Statistical Machine Learning: Dimension reduction and graphical models

Daniel J. McDonald

Indiana University, Bloomington

mypage.iu.edu/∼dajmcdon

February 24-26, 2015

What is Clustering?

All the previous applications presumed that there is a response Y

However, in some cases, there is no response at all.

- A large curpus of emails sent at the Enron main office right before it collapsed
- Everyone's cell phone behavior and location in a particular city (these data sets do exist)
- The relationship between all stocks on the S&P 500
- A cancer researcher might assay gene expression levels in a group of patients with different cancers

What is Clustering?

All the previous applications presumed that there is a response Y

However, in some cases, there is no response at all.

- A large curpus of emails sent at the Enron main office right before it collapsed
- Everyone's cell phone behavior and location in a particular city (these data sets do exist)
- The relationship between all stocks on the S&P 500
- A cancer researcher might assay gene expression levels in a group of patients with different cancers

An Overview of Clustering

The idea is to find paterns in the data. However, we don't have a supervisor (Y) and hence clustering is sometimes known as unsupervised learning.

Clustering is more difficult than classification/regression because solutions are vague and often unverifiable.

The set-up

Suppose we have observations

$$X_1, \ldots, X_n$$

Here, we want to find a relationship between the X's, commonly by grouping them (putting them into 'clusters').

This is fundamentally different from our previous discussions, as there is no notion of 'prediction' accuracy which we are trying to maximize.

For this talk, clustering will be informal exploratory data analysis via lower dimensional embeddings.

There are many, many other approaches, however.

Lower dimesional (metric) embeddings

Spectral connectivity analysis (SCA)

- Linear and nonlinear
- Dimension reduction or feature creation
- Examples: PCA and Fisher discriminant analysis, Locally linear embeddings, Hessian maps, Laplacian eigenmaps
- Useful tools as inputs to classification, clustering, and regression

Principal component analysis

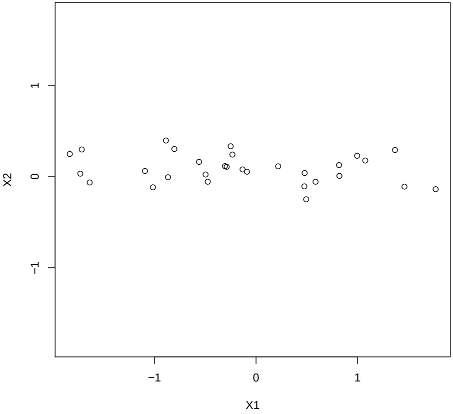
PCA

Collect data: $X_1, \ldots, X_n$ where $X_i \in \mathbb{R}^p$.

1. Center and scale the data matrix $\mathbb{X} \in \mathbb{R}^{n \times p}$
2. Compute the spectral decomposition of $\mathbb{X}^\top \mathbb{X} = VD^2V^\top$
   [Could (and should!) use SVD of $\mathbb{X} = UDV^\top$]
3. The principal component scores are in $\mathbb{R}^n$. These are the coordinates of the observations in each PC. ($UD = \mathbb{X}V$)
4. The principal component loading vectors are in $\mathbb{R}^p$. This is the rotation needed to change the alignment of the original data to the PC axis (V).

Lower dimensional embeddings: Toy example

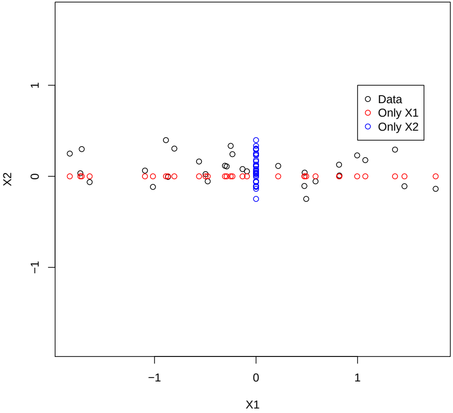
Suppose we have predictors X1 and X2



LOWER DIMENSIONAL EMBEDDINGS: TOY EXAMPLE

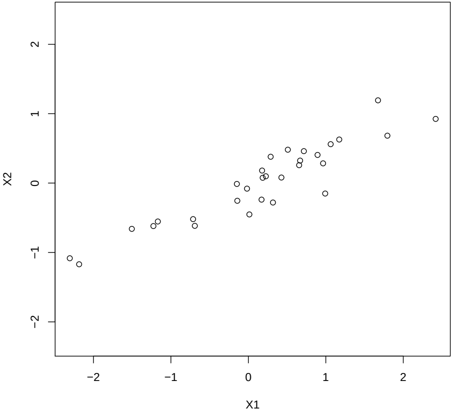
A lower dimensional embedding is given by

