

CMSC 726

Lecture 18: Spectral Clustering

Lise Getoor
November 4, 2010

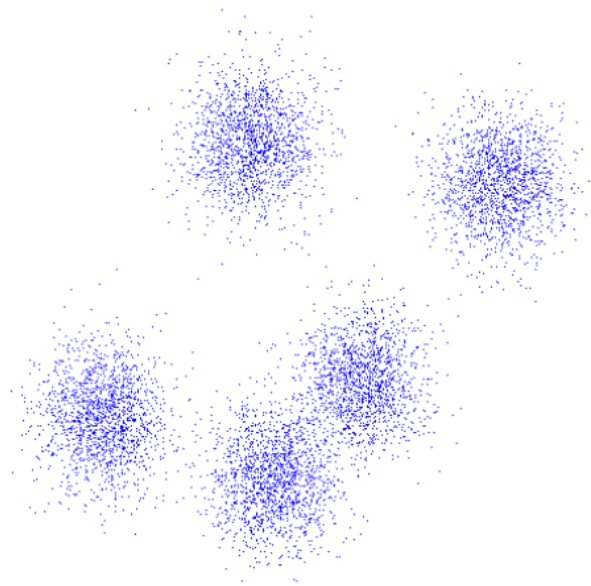
ACKNOWLEDGEMENTS: The material in this course is a synthesis of materials from many sources, including: Hal Daume III, Mark Drezde, Carlos Guestrin, **Dan Klein**, Andrew Ng, Ben Taskar, **Eric Xing**, and others. I am very grateful for their generous sharing of insights and materials.

Families of Clustering Algorithms

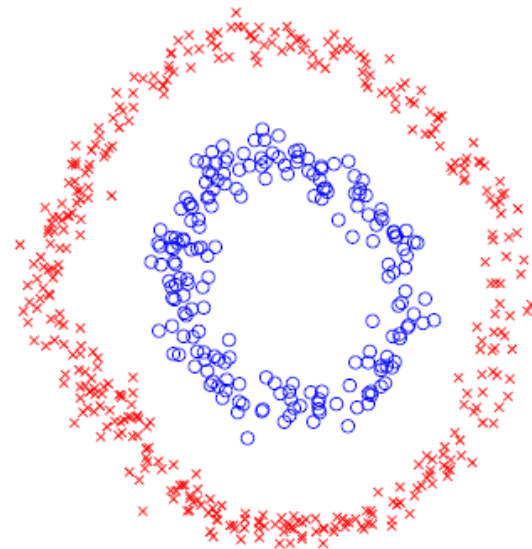
- ▶ Partition-based methods
 - e.g., K-means
- ▶ Hierarchical clustering
 - e.g., hierarchical agglomerative clustering
- Probabilistic model-based clustering
 - e.g., mixture models, Gaussian Mixture Models
 - expectation maximization
- Spectral Clustering

Data Clustering

- ▶ Two different criteria
 - Compactness, e.g., k-means, mixture models
 - Connectivity, e.g., spectral clustering



Compactness



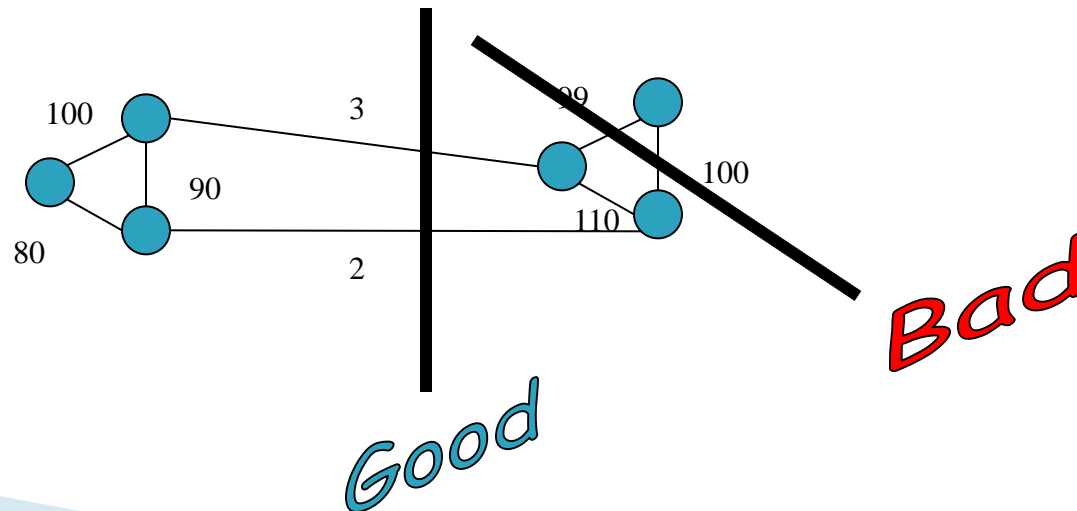
Connectivity

Spectral Clustering Algorithms

- ▶ Refers to general problems of partitioning rows of a matrix according to components in top few matrix singular vectors
- ▶ The problem of clustering rows of a matrix is ubiquitous. Examples include:
 - The matrix encodes the pairwise similarities of vertices of a graph.
 - The rows of the matrix are points in a d -dimensional Euclidean space. The columns are the coordinates.
 - The rows of the matrix are documents of a corpus. The columns are the terms. The (i,j) entry encodes information about the occurrence of the j th term in the i th document.
 - The rows and columns of the matrix represent web pages and entry (i,j) indicates whether site i has a link to site j
 - The columns refer to individuals, rows refer to products and the (i,j) entry indicates something about how much individual j likes product i .
 - The rows refer to experiments and the columns refer to genes and entry (i,j) indicates the expression level of gene j in experiment i .

