Fuzzy Clustering

- Each point x_i takes a probability w_{ij} to belong to a cluster C_j
- Requirements
  - For each point x_i, $\sum_{j=1}^{k} w_{ij} = 1$
  - For each cluster C_j $0 < \sum_{i=1}^{m} w_{ij} < m$

Fuzzy C-Means (FCM)

- Select an initial fuzzy pseudo-partition, i.e., assign values to all the w_{ij}
- Repeat
  - Compute the centroid of each cluster using the fuzzy pseudo-partition
  - Recompute the fuzzy pseudo-partition, i.e., the w_{ij}
- Until the centroids do not change (or the change is below some threshold)

Critical Details

- Optimization on sum of the squared error (SSE): $SSE(C_1,\ldots,C_k)=\sum_{j=1}^{k}\sum_{i=1}^{m} w_{ij}^{p}\, dist(x_i,c_j)^2$
- Computing centroids: $c_j=\sum_{i=1}^{m} w_{ij}^{p} x_i \Big/ \sum_{i=1}^{m} w_{ij}^{p}$
- Updating the fuzzy pseudo-partition

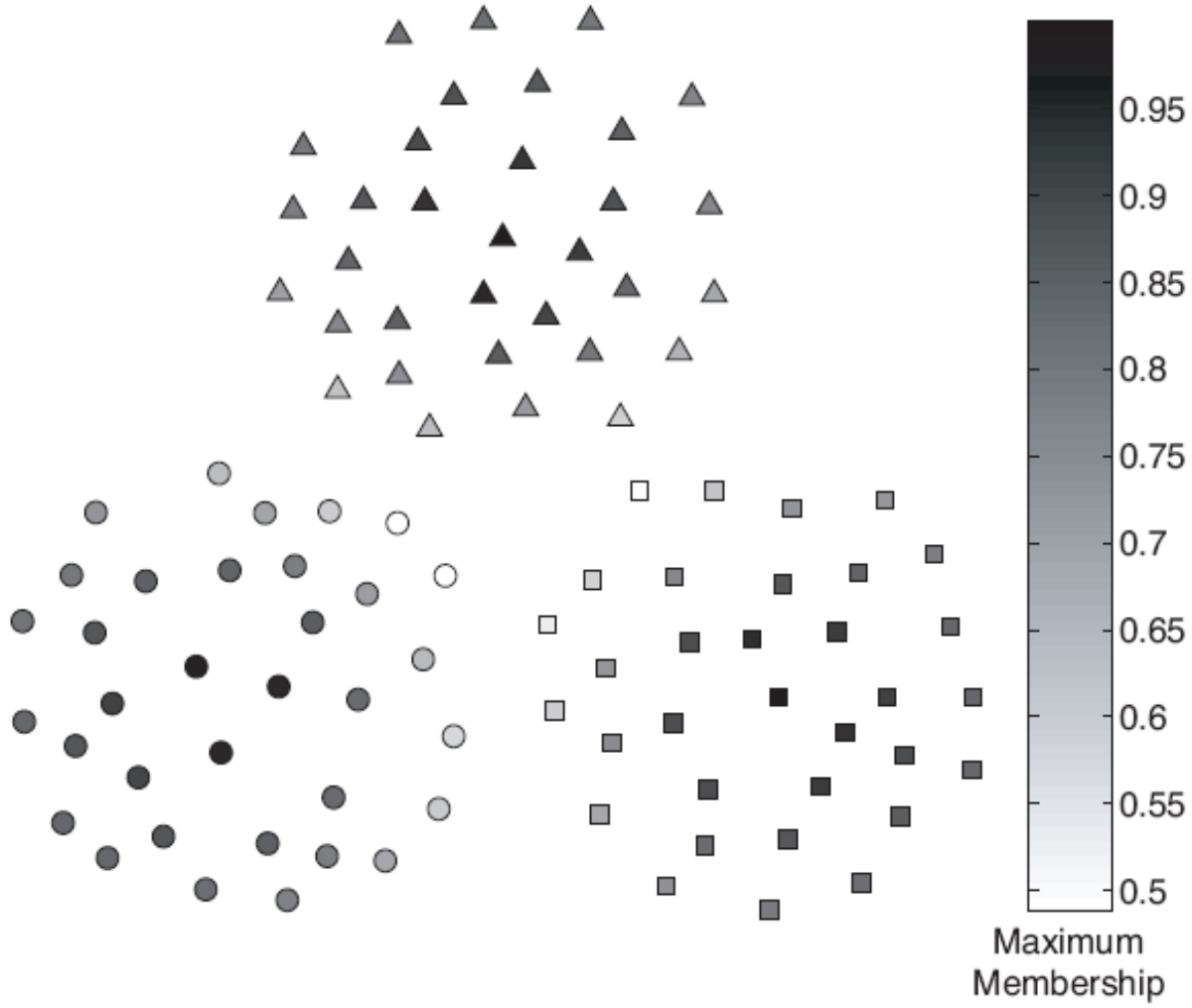
$$w_{ij}=(1/dist(x_i,c_j)^2)^{\frac{1}{p-1}} \Big/ \sum_{q=1}^{k}(1/dist(x_i,c_q)^2)^{\frac{1}{p-1}}$$

