Clustering

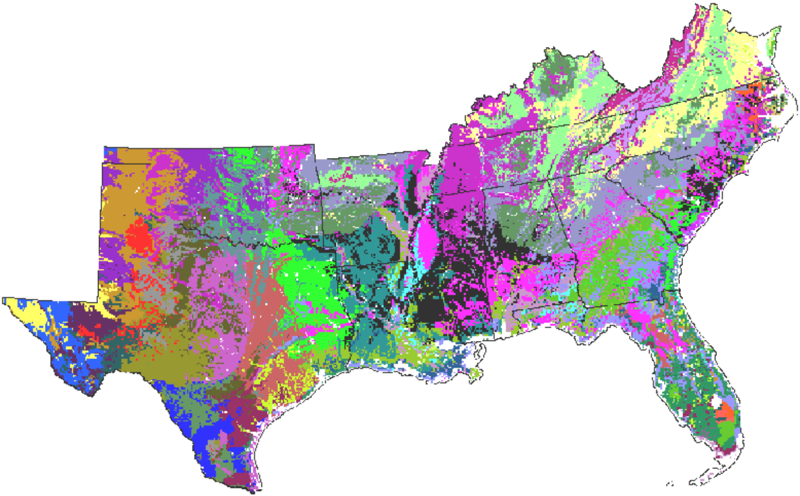
Jiayu Zhou

[1] Department of Computer Science and Engineering
Michigan State University
East Lansing, MI USA

Table of contents

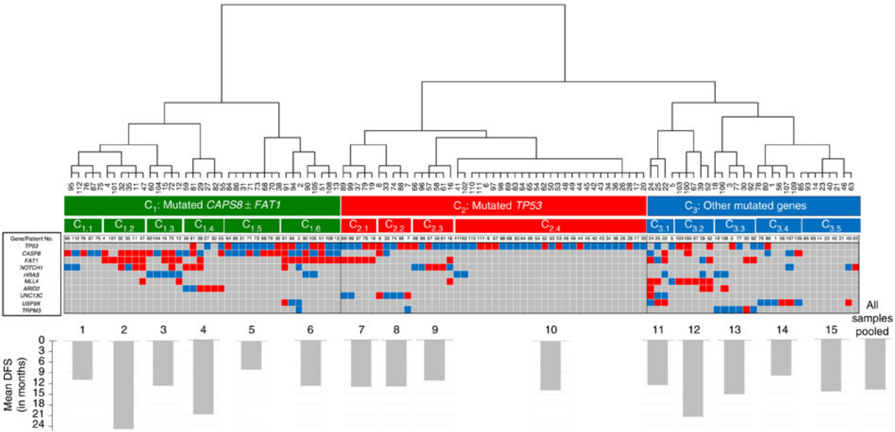
Clustering Application - Geo

13 States Clustered into 51 Custom Ecoregions.



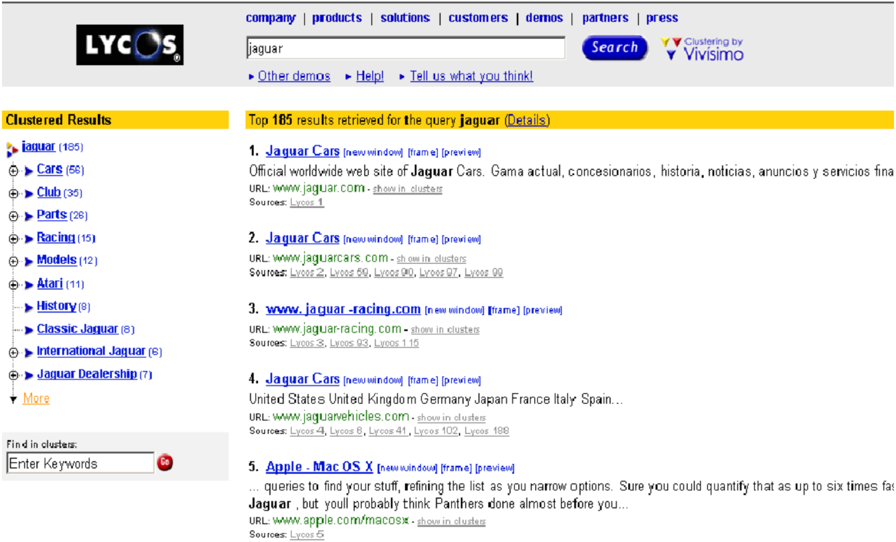
Clustering Application - Cancer Patients

Clustering of gingivo-buccal oral cancer patients based on mutational profiles.