Graph-Theoretic Clustering

- ▶ Weighted graph
- ▶ Edge weights correspond to similarity
- ▶ Cut edges in the graph to form a good set of connected components—ideally the within component edges in the graph have large weights and the across component edges have small weights



Application: Image Segmentation

- ▶ The weights in the graph are called affinity measures.
- ▶ Affinity measure depends on problem at hand
- ▶ **affinity by distance:** affinity should go down quite sharply with the distance once the distance is over some threshold. One appropriate expression has the form:

$$\text{aff}(x, y) = \exp\left[-(\text{dist}(x, y) / 2\sigma_d^2)\right]$$

- ▶ where σ_d is a parameter that is large if quite distance points should be grouped and small if only nearby pointed should be grouped.

Application: Image Segmentation

- ▶ **affinity by intensity:** affinity large for similar intensities and smaller as the difference increases

$$aff(x, y) = \exp\left[-((I(x) - I(y))^t (I(x) - I(y)) / 2\sigma_d^2)\right]$$

- ▶ **affinity by color:** affinity large for similar colors

$$aff(x, y) = \exp\left[-((C(x) - C(y))^t (C(x) - C(y)) / 2\sigma_c^2)\right]$$

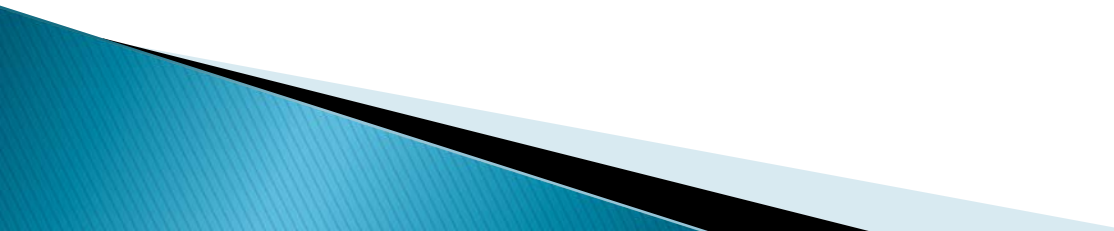
- ▶ **affinity by texture:**

$$aff(x, y) = \exp\left[-((F(x) - F(y))^t (F(x) - F(y)) / 2\sigma_t^2)\right]$$

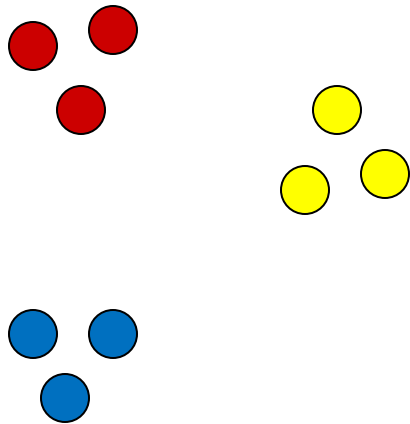
Application: Information Retrieval

- ▶ Document collection represented in vector space model
 - rows correspond to words
 - columns correspond to documents
- ▶ Latent Semantic Indexing: clusters words by co-occurrence
 - projects documents and queries into a space with 'latent' semantic dimensions
 - co-occurring terms are project onto the same dimensions, non-co-occurring terms are projected onto different dimensions
 - in latent semantic space, a query and a document can have high cosine similarity even if they do not share terms – as long as their terms are semantically similar according to the co-occurrence analysis.
 - It overcomes both synonymy (car vs. automobile) and polysemy (WWW spider vs. eight-legged spider).

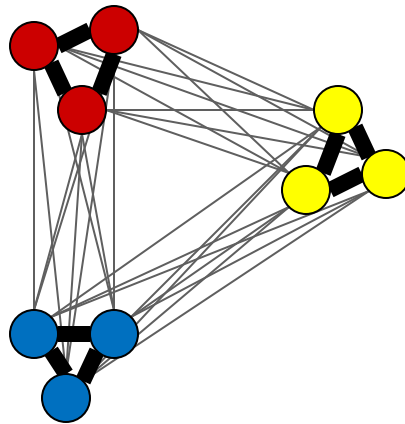
Application: Collaborative Filtering

- ▶ aka Recommender Systems
 - ▶ Paul Resnick: "Guiding people's choices of what to read, what to look at, what to watch, what to listen to (the filtering part); and doing that guidance based on information gathered from some other people (the collaborative part)."
 - ▶ movies, book, restaurant, music, web pages
 - ▶ cluster rows
 - ▶ fill in missing values, based on similar individuals
- 

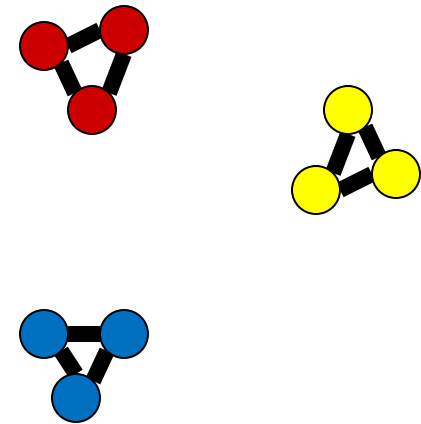
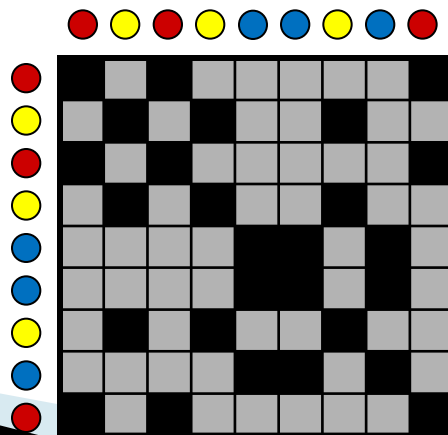
Spectral Clustering Overview