  - When p=2 $$w_{ij}=1/dist(x_i,c_j)^2 \Big/ \sum_{q=1}^{k} 1/dist(x_i,c_q)^2$$

Choice of P

- When p → 1, FCM behaves like traditional k-means
- When p is larger, the cluster centroids approach the global centroid of all data points
- The partition becomes fuzzier as p increases

Effectiveness



Mixture Models

- A cluster can be modeled as a probability distribution
  - Practically, assume a distribution can be approximated well using multivariate normal distribution
- Multiple clusters is a mixture of different probability distributions
- A data set is a set of observations from a mixture of models

Object Probability

- Suppose there are k clusters and a set X of m objects
  - Let the j-th cluster have parameter $\theta_j = (\mu_j, \sigma_j)$
  - The probability that a point is in the j-th cluster is w_j, $w_1 + \ldots + w_k = 1$
- The probability of an object x is

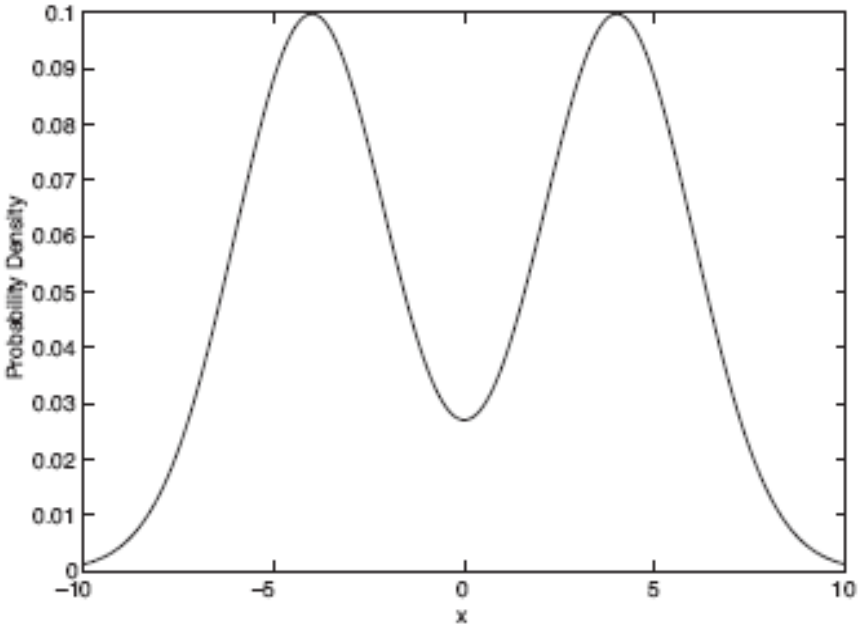
$$prob(x \mid \Theta) = \sum_{j=1}^{k} w_j p_j(x \mid \theta_j)$$

$$prob(X \mid \Theta) = \prod_{i=1}^{m} prob(x_i \mid \Theta) = \prod_{i=1}^{m} \sum_{j=1}^{k} w_j p_j(x_i \mid \theta_j)$$