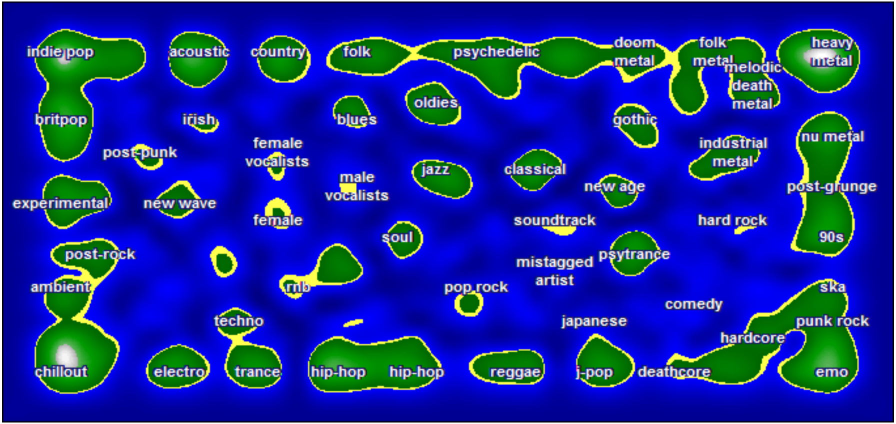


Mutational landscape of gingivo-buccal oral squamous cell carcinoma reveals new recurrently-mutated genes and molecular subgroups, Nature Communications, 2013

Clustering Application - Search Result Clustering

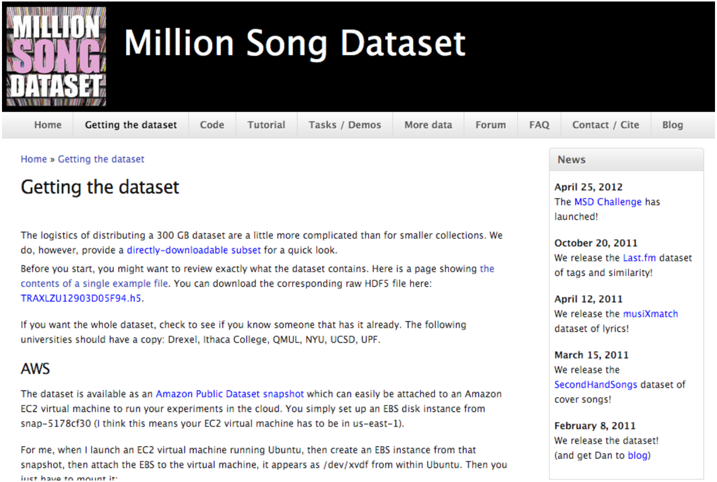


Clustering Application - Island of Music



Pampalk, Elias, Andreas Rauber, and Dieter Merkl. "Content-based organization and visualization of music archives." Proceedings of the tenth ACM international conference on Multimedia. ACM, 2002.

Million Song



Million Song Dataset

Home | Getting the dataset | Code | Tutorial | Tasks / Demos | More data | Forum | FAQ | Contact / Cite | Blog

Home » Getting the dataset

Getting the dataset

The logistics of distributing a 300 GB dataset are a little more complicated than for smaller collections. We do, however, provide a directly-downloadable subset for a quick look.

Before you start, you might want to review exactly what the dataset contains. Here is a page showing the contents of a single example file. You can download the corresponding raw HDF5 file here: TRAXLZU12903D05F94.h5.

If you want the whole dataset, check to see if you know someone that has it already. The following universities should have a copy: Drexel, Ithaca College, QMUL, NYU, UCSD, UPF.

AWS

The dataset is available as an Amazon Public Dataset snapshot which can easily be attached to an Amazon EC2 virtual machine to run your experiments in the cloud. You simply set up an EBS disk instance from snap-5178cf30 (I think this means your EC2 virtual machine has to be in us-east-1).

For me, when I launch an EC2 virtual machine running Ubuntu, then create an EBS instance from that snapshot, then attach the EBS to the virtual machine, it appears as /dev/xvdf from within Ubuntu. Then you just have to mount it:

News

April 25, 2012
The MSD Challenge has launched!

October 20, 2011
We release the Last.fm dataset of tags and similarity!

