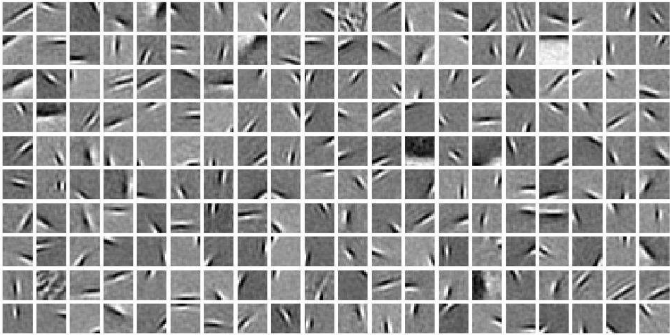
Codewords/patch 8.14, RSS 0.1894

Sparse Coding

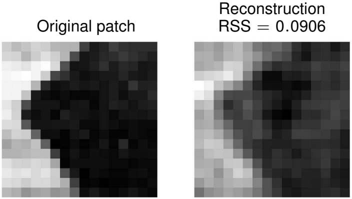
Mathematical formulation of dictionary learning:

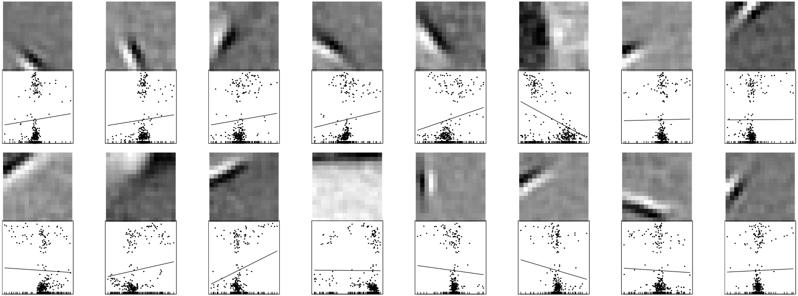
$$\min_{\alpha, X} \quad \sum_{g=1}^{G} \left\{ \frac{1}{2n} \left\| y^{(i)} - X\alpha^{(i)} \right\|_2^2 + \lambda \left\| \alpha^{(i)} \right\|_1 \right\}$$

$$\text{such that} \quad \|X_j\|_2 \leq 1$$



Sparse Coding for Natural Images



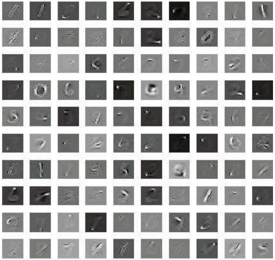


Properties

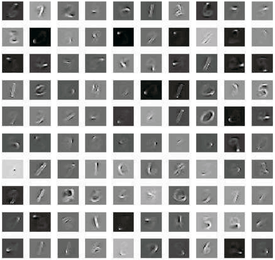
- Provides high dimensional, nonlinear representation
- Sparsity enables codewords to specialize, isolate “features”
- Overcomplete basis, adapted to data automatically
- Frequentist form of topic modeling, soft VQ

Sparse Coding for Computer Vision

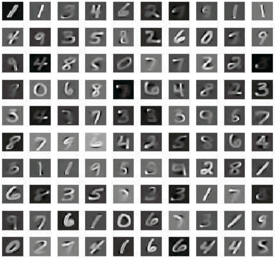
source: Kai Yu



Error: 4.54%



Error: 3.75%



Error: **2.64%**

- Best accuracy when learned codewords are like digits
- Advanced versions are state-of-art for object classification

Sparse Coding for Multivariate Regression

- Intuition of sparse coding extends to multivariate regression with grouped data (e.g., time series over different blocks of time).
- Estimate a regression matrix for each group.
- Each estimate is a sparse combination of a common dictionary of low-rank matrices.
- Low-rank dictionary elements are estimated by pooling across groups.

Problem Formulation

- Data fall into G groups, indexed by $g = 1, \ldots, G$
- Covariate $X_i^{(g)} \in \mathbb{R}^p$ and response $Y_i^{(g)} \in \mathbb{R}^q$, model

$$Y_i^{(g)} = B^{*(g)} X_i^{(g)} + \epsilon_i^{(g)}$$

- Goal: estimate $B^{*(g)} \in \mathbb{R}^{q \times p}$ with

$$\widehat{B}^{(g)} = \sum_{k=1}^{K} \widehat{\alpha}_k^{(g)} D_k$$

where each D_k is low rank, $\widehat{\alpha}^{(g)} = (\widehat{\alpha}_1^{(g)}, \ldots, \widehat{\alpha}_K^{(g)})$ is sparse

Interlude: Low-Rank Matrices

- 2×2 symmetric matrices:

$$X = \begin{pmatrix} x & y \\ y & z \end{pmatrix}$$

- By scaling, can assume $|x + z| = 1$.