- Using the red dots (that is, by setting X2 to zero):
- Using the blue dots (that is, by setting X1 to zero):



Lower dimensional embeddings: Toy example

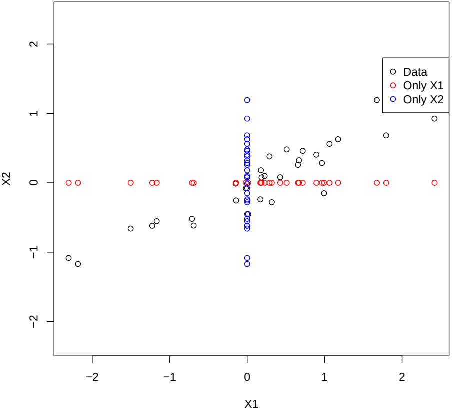
An important feature of the First Example is that X1 and X2 aren't correlated with each other. What if they are correlated?



LOWER DIMENSIONAL EMBEDDINGS: TOY EXAMPLE

A lower dimensional embedding is given by

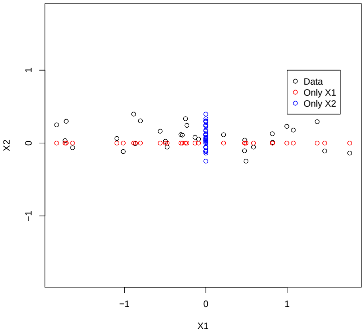
- Using the red dots (that is, by setting X2 to zero):
- Using the blue dots (that is, by setting X1 to zero):

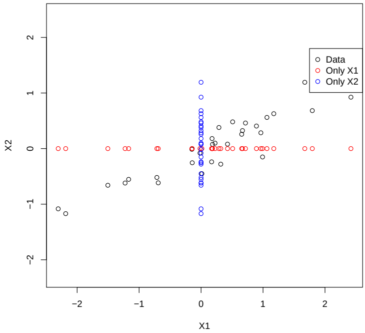


LOWER DIMENSIONAL EMBEDDINGS: COMPARISON OF EXAMPLES

The second example loses much more information with this simplistic dimension reduction strategy.

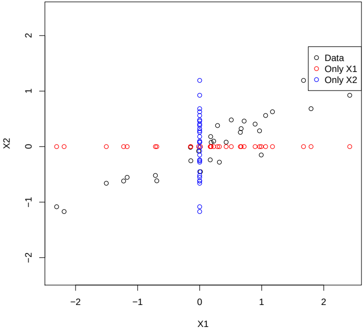
However, if we can find the rotation between the examples, then we can use this simple approach.

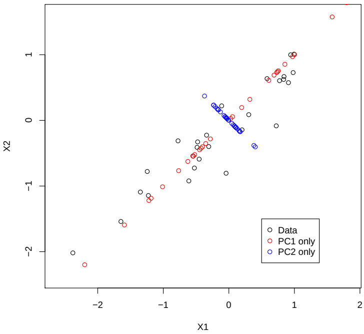




LOWER DIMENSIONAL EMBEDDINGS

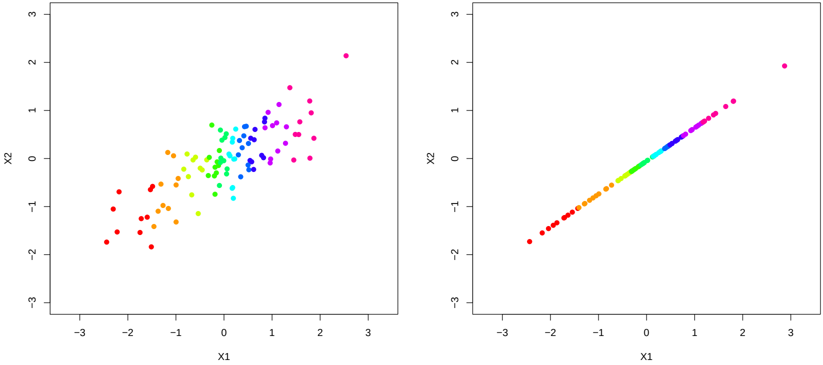
It turns out that Principal Components Analysis (PCA) gives us exactly this rotation (the matrix V).