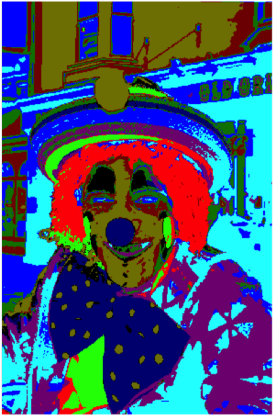
April 12, 2011
We release the musiXmatch dataset of lyrics!

March 15, 2011
We release the SecondHandSongs dataset of cover songs!

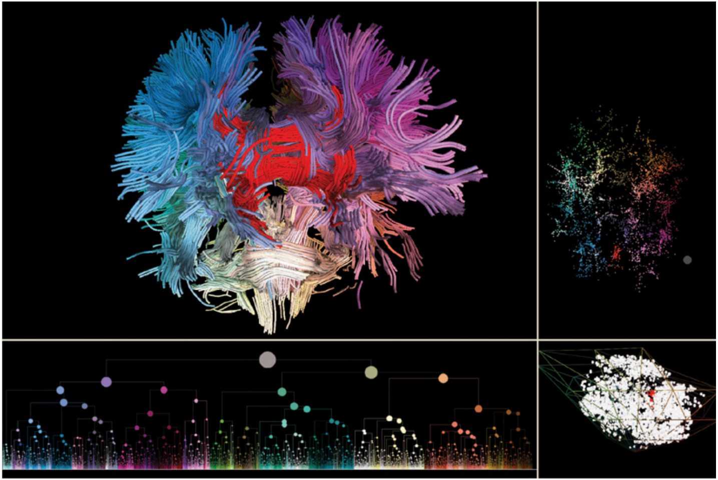
February 8, 2011
We release the dataset! (and get Dan to blog)

http://labrosa.ee.columbia.edu/millionsong/pages/getting-dataset

Clustering Application - Image Compression



Clustering Application - MRI TDI Fibers

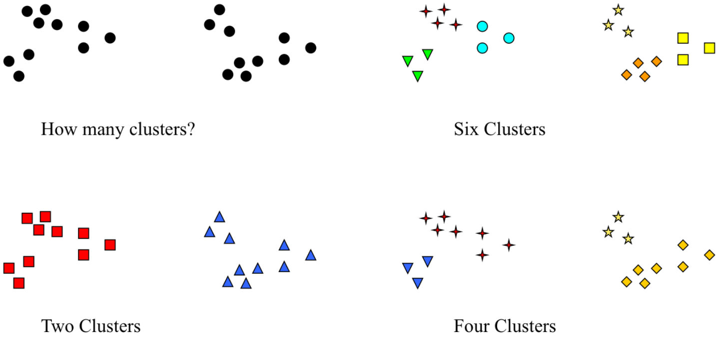


"Exploring 3D DTI fiber tracts with linked 2D representations." Visualization and Computer Graphics, IEEE Transactions on 15.6 (2009): 1449-1456.

Notion of a Cluster can be ...

How many clusters?

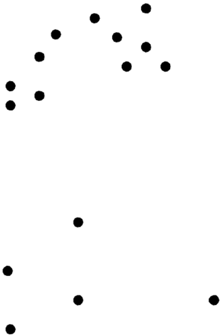
Notion of a Cluster can be Ambiguous



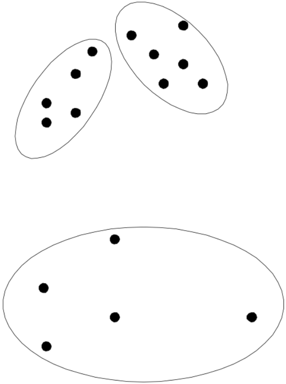
Types of Clustering

- A clustering is a set of clusters.
- Important distinction between hierarchical and partitional sets of clusters
  - **Partitional Clustering**
    A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
  - **Hierarchical clustering**
    A set of nested clusters organized as a hierarchical tree

Partitional Clustering

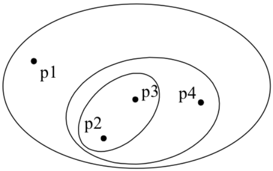


Original Points

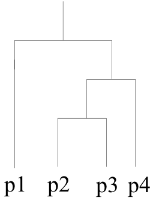


A Partitional Clustering

Hierarchical clustering



Traditional Hierarchical Clustering



Traditional Dendrogram

K-means for Clustering

K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a **centroid** (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- Optimization objective

K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a **centroid** (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- Optimization objective

$$\arg\min_{\{c_j, m_{i,j}\}} \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} \|x_i - c_j\|^2$$

where memberships $\{m_{i,j}\}$ and centers $\{c_j\}$ are correlated.