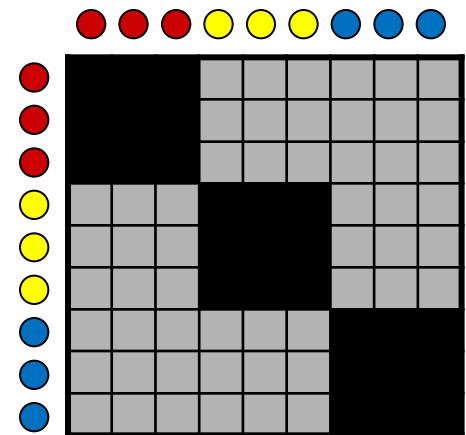
Data



Similarities



Block-Detection



aside: Linear Algebra Refresher

- ▶ **Eigen decomposition/matrix diagonalization** of a square $k \times k$ matrix A into eigenvalues and eigenvectors.
Suppose A has nondegenerate eigenvalues $\lambda_1 \lambda_2 \dots \lambda_k$ and corresponding linearly independent eigenvectors $X_1 X_2 \dots X_k$. Let $D = \text{diag}(\lambda_1 \lambda_2 \dots \lambda_k)$ and $P = [X_1 X_2 \dots X_k]$ then

$$A = P D P^{-1}$$

- ▶ **Singular Value Decomposition (SVD)**: decomposition of $m \times n$ matrix A

$$A = U D V^T$$

where U and V are orthogonal $m \times m$ and $n \times n$ matrices and D is a diagonal matrix whose diagonal entries σ_i are called the singular values.

Spectral Graph Analysis

- ▶ http://monod.biomath.nyu.edu/rna/tutorials/spectral_analysis.html
- ▶ <http://mathworld.wolfram.com/EigenDecomposition.html>

Eigenvectors and Blocks

- ▶ Block matrices have block eigenvectors:

1	1	0	0
1	1	0	0
0	0	1	1
0	0	1	1

eigensolver

$\lambda_1 = 2$

.71
.71
0
0

$\lambda_2 = 2$

0
0
.71
.71

$\lambda_3 = 0$

$\lambda_4 = 0$

- ▶ Near-block matrices have near-block eigenvectors: [Ng *et al.*, NIPS 02]

1	1	.2	0
1	1	0	-.2
.2	0	1	1
.0	-.2	1	1

eigensolver

$\lambda_1 = 2.02$

.71
.69
.14
0

$\lambda_2 = 2.02$

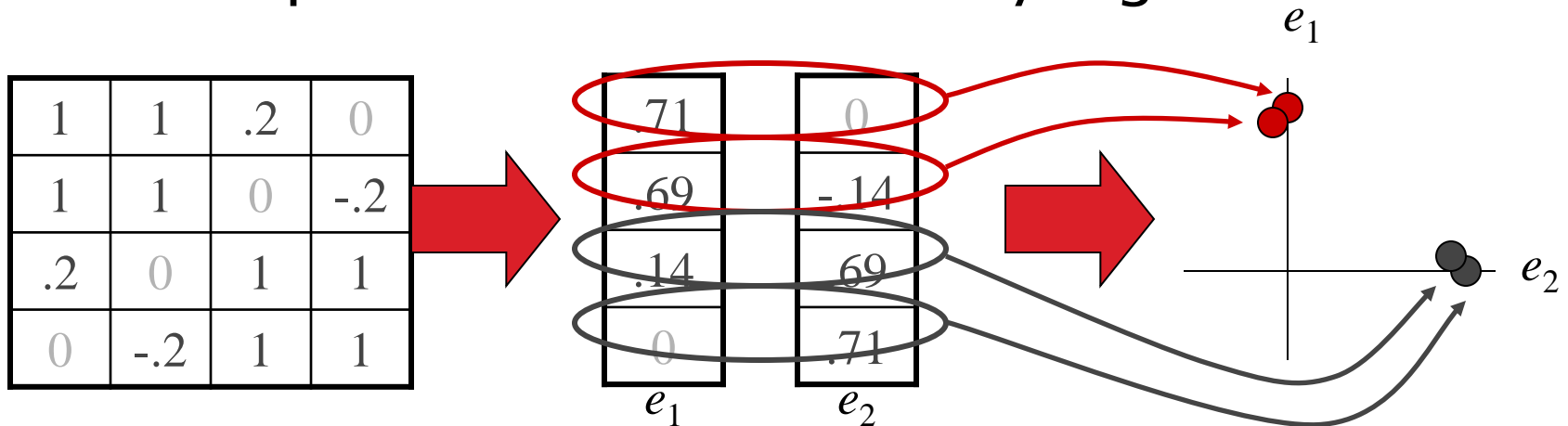
0
-.14
.69
.71

$\lambda_3 = -0.02$

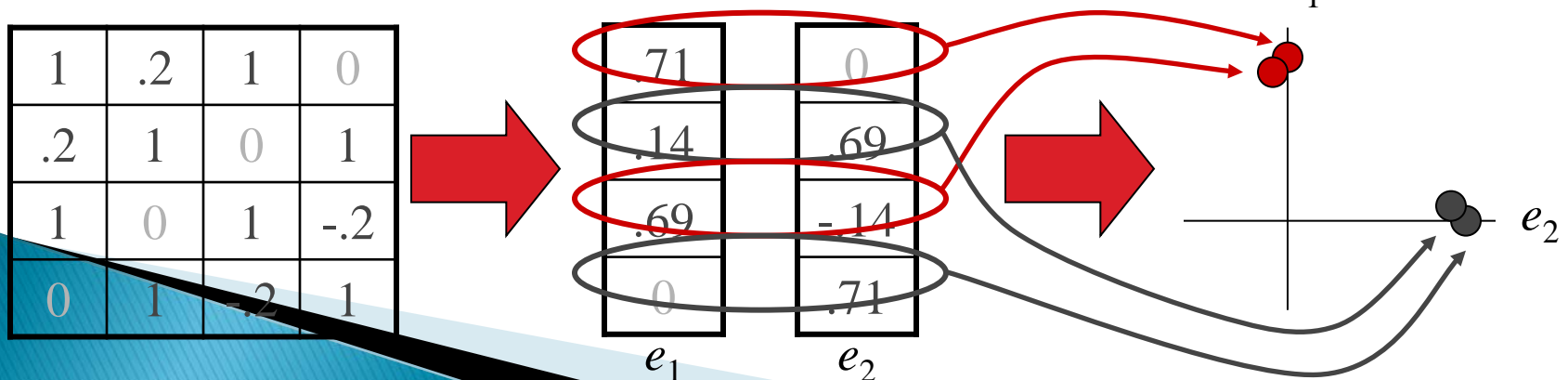
$\lambda_4 = -0.02$

Spectral Space

- ▶ Can put items into blocks by eigenvectors:



- ▶ Clusters clear regardless of row ordering:



Spectral Algorithm (Unnormalized)

- ▶ Algorithm:

Find the top k right singular vectors v_1, \dots, v_k

Let C be the matrix whose j th column is given by $A v_j$

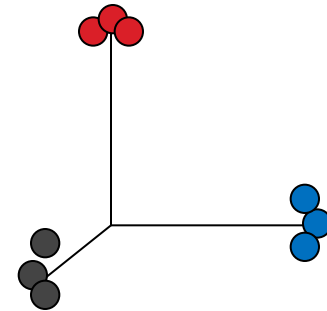
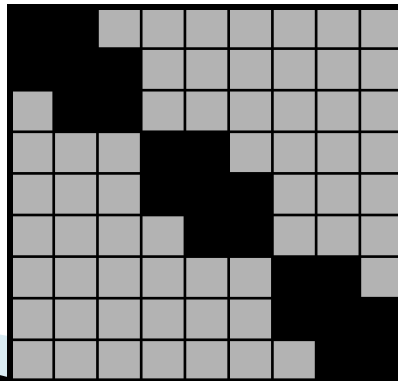
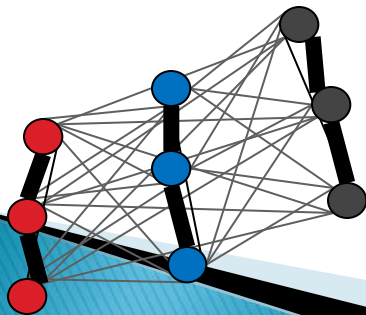
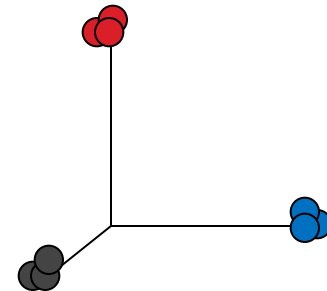
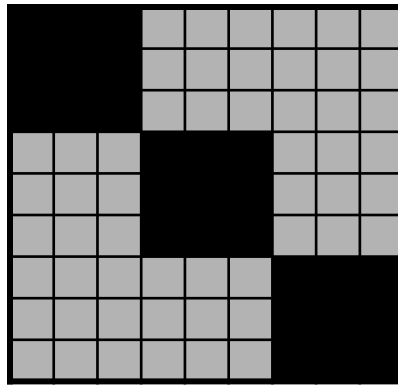
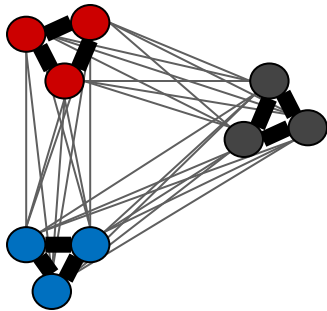
Place row i in cluster j if C_{ij} is largest entry in the i th row of C

- ▶ Interpretation:

Suppose the rows of A are points in a high-dimensional space. Then the subspace defined by the top k right singular vectors of A is the rank- k subspace that best approximates A . The spectral algorithm projects all the points onto this subspace. Each singular vector then defines a cluster; to obtain a clustering we map each project point to the cluster of the singular vector that is closest to it in angle.

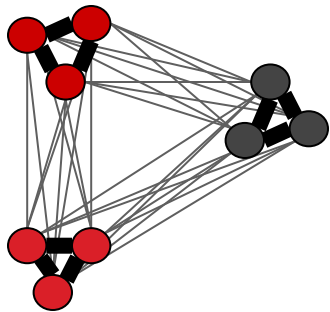
The Spectral Advantage

- ▶ The key advantage of spectral clustering is the spectral space representation:

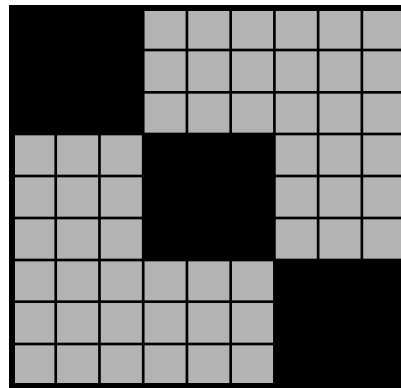


Spectral Clustering Example

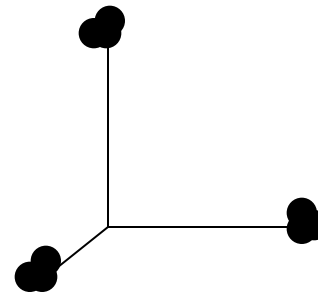
Data



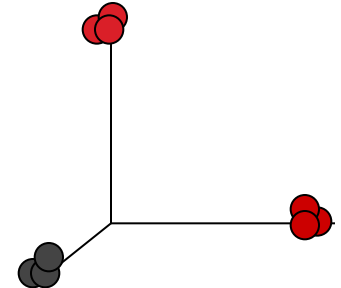
Similarities



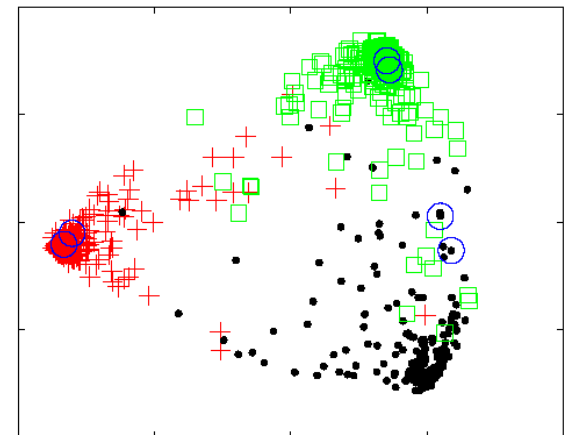
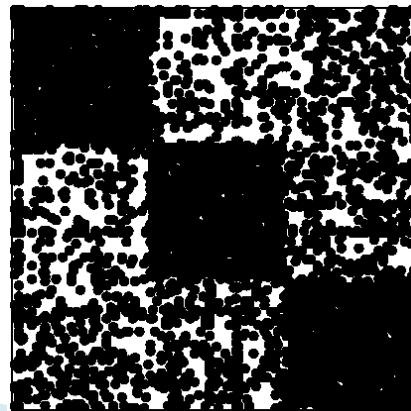
Spectral Space



Cluster



3 Sets of
News-
group
Postings



Approach #2: Normalized Cuts (Shi and Malik)

- ▶ Rather than looking at top k eigenvectors, look at generalized eigenvectors.
- ▶ Can show that solving this eigenvalue problem is a relaxation of the min cut problem

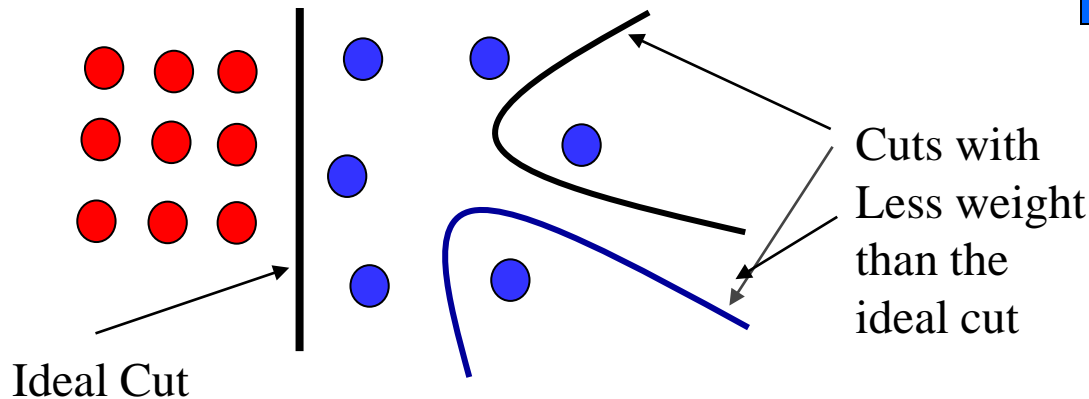
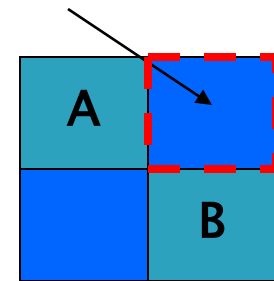
Approach #2a: Directly optimize Minimum cut

- Criterion for partition:

Problem!

Weight of cut is directly proportional to the number of edges in the cut.

$$\min cut(A, B) = \min_{A, B} \sum_{i \in A, j \in B} W_{i, j}$$



Normalized Cuts

- ▶ Cut the graph into two connected components such that

$$\text{ncut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)}$$

where $\text{cut}(A, B)$ is the sum of weights of all edges in V that have one end in A and the other in B , and $\text{assoc}(A, V)$ is the sum of weights of all edges that have one end in A

- ▶ This score is small if the cut separates two components that have few edges of low weight between them and many internal edges of high weight
- ▶ We would like to find the cut with minimum value, called a **normalized cut**

Formulation as IP

- ▶ y : vector, for each node, 1 or -1 ; all the nodes with value 1 are in one component, all the values with -1 are in the other component
- ▶ Let D be the degree matrix of A :
 $D(i,i) = \sum_j A(i,j)$
- ▶ Then our criteria can be rewritten as:

$$\frac{y^t (D - A) y}{y^t D y}$$

and now we wish to find a vector y that minimizes this criterion.

- ▶ This is an integer programming problem (IP) and solving it is NP-complete

Relaxation of Problem

- ▶ Approximate solution by finding a real vector y , rather than an integer solution.
- ▶ Define the generalized eigenvector y_i as solution to:

$$(D-A)y_i = \lambda_i D y_i$$

and define the second generalized eigenvector as the y_i corresponding to the second smallest λ_i

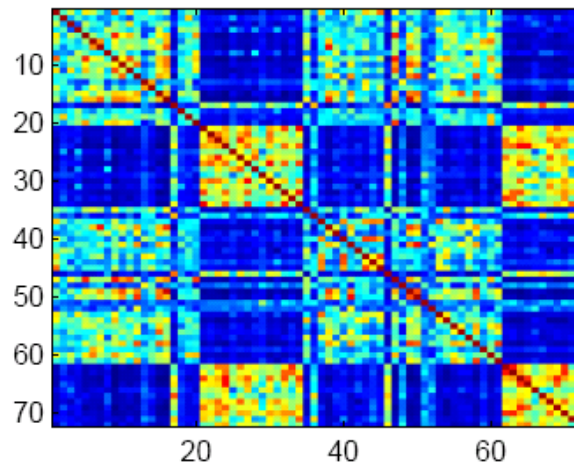
- ▶ Find a threshold to split into components