Example

$$prob(x_i \mid \Theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\theta_1 = (-4,2) \quad \theta_2 = (4,2)$$



$$prob(x \mid \Theta) = \frac{1}{2\sqrt{2\pi}} e^{-\frac{(x+4)^2}{8}} + \frac{1}{2\sqrt{2\pi}} e^{-\frac{(x-4)^2}{8}}$$

Maximal Likelihood Estimation

- Maximum likelihood principle: if we know a set of objects are from one distribution, but do not know the parameter, we can choose the parameter maximizing the probability
- Maximize $$prob(x_i \mid \Theta) = \prod_{j=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$
  - Equivalently, maximize $$\log prob(X \mid \Theta) = -\sum_{i=1}^{m} \frac{(x_i - \mu)^2}{2\sigma^2} - 0.5m \log 2\pi - m \log \sigma$$

EM Algorithm

- Expectation Maximization algorithm
  - Select an initial set of model parameters
  - Repeat
    - Expectation Step: for each object, calculate the probability that it belongs to each distribution θ_i, i.e., prob($x_i|\theta_i$)
    - Maximization Step: given the probabilities from the expectation step, find the new estimates of the parameters that maximize the expected likelihood
  - Until the parameters are stable

Advantages and Disadvantages

- Mixture models are more general than k-means and fuzzy c-means
- Clusters can be characterized by a small number of parameters
- The results may satisfy the statistical assumptions of the generative models
- Computationally expensive
- Need large data sets
- Hard to estimate the number of clusters

Grid-based Clustering Methods

- Ideas
  - Using multi-resolution grid data structures
  - Using dense grid cells to form clusters
- Several interesting methods
  - CLIQUE
  - STING
  - WaveCluster

CLIQUE

- Clustering In QUEst
- Automatically identify subspaces of a high dimensional data space
- Both density-based and grid-based