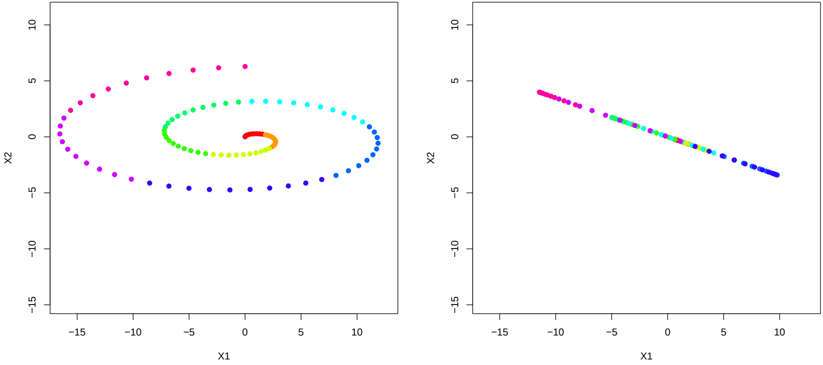
When PCA Works Well

PCA can do effective dimension reduction (that is, explain most of the data with $m < p$ components) as long as the data can be efficiently represented as 'lines' (or planes, or hyperplanes). So, in two dimensions:



When PCA doesn't work well

What about other data structures? Again in two dimensions



Here, we have failed miserably.

Explanation

- PCA wants to minimize distances (equivalently maximize variance). This means it ‘slices’ through the data at the ‘meatiest’ point, and then the next one, and so on. If the data are ‘curved’ this is going to induce artifacts.
- PCA also looks at things as being ‘close’ if they are near each other in a Euclidean sense

  [this is essentially all correlation is].
- On the spiral, our intuition says that things are ‘close’ only if the distance is constrained to go along the curve. In other words, purple and blue are close, blue and red are not.

Nonlinear embeddings

Laplacian Eigenmaps

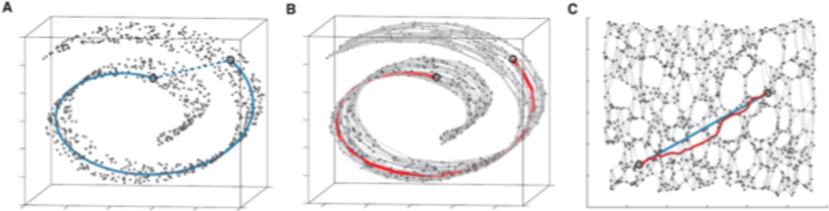
In order to use the intuitive distance, we need to know the geometry of the data. This needs to be estimated.

We can get an estimate of the distance in the unknown geometry that the data come from (known as a manifold) by altering the usual Euclidean distance.

Some notes:

- The name ‘Laplacian Eigenmaps’ comes from getting the eigenvector decomposition of the Laplacian restricted to the manifold (which is the second derivative version of the gradient).
- If the manifold is smooth, then local Euclidean distance is an approximation to the distance on the manifold.

Local Euclidean distance approximates the geodesic



The red line is the local Euclidean path between the two points, while the blue line is the path along the manifold.

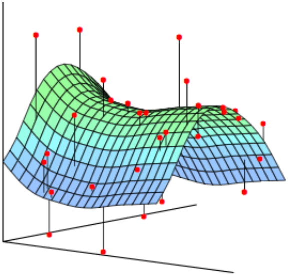
Source: James, Witten, Hastie, Tibshirani (2013)

What is a manifold?

Let's think of a manifold as a lower dimensional structure in our data (that is, $\mathbb{R}^p$).

If that structure is linear, then Euclidean distance is still a fine choice

If that structure is nonlinear, then Euclidean distance isn't applicable:

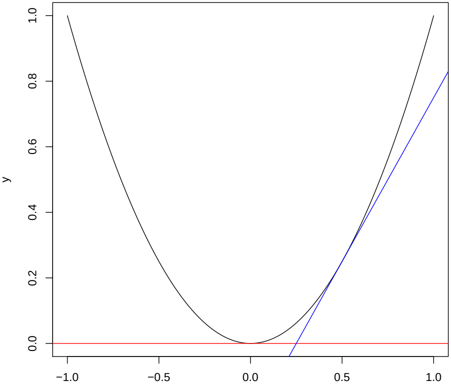


Source: James, Witten, Hastie, Tibshirani (2013)

What is a manifold?