K-means Clustering Algorithm

$$\underset{\{c_j, m_{i,j}\}}{\arg\min} \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} \|x_i - c_j\|^2$$

Alternating procedure.

K-means Clustering Algorithm

$$\arg\min_{\{c_j, m_{i,j}\}} \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} \|x_i - c_j\|^2$$

Alternating procedure.

- Given centroids $\{c_j\}$, $m_{i,j} = \begin{cases} 1 & j = \arg\min_{j \in [1 \ldots K]} \|x_i - c_j\|^2 \\ 0 & \text{otherwise} \end{cases}$
- Given memberships $\{m_{i,j}\}$, $c_j = \frac{\sum_{i=1}^{n} m_{i,j} x_i}{\sum_{i=1}^{n} m_{i,j}}$

K-means Clustering Algorithm

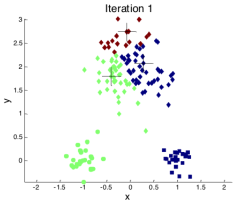
$$\arg\min_{\{c_j, m_{i,j}\}} \sum_{j=1}^{K} \sum_{i=1}^{n} m_{i,j} \|x_i - c_j\|^2$$

Alternating procedure.

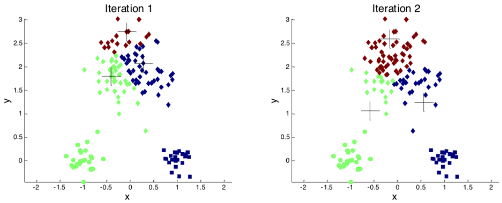
- Given centroids $\{c_j\}$, $m_{i,j} = \begin{cases} 1 & j = \arg\min_{j \in [1 \ldots K]} \|x_i - c_j\|^2 \\ 0 & \text{otherwise} \end{cases}$
- Given memberships $\{m_{i,j}\}$, $c_j = \frac{\sum_{i=1}^{n} m_{i,j} x_i}{\sum_{i=1}^{n} m_{i,j}}$

```
1: Select K points as the initial centroids.
2: repeat
3:     Form K clusters by assigning all points to the closest centroid.
4:     Recompute the centroid of each cluster.
5: until The centroids don't change
```