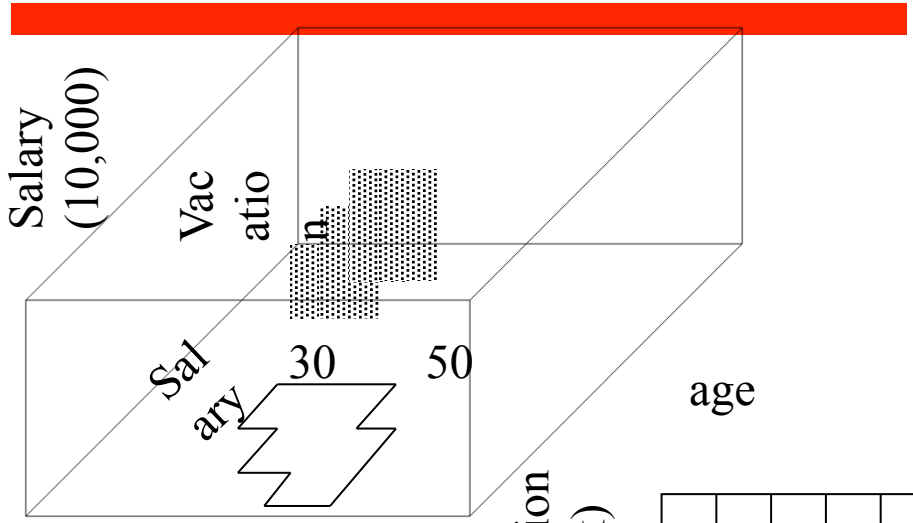
CLIQUE: the Ideas

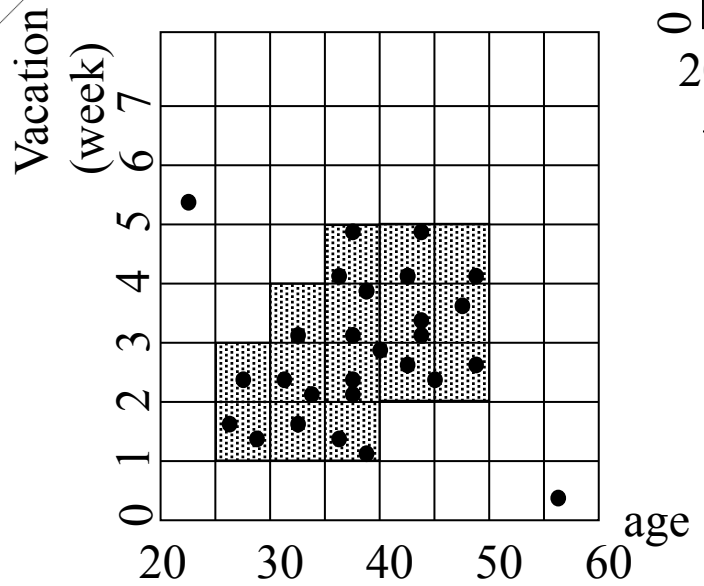
- Partition each dimension into the same number of equal length intervals
  - Partition an m-dimensional data space into non-overlapping rectangular units
- A unit is dense if the number of data points in the unit exceeds a threshold
- A cluster is a maximal set of connected dense units within a subspace

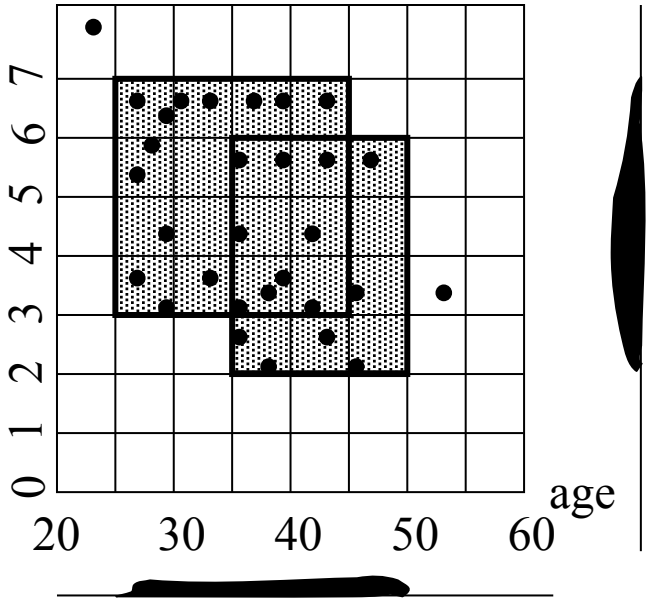
CLIQUE: the Method

- Partition the data space and find the number of points in each cell of the partition
  - Apriori: a k-d cell cannot be dense if one of its (k-1)-d projection is not dense
- Identify clusters:
  - Determine dense units in all subspaces of interests and connected dense units in all subspaces of interests
- Generate minimal description for the clusters
  - Determine the minimal cover for each cluster