How good of an approximation is Euclidean distance? This question is equivalent to how asking: how quickly does the tangent space change? In 1-D, the tangent space is just the first derivative at that point:

$$f(x) = x^2 \Rightarrow f'(x) = 2x.$$



What is a Manifold?

How quickly does the tangent space change? Well, this is the second derivative:

$$f(x) = x^2 \Rightarrow f''(x) = 2$$

Therefore, the quality of the (local) Euclidean distance, depends on the second derivative.

In higher dimensions, the second derivative is known as the Laplacian:

$$\sum_j \frac{\partial^2 f}{\partial x_j^2}$$

Note: This is also known as the divergence of the gradient.

What are Laplacian Eigenmaps, then?

If we think of the Laplacian as an operator mapping a function to the divergence of its gradient, it turn out it looks almost like a matrix operation.

Key Idea: **We can get the Eigenvectors/Eigenvalues of this operator. Analogously to PCA, we can now do inference with these Eigenvectors.**

Laplacian Eigenmaps

Collect data: $X_1, \ldots, X_n$ where $X_i \in \mathbb{R}^p$.

1. Form the distance matrix $\Delta_{ij} = ||X_i - X_j||_2^2$.
2. Compute
$$\mathbb{K} = \exp\left(-\frac{\Delta}{\gamma}\right)$$
3. Form the Laplacian $\mathbb{L} = \mathbb{I} - \mathbb{M}^{-1}\mathbb{K}$,
$$\mathbb{M} = \texttt{diag}(\texttt{rowSums}(\mathbb{K}))$$
4. Compute the spectrum: $\mathbb{L} = U\Sigma U^\top$.
5. Return U_d, where U_d corresponds to the smallest d (nontrivial) eigenvalues of $\mathbb{L}$

   (Note that the eigenvectors of $\mathbb{L}$ and $\mathbb{M}^{-1}\mathbb{K}$ are the same, but $\Sigma(\mathbb{L}) = \mathbb{I} - \Sigma(\mathbb{M}^{-1}\mathbb{K})$)

Deeper investigation

1. Form the distance matrix Δ.

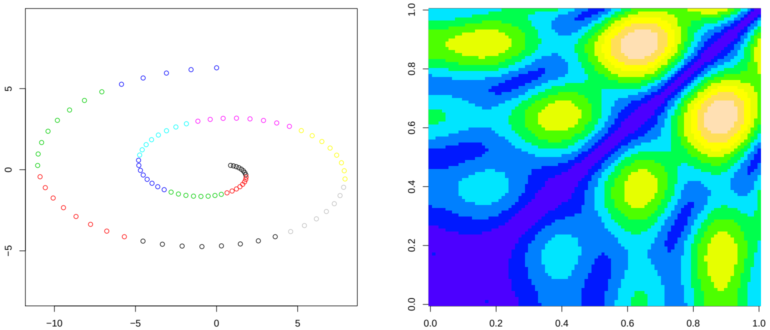


Figure: If we think about the center as 0 and the last blue circle as 1, then each entry the plot on the Right is the Euclidean distance between each data point on the plot on the Left (that is, Δ). The color on the Right plot goes from purple (small distance) to beige/pink (large distance).

DEEPER INVESTIGATION

2. Exponentiate Δ to form $\mathbb{K}$ for some fixed γ.

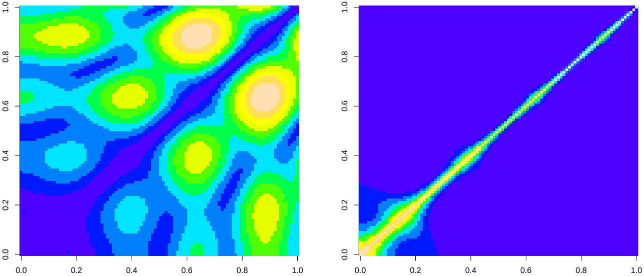
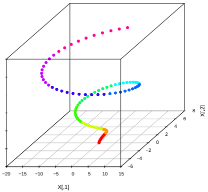
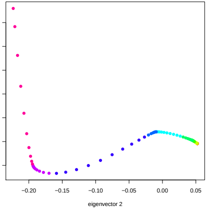


FIGURE: The Left plot is Δ and the Right plot is $\mathbb{K}$ for $\gamma = 0.95$.