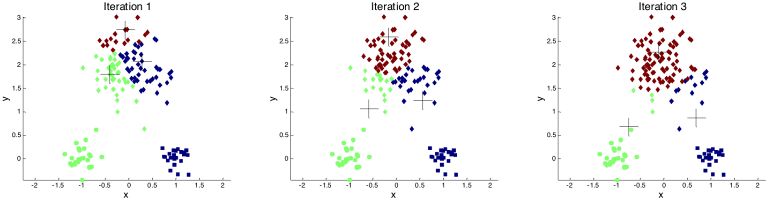
K-means illustration



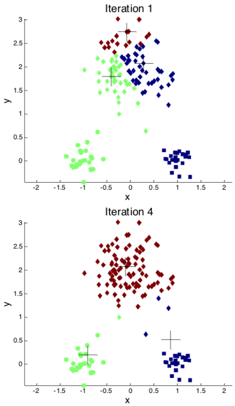
K-means illustration



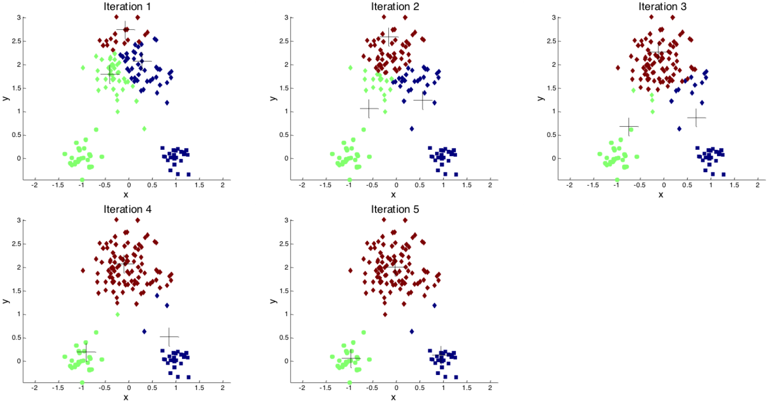
K-means illustration



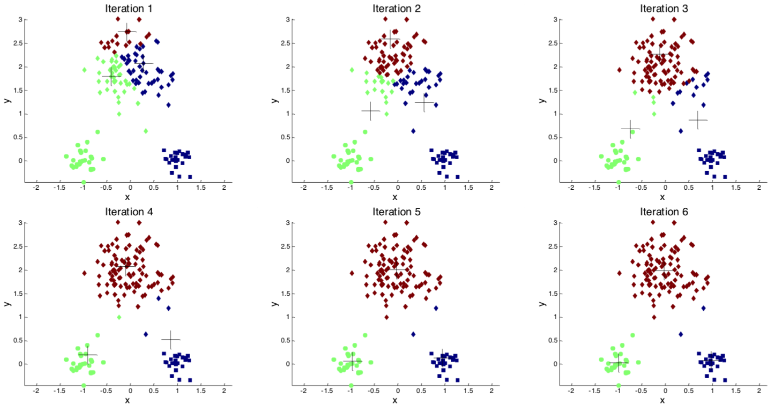
K-means illustration



K-means illustration



K-means illustration



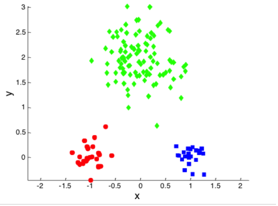
K-means Clustering Details

- Initial centroids are often chosen randomly.
- The centroid is (typically) the mean of the points in the cluster.
- Closeness is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations. Often the stopping condition is changed to "Until relatively few points change clusters"
- Let $n =$ number of points, $K =$ number of clusters, $I =$ number of iterations, $d =$ number of attributes, complexity is

K-means Clustering Details

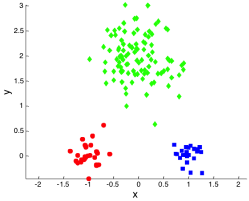
- Initial centroids are often chosen randomly.
- The centroid is (typically) the mean of the points in the cluster.
- Closeness is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations. Often the stopping condition is changed to "Until relatively few points change clusters"
- Let $n =$ number of points, $K =$ number of clusters, $I =$ number of iterations, $d =$ number of attributes, complexity is $O(n \times K \times I \times d)$

K-means revisited

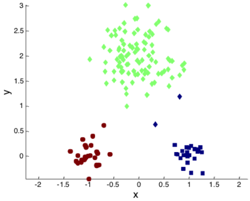


Original Points

K-means revisited

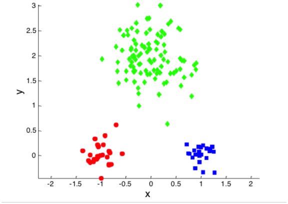


Original Points

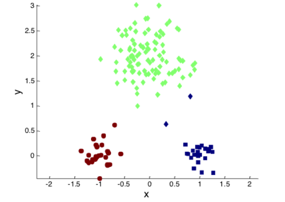


Optimal Clustering

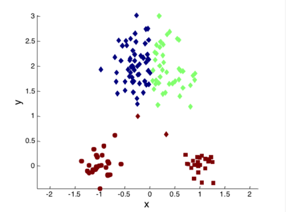
K-means revisited



Original Points



Optimal Clustering



Sub-optimal Clustering

Problems with Selecting Initial Points

If there are K "real" clusters then the chance of selecting one centroid from each cluster is small.

- Chance is relatively small when K is large
- If clusters are the same size, n, then the probability is

Problems with Selecting Initial Points

If there are K "real" clusters then the chance of selecting one centroid from each cluster is small.

- Chance is relatively small when K is large
- If clusters are the same size, n, then the probability is

$$P = \frac{\text{ways to select one centroid from each cluster}}{\text{ways to select K centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

- For example, if $K = 10$, then probability $= 10!/10^{10} = 0.00036$
- Sometimes the initial centroids will readjust themselves in "right" way, and sometimes they don't.

Solutions to Initial Centroids Problem

- Multiple runs

Solutions to Initial Centroids Problem

- Multiple runs
  Helps, but probability is not on your side

Solutions to Initial Centroids Problem

- Multiple runs
  Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids

Solutions to Initial Centroids Problem

- Multiple runs
  Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- Select more than K initial centroids and then select among these initial centroids

Solutions to Initial Centroids Problem

- Multiple runs
  Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- Select more than K initial centroids and then select among these initial centroids
- Bisecting K-means

Solutions to Initial Centroids Problem

- Multiple runs
  Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- Select more than K initial centroids and then select among these initial centroids
- Bisecting K-means
  1. Pick a cluster to split.
  2. Find 2 sub-clusters using the basic k-Means algorithm (Bisecting step)
  3. Repeat step 2, the bisecting step, for ITER times and take the split that produces the clustering with the highest overall similarity.
  4. Repeat steps 1, 2 and 3 until the desired number of clusters is reached.