CLIQUE: An Example



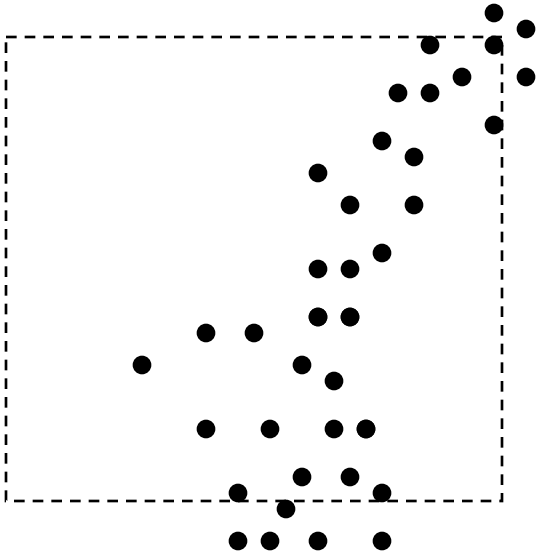




CLIQUE: Pros and Cons

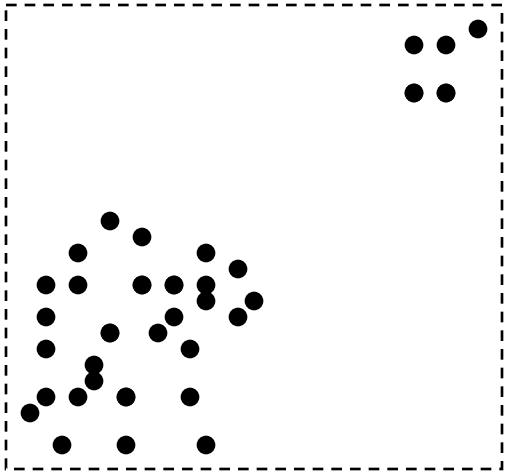
- Automatically find subspaces of the highest dimensionality with high density clusters
- Insensitive to the order of input
  - Not presume any canonical data distribution
- Scale linearly with the size of input
- Scale well with the number of dimensions
- The clustering result may be degraded at the expense of simplicity of the method