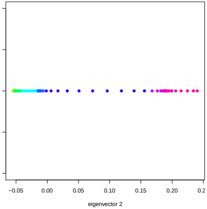
Spiral in $\mathbb{R}^3$



Original data



1^{st} & 2^{nd} nontrivial eigenvectors



1-dimensional

Other methods

We have only scratched the surface of dimension reduction, and barely talked about clustering at all.

Dimension reduction:

- principal components regression (PCR),
- partial least squares (PLS),
- multi-dimensional scaling (MDS),
- factor analysis, ...

Clustering:

- K-nearest neighbors (KNN),
- K-means,
- hierarchical with various linkages
(ie: single, complete, average, centroid, ...)
- non-negative matrix factorization
- naive Bayes ...

Graphical Models

Conditional independence

The core idea encoded in graphical models are conditional independence relations

A priori independent causes of an event can become not independent with a new measurement

Conditional independence

Example: Suppose you live in Los Angeles and are on a business trip to New York.

Your phone rings to notify you that your home security system has been activated

Simultaneously, you notice a news report that there has been an earthquake in LA

Given that you know from prior experience that earthquakes sometimes cause false alarms, you feel that an actual burglary is less likely.

GRAPHS

The expression of conditional independence relations can be expressed with a graph

A graph is a pair $G = \{V, E\}$, where

- V is a set of vertices
- E is a set of edges

  (Really, E is a set of (possibly ordered) pairs from V)

For our purposes, each vertex corresponds to a random variable

Each edge represents some aspect of their joint distribution

(For our purposes, we will only consider undirected graphs and hence the ordering doesn't matter)

GRAPHS

Let $x = (X_1, \ldots, X_p)^\top \sim \mathbb{P}$

A graph G for $\mathbb{P}$ has p vertices (aka nodes)

The crucial aspect is that the absence of an edge encodes conditional independence

$$\{j, k\} \notin E \Rightarrow X_j \perp X_k | \text{rest}$$

(This a Markov property that is a bit technical in full generality. See http://www.stat.cmu.edu/ larry/=sml/GraphicalModels.pdf for an indepth discussion)

EXAMPLE:

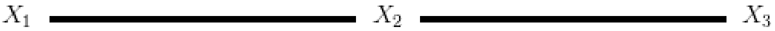


FIGURE: *

$$X_1 \perp X_3 | X_2$$

Statistical graphical models

The Markov part is tricky as there isn't, in general, a 1-1 map between $\mathbb{P}$ and G

For our purposes, let's somewhat reductively define the following

- $I(G) =$ all independence statements implied by G
- $I(\mathbb{P}) =$ all independence statements implied by $\mathbb{P}$
- $\mathcal{P}(G) = \{\mathbb{P} : I(G) \subseteq I(\mathbb{P})\}$
- If $\mathbb{P} \in \mathcal{P}(G)$ then we say that $\mathbb{P}$ is Markov to G
- In this case, G represents a class of distributions

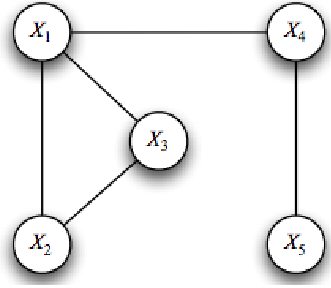
Example: The graph $X_1 \cdots X_2$ has $I(G) = \emptyset$. All bivariate distributions are in $\mathcal{P}(G)$, including $p(X_1, X_2) = p(X_1)p(X_2)$

NONPARAMETRIC STATISTICAL GRAPHICAL MODELS

Undirected (Markov) graphical models allow a decomposition into clique potentials

A clique is a fully connected subgraph

A maximal clique is such that it is not contained in any larger clique



Nonparametric statistical graphical models

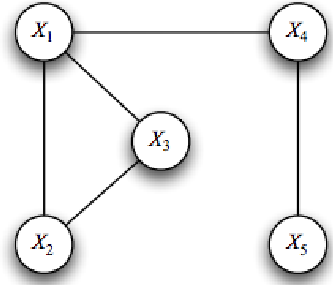
Let $\mathcal{C}$ be the set of all maximal cliques in a graph