  Not as susceptible to initialization issues

Evaluating K-means Clusters

Most common measure is Sum of Squared Error (SSE)

- For each point, the error is the distance to the nearest cluster
- To get SSE, we square these errors and sum them.

$$SSE = \sum_{i=1}^{K} \sum_{x \in c_i} d^2(m_i, x)$$

- x is a data point in cluster c_i and m_i is the representative point (center/mean) for cluster c_i.
- Given two clusters, we can choose the one with the smaller error
- One easy way to reduce SSE is to increase K, the number of clusters
  - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K.

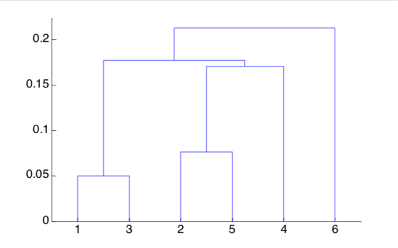
Limitations of K-means

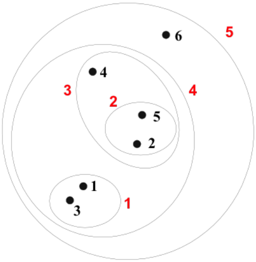
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes
- K-means has problems when the data contains outliers.
- The number of clusters (K) is difficult to determine.

Hierarchical Clustering

Hierarchical Clustering

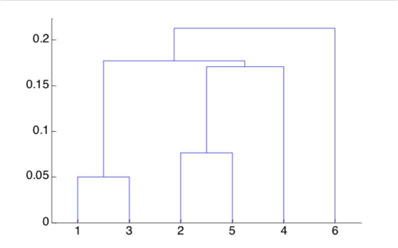
- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits

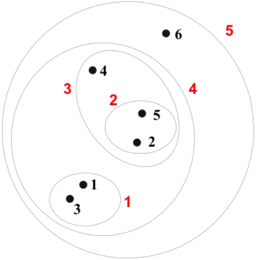




Strengths of Hierarchical Clustering

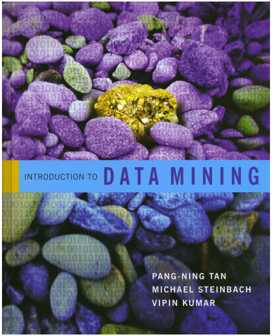
- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by "cutting" the dendrogram at the proper level
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, movie genre, etc.)





Hierarchical Clustering

- Tan, Seinbach, and Kumar, *Introduction to Data Mining*, Addison-Wesley, 2006.
- Chapter 8, Cluster Analysis.
- `http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf`



Spectral Relaxation for k-means Clustering

k-means revisited

- We assume that we have n data points $\{x_i\}_{i=1}^n \in \mathbb{R}^m$, which we organize as columns in a matrix

$$X = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{m \times n}.$$

- Let $\Pi = \{\pi_j\}_{j=1}^k$ denote a partitioning of the data in X into k clusters:

$$\pi_j = \{v \,|\, x_v \text{ belongs to cluster } j\}.$$

- Let the mean, or the centroid, of the cluster be

$$c_j = \frac{1}{n_j} \sum_{v \in \pi_j} x_v,$$

where n_j is the number of elements in π_j.

K-means revisited

- We describe K-means algorithm based on the Euclidean distance measure.
  - The tightness or coherence of cluster π_j can be measured as the sum

$$q_j = \sum_{v \in \pi_j} ||x_v - c_j||^2.$$

K-means revisited

- We describe K-means algorithm based on the Euclidean distance measure.
  - The tightness or coherence of cluster π_j can be measured as the sum

$$q_j = \sum_{v \in \pi_j} ||x_v - c_j||^2.$$

- The closer the vectors are to the centroid, the smaller the value of q_j. The quality of a clustering can be measured as the overall coherence,

$$Q(\Pi) = \sum_{j=1}^{k} \sum_{v \in \pi_j} ||x_v - c_j||^2.$$

K-means revisited

- Let e be the vector of all ones with appropriate length. It is easy to see that $c_j = X_j e / n_j$, where X_j is the data matrix of the j-th cluster.