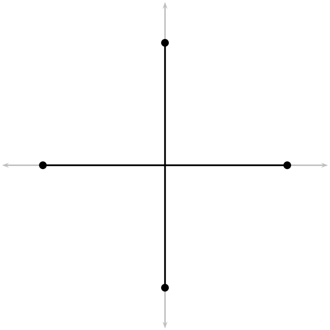
$$X \text{ has rank one iff } x^2 + 2y^2 + z^2 = 1$$

- Union of two ellipses in $\mathbb{R}^3$.
- Convex hull is a cylinder.

Recall: Sparse Vectors and ℓ_1 Relaxation

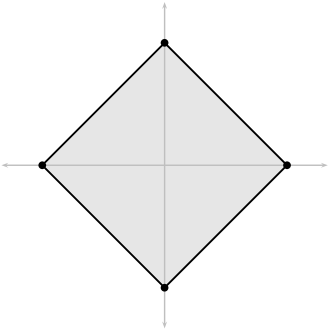
sparse vectors

$\|X\|_0 \leq t$



convex hull

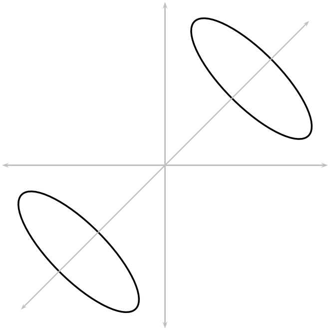
$\|X\|_1 \leq t$



Low-Rank Matrices and Convex Relaxation

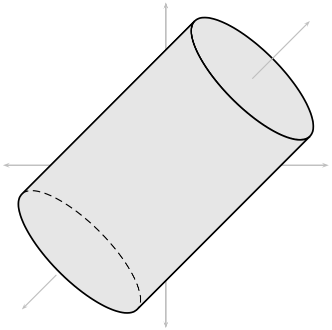
low rank matrices

$\text{rank}(X) \leq t$



convex hull

$\|X\|_* \leq t$



Nuclear Norm Regularization

Nuclear norm $\|X\|_*$ of $p \times q$ matrix X

$$\|X\|_* = \sum_{j=1}^{\min(p,q)} \sigma_j(X)$$

Sum of singular values. (a.k.a. *trace norm* or *Ky-Fan norm*)

Generalization to matrices of ℓ_1 norm for vectors.

Nuclear Norm Regularization

Algorithms for nuclear norm minimization are a lot like iterative soft thresholding for lasso problems.

To project a matrix B onto the nuclear norm ball $\|X\|_* \leq t$:

- Compute the SVD:

$$B = U\,\mathrm{diag}(\sigma)\,V^T$$

- Soft threshold the singular values:

$$B \leftarrow U\,\mathrm{diag}(\mathrm{Soft}_\lambda(\sigma))\,V^T$$

Conditional Sparse Coding

- Objective function:

$$f(\alpha, D) = \frac{1}{G} \sum_{g=1}^{G} \left\{ \frac{1}{n} \| Y^{(g)} - \Big(\sum_{k=1}^{K} \alpha_k^{(g)} D_k \Big) X^{(g)} \|_F^2 + \lambda \|\alpha^{(g)}\|_1 \right\}$$

minimized over $D_k \in \mathcal{C}(\tau)$,

$$\mathcal{C}(\tau) = \left\{ D \in \mathbb{R}^{q \times p} : \|D\|_* \leq \tau \text{ and } \|D\|_2 \leq 1 \right\}$$

- Dictionary entries D_k are shared across groups; nuclear norm constraint forces them to be low rank

Conditional Sparse Coding

Input: Data $\{(Y^{(g)}, X^{(g)}\}_{g=1,...,G}$, parameters λ and τ

1. Initialize dictionary $\{D_1, ..., D_K\}$ as random rank one matrices
2. Alternate following steps until convergence of $f(\alpha, D)$:

   a. Encoding step: $\{\alpha^{(g)}\} \leftarrow \arg\min_{\alpha^{(g)}} f(\alpha, D)$
   b. Learning step: $\{D_k\} \leftarrow \arg\min_{D_k \in \mathcal{C}(\tau)} f(\alpha, D)$

$$f(\alpha, D) = \frac{1}{G} \sum_{g=1}^{G} \left\{ \frac{1}{n} \| Y^{(g)} - \Big(\sum_{k=1}^{K} \alpha_k^{(g)} D_k \Big) X^{(g)} \|_F^2 + \lambda \| \alpha^{(g)} \|_1 \right\}$$