Hammersley and Clifford: A Markov and "nice" measure $\mathbb{P}$ can be factored multiplicatively as

$$p(X_1, \ldots, X_p) = \prod_{C \in \mathcal{C}} \psi_C(X_C)$$

The ψ_C are known as clique potentials

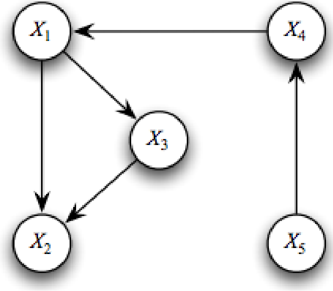
NONPARAMETRIC STATISTICAL GRAPHICAL MODELS



The family of distributions represented by this graph can be factored as

$$p(X_1, \ldots, X_5) = \psi_{1,2,3}(X_1, X_2, X_3)\psi_{1,4}(X_1, X_4)\psi_{4,5}(X_4, X_5)$$

DIRECTED GRAPHICAL MODELS



The family of distributions represented by this graph can be factored as

$$p(X_1,\ldots,X_5) = p(X_5)p(X_4|X_5)p(X_1|X_4)p(X_3|X_1)p(X_2|X_1,X_3)$$

Parametric statistical graphical models

Goal: Given a sample $x_1, \ldots, x_n \sim \mathbb{P}$, we wish to estimate (or less ambitiously, constrain) the graph G

Using properties of Gaussian distributions, we know that

$$X_j \perp X_k | \text{rest} \Leftrightarrow \Omega_{jk} = 0$$

where $\Omega = \Sigma^{-1}$ is the precision matrix

Parametric statistical graphical models

Suppose we are in low dimensions

(That is, $n >> p$)

We can use the usual MLE to find $\widehat{\Omega}$

$$\begin{aligned}\log p(x_1, \ldots, x_n|\Omega) &\propto \log\left(|\Omega|^{n/2} e^{-\frac{1}{2}\sum_{i=1}^{n}(x_i - \widehat{\mu})^\top \Omega (x_i - \widehat{\mu})}\right) \\ &\propto \frac{1}{2}\left(\log|\Omega| - n\mathrm{trace}(\Omega S)\right)\end{aligned}$$

where S is the sample covariance (Here, I've maximized over μ)

This gives $S^{-1} = \widehat{\Omega}$ and we can test where $\Omega_{jk} = 0$

Parametric statistical graphical models

As usual, use the MLE at your own risk, especially if n isn't extremely large relative to p

Example: Suppose we collect S& P 500 data from January 1, 2003 to January 1, 2008

(This will only be 452 stocks, as we'll only take the intersection of the listing over time)

This gives us $X_j \in \mathbb{R}^{1258}$ for $j = 1, \ldots, 452$

(X_{jt} is the price of j^{th} stock on t^{th} day)

Of course, these are autocorrelated, hence we report[1]

$$X_{jt} = \log(X_{jt}/X_{j,t-1})$$

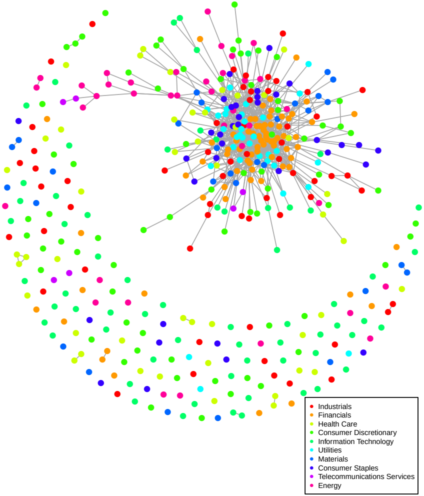
[1] Plus some outlier truncation

Parametric statistical graphical models

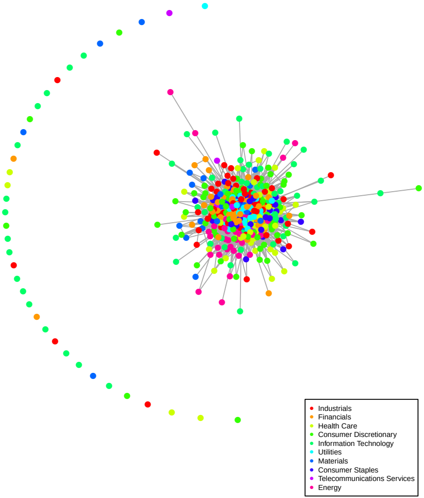
After estimating Ω, we can plot the resulting graph for various thresholds on the size of the entry in Ω