K-means revisited

- Let e be the vector of all ones with appropriate length. It is easy to see that $c_j = X_j e / n_j$, where X_j is the data matrix of the j-th cluster.
- The sum-of-squares cost function of the j-th cluster is

$$q_j = \sum_{v \in \pi_j} ||x_v - c_j||^2 = ||X_j - c_j e^T||_F^2 = ||X_j(I_{n_j} - ee^T/n_j)||_F^2.$$

K-means revisited

- Let e be the vector of all ones with appropriate length. It is easy to see that $c_j = X_j e / n_j$, where X_j is the data matrix of the j-th cluster.
- The sum-of-squares cost function of the j-th cluster is

$$q_j = \sum_{v \in \pi_j} ||x_v - c_j||^2 = ||X_j - c_j e^T||_F^2 = ||X_j(I_{n_j} - ee^T/n_j)||_F^2.$$

- Note that $I_{n_j} - ee^T/n_j$ is a projection matrix (Why?)

K-means revisited

- Let e be the vector of all ones with appropriate length. It is easy to see that $c_j = X_j e / n_j$, where X_j is the data matrix of the j-th cluster.
- The sum-of-squares cost function of the j-th cluster is

$$q_j = \sum_{v \in \pi_j} ||x_v - c_j||^2 = ||X_j - c_j e^T||_F^2 = ||X_j (I_{n_j} - ee^T / n_j)||_F^2.$$

- Note that $I_{n_j} - ee^T / n_j$ is a projection matrix (Why?)

$$(I_{n_j} - ee^T / n_j)^2 = I_{n_j} - ee^T / n_j.$$

  It follows that

$$q_j = \text{trace}\left(X_j (I_{n_j} - ee^T / n_j) X_j^T \right) = \text{trace}\left((I_{n_j} - ee^T / n_j) X_j^T X_j \right).$$

K-means revisited

- Let e be the vector of all ones with appropriate length. It is easy to see that $c_j = X_j e / n_j$, where X_j is the data matrix of the j-th cluster.
- The sum-of-squares cost function of the j-th cluster is

$$q_j = \sum_{v \in \pi_j} ||x_v - c_j||^2 = ||X_j - c_j e^T||_F^2 = ||X_j (I_{n_j} - ee^T/n_j)||_F^2.$$

- Note that $I_{n_j} - ee^T/n_j$ is a projection matrix (Why?)

$$(I_{n_j} - ee^T/n_j)^2 = I_{n_j} - ee^T/n_j.$$

  It follows that

$$q_j = \text{trace}\left(X_j(I_{n_j} - ee^T/n_j)X_j^T\right) = \text{trace}\left((I_{n_j} - ee^T/n_j)X_j^T X_j\right).$$

  Therefore,

$$Q(\Pi) = \sum_{j=1}^k q_j = \sum_{j=1}^k \left(\text{trace}\left(X_j^T X_j\right) - \frac{e^T}{\sqrt{n_j}} X_j^T X_j \frac{e}{\sqrt{n_j}}\right).$$

k-means revisited

Define the n-by-k orthogonal matrix Y as follows

$$Y = \begin{pmatrix} e/\sqrt{n_1} & & & \\ & e/\sqrt{n_2} & & \\ & & \vdots & \\ & & & e/\sqrt{n_k} \end{pmatrix} \tag{1}$$

Then

$$Q(\Pi) = \text{trace}\left(X^T X\right) - \text{trace}\left(Y^T X^T X Y\right).$$

The k-means objective, minimization of $Q(\Pi)$, is equivalent to the maximization of trace $\left(Y^T X^T X Y\right)$ with Y is of the form in Eq. (1).

Spectral Clustering

- Ignoring the special structure of Y and let it be an arbitrary orthonormal matrix, we obtain a relaxed maximization problem

$$\max_{Y^TY=I_k} \text{trace}\left(Y^TX^TXY\right).$$

Spectral Clustering

- Ignoring the special structure of Y and let it be an arbitrary orthonormal matrix, we obtain a relaxed maximization problem

$$\max_{Y^TY=I_k} \text{trace}\left(Y^TX^TXY\right).$$

- It turns out the above trace maximization problem has a closed-form solution.
  - Theorem (Ky Fan): Let H be a symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_n$ and the corresponding eigenvectors $U = [u_1, \cdots, u_n]$. Then

$$\lambda_1 + \cdots \lambda_k = \max_{Y^TY=I_k} \text{trace}(Y^THY).$$

    Moreover, the optimal Y^* is given by $Y^* = [u_1, \cdots, u_k]Q$ with Q an arbitrary orthogonal matrix of size k by k.