Algorithm

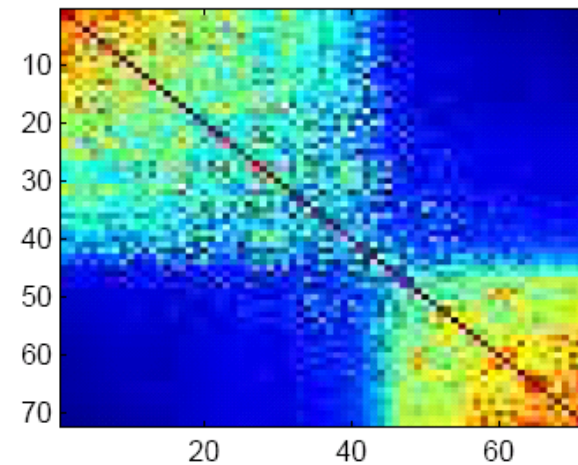
1. Define a similarity function between 2 nodes. i.e.:
$$w_{i,j} = e^{\frac{-\|X_{(i)} - X_{(j)}\|_2^2}{\sigma_x^2}}$$
2. Compute affinity matrix (W) and degree matrix (D).
3. Solve $(D - W)y = \lambda Dy$
 - Do singular value decomposition (SVD) of the graph Laplacian $L = D - W$
$$L = V^T \Lambda V$$
4. Use the eigenvector with the second smallest eigenvalue, y^* , to bipartition the graph.
 - For each threshold k ,
 - $A_k = \{i \mid y_i \text{ among } k \text{ largest element of } y^*\}$
 - $B_k = \{i \mid y_i \text{ among } n-k \text{ smallest element of } y^*\}$
 - Compute $\text{Ncut}(A_k, B_k)$
 - Output
$$k^* = \arg \max \text{Ncut}(A_k, B_k) \quad \text{and} \quad A_{k^*}, B_{k^*}$$

Example

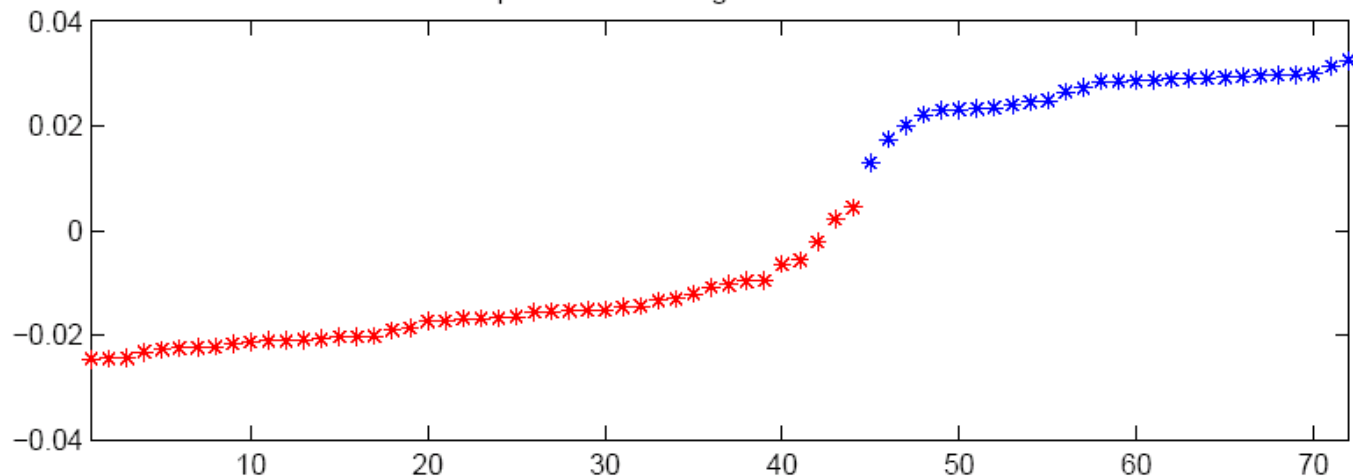
input affinity matrix



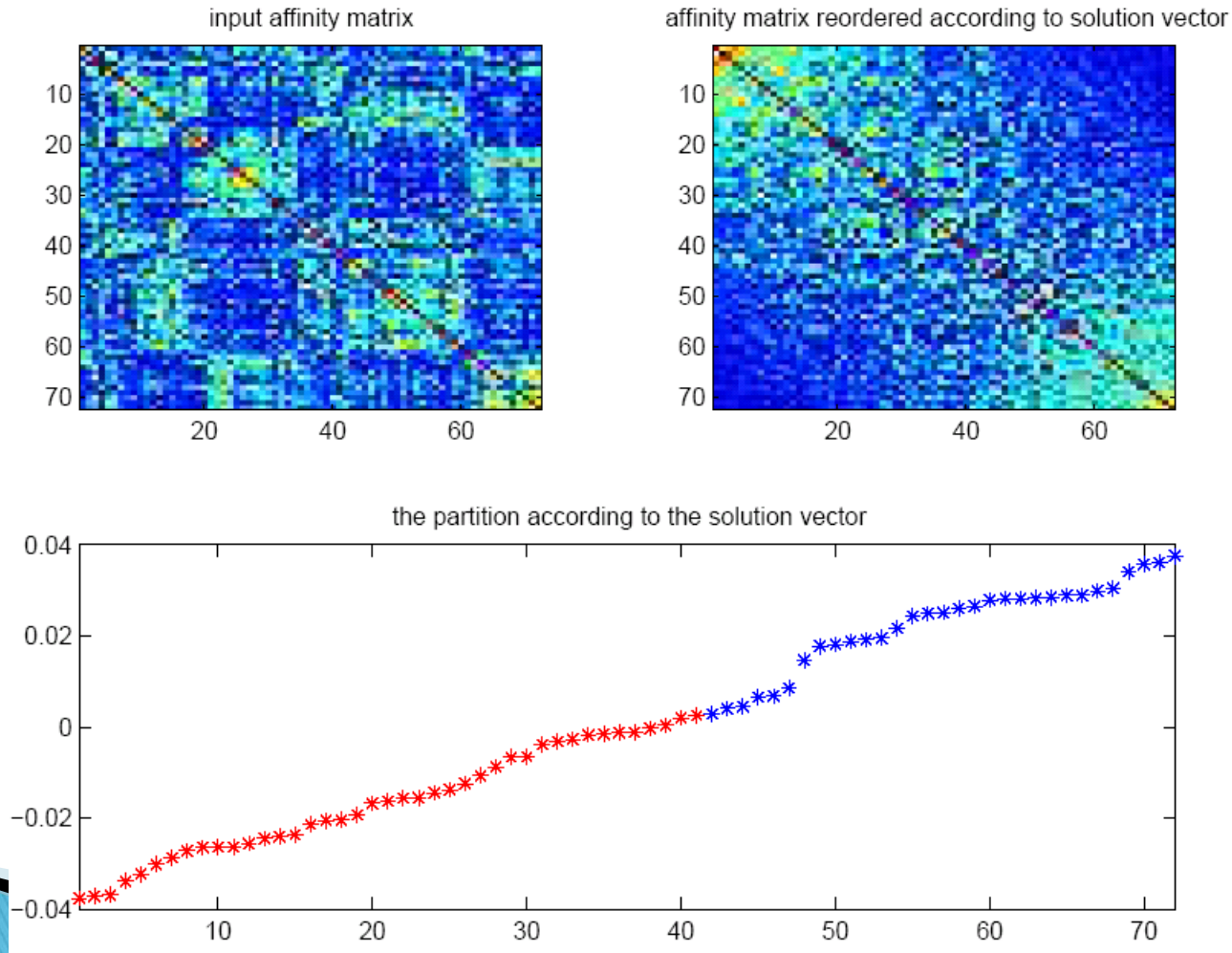
affinity matrix reordered according to solution vector