This gives us an idea of some strength of conditional independence relations

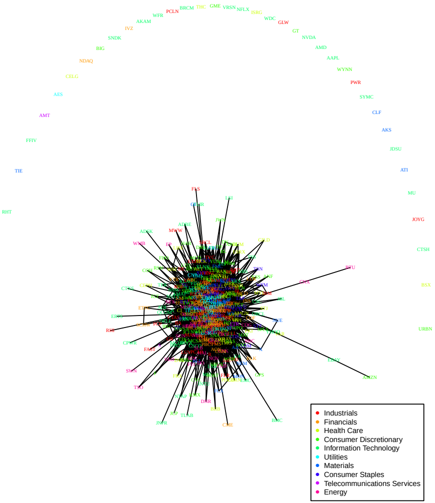
Parametric statistical graphical models



Parametric statistical graphical models



Parametric statistical graphical models



Parametric statistical graphical models

There is an R package for doing this a bit more formally: SIN

The etymology is from terminology rampant in the field of graphical models

- Faithfulness: This occurs when $I(\mathbb{P}) = I(G)$
- Moral graph: The undirected version of a DAG that has the 'same' independence relations

Hence terms related to morality persist in the field

Parametric statistical graphical models

SIN is a pseudo-acronym for partitioning the vertices into a(n)

- significant set S
- indeterminate set I
- non-significant N

This can be thought of as a way for controlling the overall error rate for incorrect edge inclusion

SIN output two graphs:

- A graph whose edges are in $S \cup I$
- A graph whose edges are in S

SIN

SIN is comprised of testing the partial correlation of each pair of covariates, given the others

In previous work, this testing was done in a backwards stepwise fashion

The largest p-value is determined and the edge is removed from the graph

(The null-hypothesis is that the correlation coefficient is 0)

This approach has some obvious flaws

(A clear mis-use of p-values, no control of familiy-wise error rate)

SIN

In the SINful approach, they do the following (details omitted)

1. Identify that the sample covariance approximately follows a Wishart distribution
2. Using the delta method + a z-transformation, we get an asymptotic normal for the sample partial correlations
3. Use a Gaussian concentration result to get family-wise p-values
4. These p-values get partitioned into S, I, and N

   (Perhaps using $S = (0, 0.05]$, $I = (0.05, .25]$, and $N = (.25, 1]$. Most common is to visualize the p-values and subjectively bin them)

(See Drton, Perlman (2004) for details)

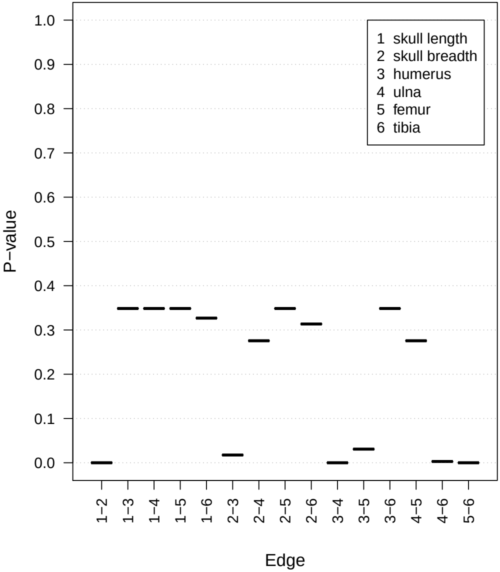
SIN

(The stock data doesn't work well with SIN (too high dimesional). Example from help file instead)

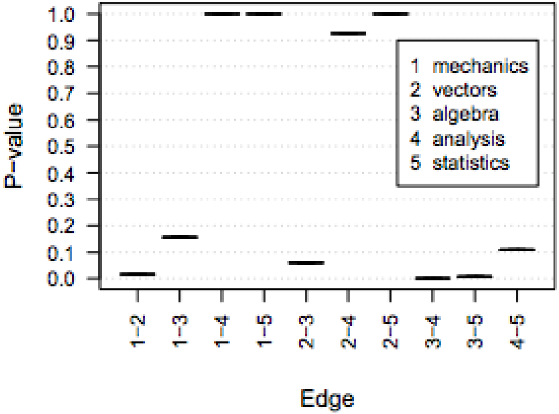
```
data(fowlbones)
pvals <- holm(sinUG(fowlbones$corr,fowlbones$n))
plotUGpvalues(pvals)
```