Related Methods

- Low-rank regression: Yuan et al. (2007), Negahban and Wainwright (2011)
- Multi-task learning: Evgeniou and Pontil (2004), Maurer and Pontil (2010)

Example with Equities Data

- 29 companies in single industry sector, from 2002 to 2007
- One day log returns, $Y_t = \log S_t / S_{t-1}$, X_t lagged values
- Grouped in 35 day periods

	30 days back	50 days back	90 days back	Sparse Coding
Correlation	-0.000433	0.0527	0.0513	**0.0795**
Predictive R^2	-0.0231	-0.0011	0.00218	**0.0042**

Sparse Coding for Covariance Estimation

- Sparse code the group sample covariance matrices

$$\widehat{S}_n^{(g)} = \frac{1}{n}\sum_{i=1}^{n} Y_i^{(g)} Y_i^{(g)T}$$

- Objective function:

$$f(\alpha, \beta, D) = \frac{1}{G}\sum_{g=1}^{G}\left\{\frac{1}{n}\|\widehat{S}_n^{(g)} - \mathrm{diag}(\beta) - \sum_{k=1}^{K}\alpha_k^{(g)} D_k\|_F^2 + \lambda\|\alpha^{(g)}\|_1\right\}$$

minimized over $D_k \in \mathcal{C}(\tau)$,

$$\mathcal{C}(\tau) = \{D \succeq 0, \|D\|_* \leq \tau \text{ and } \|D\|_2 \leq 1\}$$

- Optimization over $\alpha^{(g)}$ by solving semidefinite program or nonnegative lasso

“Read the Mind” with fMRI

- Subject sees one of 60 words, each associated with a semantic vector; fMRI measures neural activity.
- Can we predict the semantic vector based on the neural activity?

“Read the Mind” with fMRI

- Subject sees one of 60 words, each associated with a semantic vector; fMRI measures neural activity.
- Can we predict the semantic vector based on the neural activity?

Multivariate Regression

$$\underbrace{Y}_{q\times n} = \underbrace{B}_{q\times p}\underbrace{X}_{p\times n} + \epsilon$$

p: dimension of neural activity ($\sim$ 400)
q: dimension of semantic vector ($\sim$ 200)
n: sample size ($\sim$ 60)

Mind Reading

Many different subjects; we have a data set for each subject.
Everyone's brain works differently—but not completely differently.

Data is grouped
For groups $g = 1, ..., G$

$$
\begin{aligned}
Y^{(1)} &= B^{(1)} X^{(1)} + \epsilon^{(1)} \\
Y^{(2)} &= B^{(2)} X^{(2)} + \epsilon^{(2)} \\
&\vdots \\
Y^{(G)} &= B^{(G)} X^{(G)} + \epsilon^{(G)}
\end{aligned}
$$

- Slight generalization of multi-task learning
- Many other applications

Experiments

- Alternating optimization relatively well-behaved.
- Improved mind-reading accuracy statistically significantly on 4 subjects. Degraded on 1 subject.
- Learned coefficients indeed sparse.

	Subj A	B	C	D	E	F	G	H	I
Dictionary	0.8833	0.8667	0.9000	0.9333	0.8333	0.7500	0.9000	0.7833	0.6667
Separate	**0.9500**	0.7000	0.9167	0.8167	0.8167	0.7667	0.8000	0.6667	0.6333
Confidence	0.6-	0.92+	0.05-	0.86+	0.03+	0.02-	0.70+	0.65+	0.07+

Theory

We analyze risk consistency, in worst case under weak assumptions.
We analyze output of non-convex procedure with initial randomization.

Theory

We analyze risk consistency, in worst case under weak assumptions.
We analyze output of non-convex procedure with initial randomization.

- With random initial dictionary, need to learn sets of dense coefficients
- Achieve good performance if learned coefficients of learned dictionary are sparse

Summary

- Undirected graphs represent conditional independence assumptions.
- Two methods for Gaussian graphical models: Parallel lasso and graphical lasso.
- Discrete graphical models are more difficult; parallel sparse logistic regression can be effective.
- Thresholding sample covariance can estimate sparse covariance matrices in high dimensions.
- Sparse coding efficiently represents high dimensional signals or regression models.