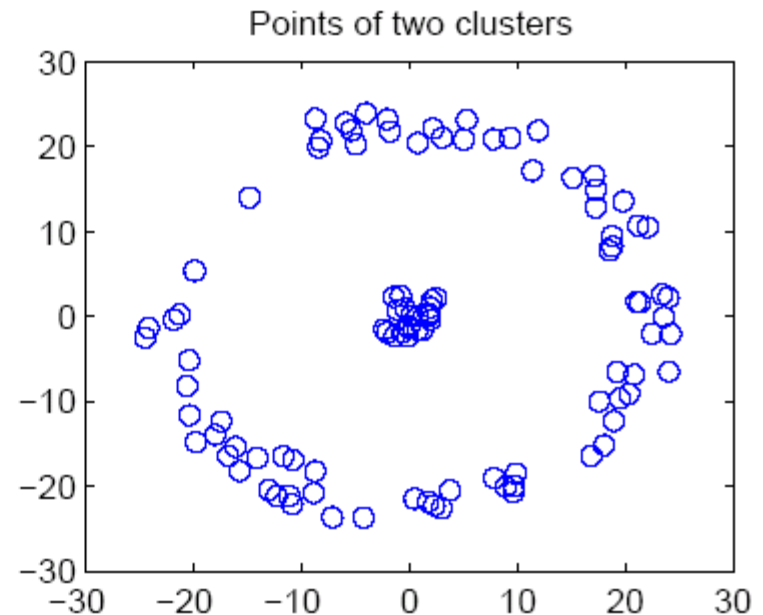
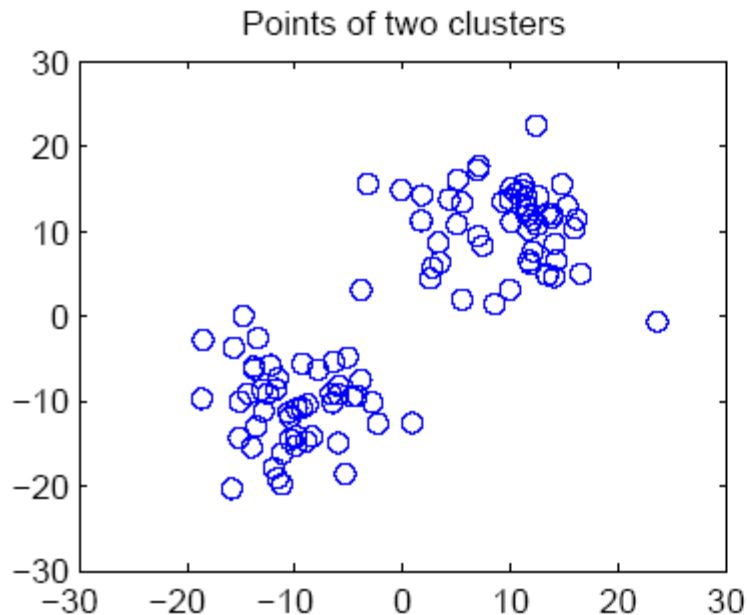
the partition according to the solution vector



Poor features can lead to poor outcome (Xing et al 2002)

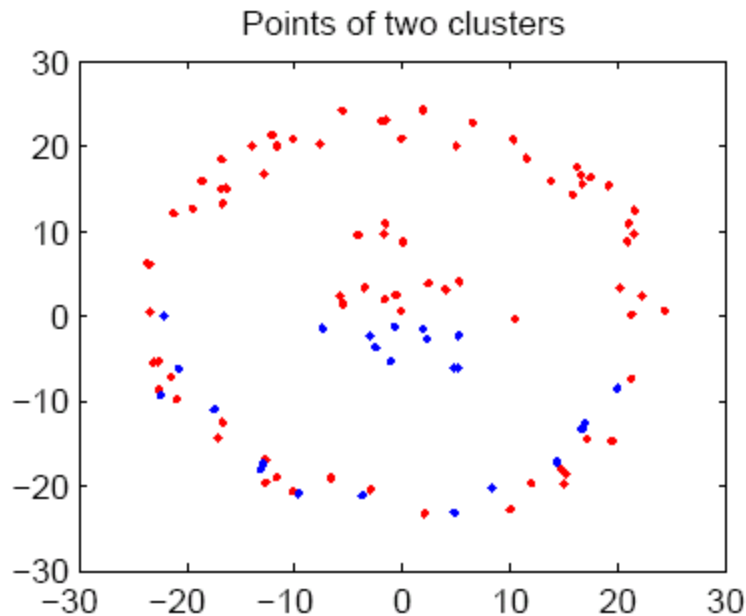


Superior performance?