- We may derive the following lower bound for the minimum of the sum-of-squares cost function:

$$\min_{\Pi} Q(\Pi) \geq \mathsf{trace}(X^T X) - \max_{Y^T Y = I_k} \mathsf{trace}\left(Y^T X^T X Y\right) = \sum_{i=k+1}^{\min\{m,n\}} \sigma_i^2(X),$$

where $\sigma_i(X)$ is the i-th largest singular value of X.

- We may derive the following lower bound for the minimum of the sum-of-squares cost function:

$$\min_{\Pi} Q(\Pi) \geq \text{trace}(X^T X) - \max_{Y^T Y = I_k} \text{trace}\left(Y^T X^T X Y\right) = \sum_{i=k+1}^{\min\{m,n\}} \sigma_i^2(X),$$

where $\sigma_i(X)$ is the i-th largest singular value of X.

- Let Y^* be the n-by-k matrix consisting of the k largest eigenvectors of $X^T X$. Each row of Y^* corresponds to a data vector. This can be considered as transforming the original data vectors which lie in a m-dimensional space to new data vectors which now lie in a k-dimensional space.

- We may derive the following lower bound for the minimum of the sum-of-squares cost function:

$$\min_{\Pi} Q(\Pi) \geq \mathsf{trace}(X^T X) - \max_{Y^T Y = I_k} \mathsf{trace}\left(Y^T X^T X Y\right) = \sum_{i=k+1}^{\min\{m,n\}} \sigma_i^2(X),$$

where $\sigma_i(X)$ is the i-th largest singular value of X.

- Let Y^* be the n-by-k matrix consisting of the k largest eigenvectors of $X^T X$. Each row of Y^* corresponds to a data vector. This can be considered as transforming the original data vectors which lie in a m-dimensional space to new data vectors which now lie in a k-dimensional space.
- **How to get our cluster assignment back?**
One might be attempted to compute the cluster assignment by applying the ordinary K-means method to those data vectors in the reduced dimension space.

Class Experiment

- Perform spectral clustering on the matrix

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \end{pmatrix}$$

Class Experiment

- Perform spectral clustering on the matrix

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \end{pmatrix}$$

- How about adding a small Gaussian noise?

Class Experiment

- Perform spectral clustering on the matrix

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \end{pmatrix}$$

- How about adding a small Gaussian noise?
- Observe the distribution of singular values and cluster number.

Issues with Clustering

Clustering

There are many clustering algorithms but unsupervised learning in general is still a challenging problem.

Clustering

There are many clustering algorithms but unsupervised learning in general is still a challenging problem.

- Non-convexity

Clustering

There are many clustering algorithms but unsupervised learning in general is still a challenging problem.

- Non-convexity
    - Sensitive to initialization (e.g., K-means).

Clustering

There are many clustering algorithms but unsupervised learning in general is still a challenging problem.

- Non-convexity
  - Sensitive to initialization (e.g., K-means).
  - Convex relaxation and convex clustering

Clustering

There are many clustering algorithms but unsupervised learning in general is still a challenging problem.

- Non-convexity
  - Sensitive to initialization (e.g., K-means).
  - Convex relaxation and convex clustering
- What is the type of clustering we are looking for?

Clustering

There are many clustering algorithms but unsupervised learning in general is still a challenging problem.

- Non-convexity
  - Sensitive to initialization (e.g., K-means).
  - Convex relaxation and convex clustering
- What is the type of clustering we are looking for?
  - Given a set of patients, what types of clustering are we expecting from the algorithm?

Clustering

There are many clustering algorithms but unsupervised learning in general is still a challenging problem.

- Non-convexity
  - Sensitive to initialization (e.g., K-means).
  - Convex relaxation and convex clustering
- What is the type of clustering we are looking for?
  - Given a set of patients, what types of clustering are we expecting from the algorithm?
  - Each distance metric specifies a clustering

Clustering

There are many clustering algorithms but unsupervised learning in general is still a challenging problem.

- Non-convexity
  - Sensitive to initialization (e.g., K-means).
  - Convex relaxation and convex clustering
- What is the type of clustering we are looking for?
  - Given a set of patients, what types of clustering are we expecting from the algorithm?
  - Each distance metric specifies a clustering
- Only applicable to elliptical shape clusters.

Clustering

There are many clustering algorithms but unsupervised learning in general is still a challenging problem.

- Non-convexity
  - Sensitive to initialization (e.g., K-means).
  - Convex relaxation and convex clustering
- What is the type of clustering we are looking for?
  - Given a set of patients, what types of clustering are we expecting from the algorithm?
  - Each distance metric specifies a clustering
- Only applicable to elliptical shape clusters.
  - Non-linear embeddings