Note: the holm function implements a technique from a paper in 1979 that 'improves p-values while still allowing valid simultaneous testing'

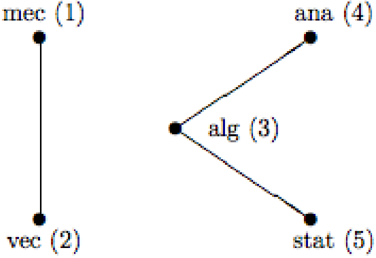
Parametric statistical graphical models



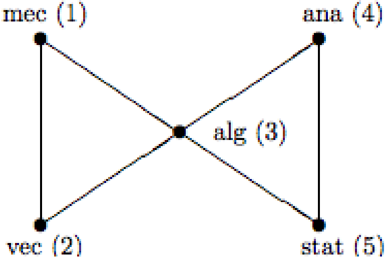
Parametric statistical graphical models



Parametric statistical graphical models



S



S, I

REGULARIZED STATISTICAL GRAPHICAL MODELS

There are two common methods for estimation when $p > n$

- parallel lasso

  (Meinshausen and Buhlmann (2006))

- Graphical lasso

  (Banerjee et al.(2008) or Hastie et al.)

Parallel lasso

This is conceptually quite simple

1. For each $j = 1, \ldots, p$, regress X_i on all other variables using lasso
2. Put an edge between X_i and X_j if each appears in the active set of the other variable

Graphical lasso

This approach takes the usual likelihood

$$\begin{aligned}\log p(x_1,\ldots,x_n|\Omega) &\propto \log\left(|\Omega|^{n/2} e^{-\frac{1}{2}\sum_{i=1}^{n}(x_i-\widehat{\mu})^\top\Omega(x_i-\widehat{\mu})}\right)\\ &\propto \frac{1}{2}\left(\log|\Omega| - n\mathrm{trace}(\Omega S)\right)\end{aligned}$$

and penalizes it

$$\min -\frac{1}{2}\left(\log|\Omega| - n\mathrm{trace}(\Omega S)\right) + \lambda\,||\Omega||_1$$

($||\cdot||_1$ is the matrix functional given by the sum of the absolute values of the entries)

Regularized statistical graphical models in R

Both can be accomplished with the huge package

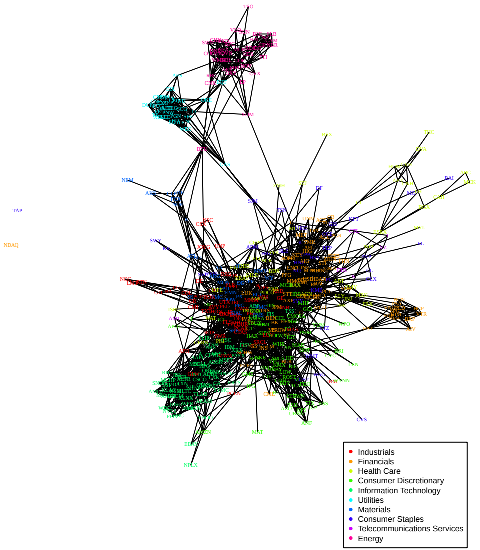
- parallel lasso

  ```
  out.parallel  = huge(cov.hat,method = "mb")
  ```

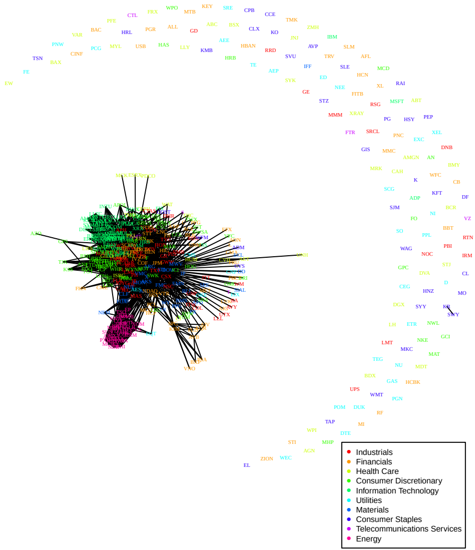
- Graphical lasso

  ```
  out.glasso    = huge(cov.hat,method = "glasso")
  ```

Parallel lasso



Graphical lasso



Up next:
Conclusion