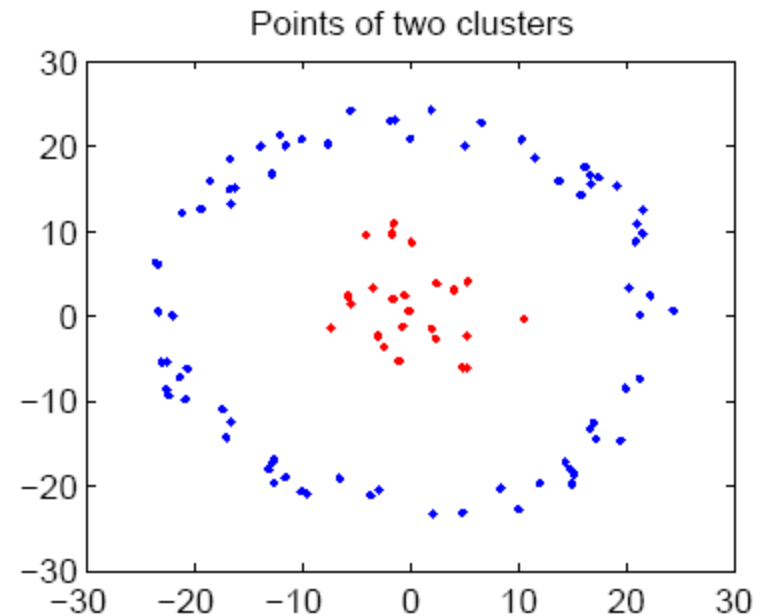


- ▶ K-means and Gaussian mixture methods are biased toward convex clusters

Ncut is superior in certain cases



K-means



Ncut

Spectral Clustering

- ▶ Algorithms that cluster points using eigenvectors of matrices derived from the data
- ▶ Obtain data representation in the low-dimensional space that can be easily clustered
- ▶ Variety of methods that use the eigenvectors differently (we have seen an example)
- ▶ Empirically very successful
- ▶ Authors disagree:
 - Which eigenvectors to use
 - How to derive clusters from these eigenvectors
- ▶ Two general methods

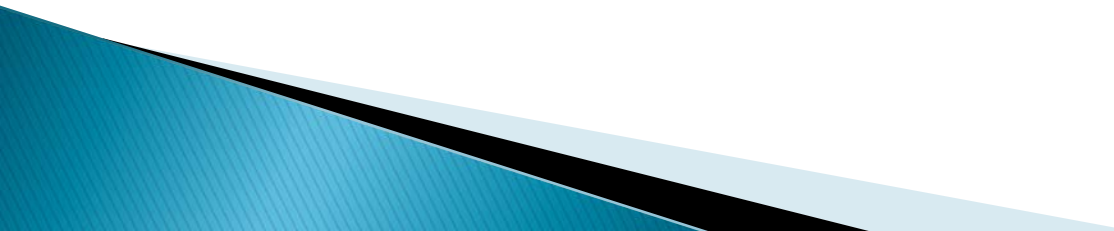
Method #1

- ▶ Use k eigenvectors (k chosen by user)
- ▶ Directly compute k -way partitioning
- ▶ Experimentally has been seen to be “better”

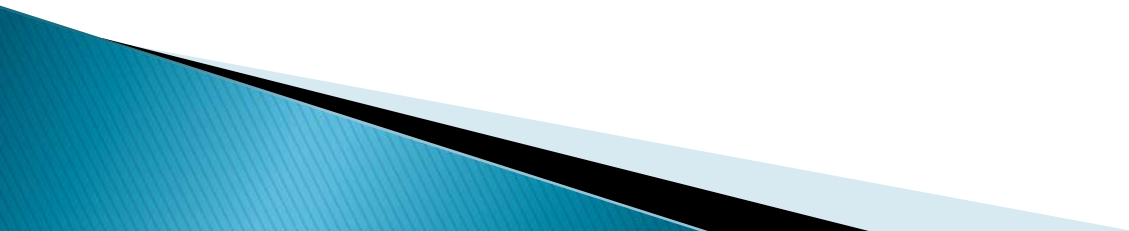
Method #2

- ▶ Partition using only one eigenvector at a time
- ▶ Use procedure recursively
- ▶ Example: Image Segmentation
 - Uses 2nd (smallest) eigenvector to define optimal cut
 - Recursively generates two clusters with each cut

Summary: Families of Clustering Algorithms

- ▶ Partition-based methods
 - e.g., K-means
 - ▶ Hierarchical clustering
 - e.g., hierarchical agglomerative clustering
 - ▶ Probabilistic model-based clustering
 - e.g., mixture models, Gaussian Mixture Models
 - expectation maximization
 - ▶ Spectral Clustering
- 

Other flavors of clustering algorithms...



Co-Clustering

- ▶ aka Two-sided clustering
- ▶ Many applications!
- ▶ Can extend spectral graph partitioning to bipartite graph partitioning

Co-clustering documents and Words

- ▶ Dhillon, KDD 2001
- ▶ Most existing work does one-way clustering, either clustering document or word cluster
 - documents clustered based upon their word distributions
 - word clustered based on their co-occurrence in documents
- ▶ Duality of word & document clustering: word clustering induces document clustering while document clustering induces word clustering

Co-clustering Gene array data

- ▶ Duality of gene & experiment clustering: gene clustering induces experiment clustering while experiment clustering induces gene clustering

Co-clustering for Collaborative Filtering

- ▶ Many existing approaches do single-sided clustering:
 - Cluster individuals
 - Cluster products
- ▶ Again, we have duality of individual & product clustering

Multirelational Clustering

- ▶ Clustering collections of objects of different types
 - Documents, document authors, companies
- ▶ Active research area

Next Time....

- ▶ Graphical Models,
- ▶ Reading: Bishop, ch. 8