Bad Cases for CLIQUE



Parts of a cluster may be missed

A cluster from CLIQUE may contain noise



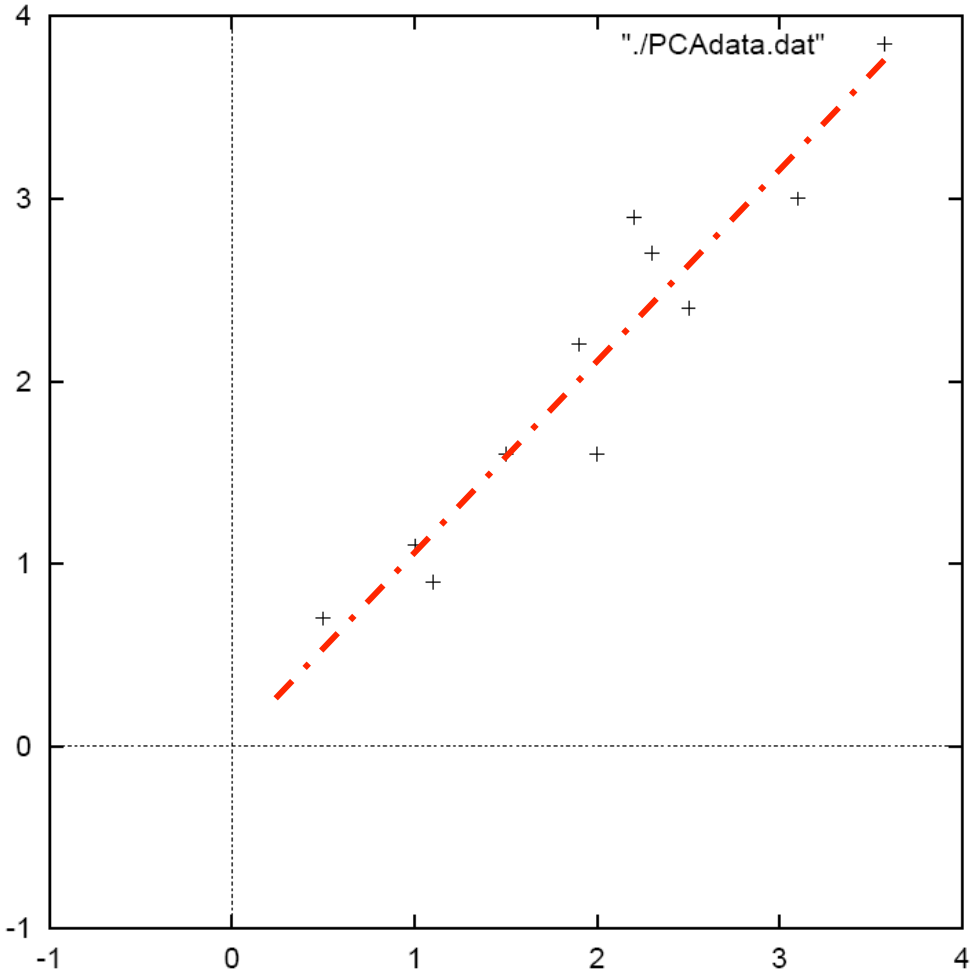
Dimensionality Reduction

- Clustering a high dimensional data set is challenging
  - Distance between two points could be dominated by noise
- Dimensionality reduction: choosing the informative dimensions for clustering analysis
  - Feature selection: choosing a subset of existing dimensions
  - Feature construction: construct a new (small) set of informative attributes

Variance and Covariance

- Given a set of 1-d points, how different are those points?
  - Standard deviation: $$s = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}}$$
  - Variance: $$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}$$
- Given a set of 2-d points, are the two dimensions correlated?
  - Covariance: $$\mathrm{cov}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{n-1}$$

Principal Components

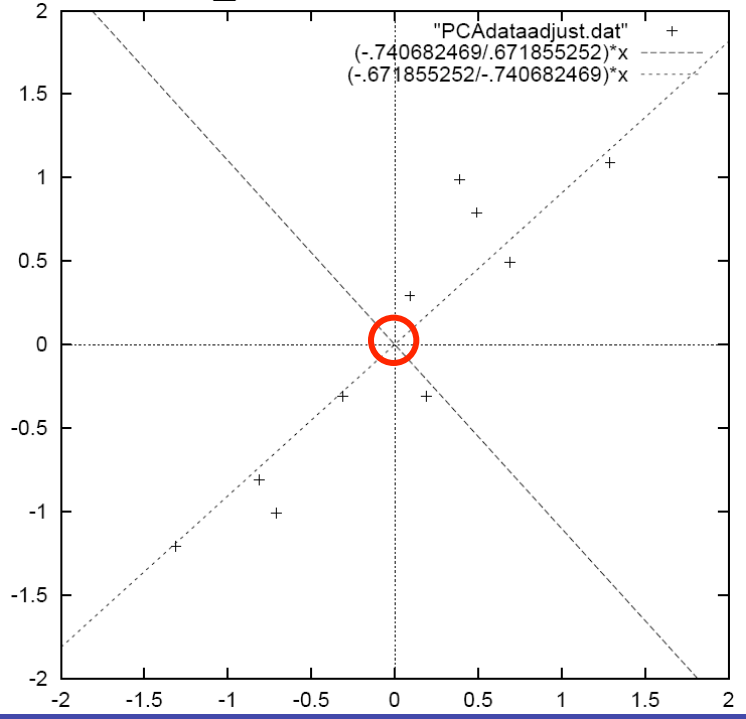


Art work and example from http://csnet.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

Step 1: Mean Subtraction

- Subtract the mean from each dimension for each data point
- Intuition: centralizing the data set

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9



x	y
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01

Step 2: Covariance Matrix

$$C = \begin{pmatrix} \mathrm{cov}(D_1, D_1) & \mathrm{cov}(D_1, D_2) & \cdots & \mathrm{cov}(D_1, D_n) \\ \mathrm{cov}(D_2, D_1) & \mathrm{cov}(D_2, D_2) & \cdots & \mathrm{cov}(D_2, D_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{cov}(D_n, D_1) & \mathrm{cov}(D_n, D_2) & \cdots & \mathrm{cov}(D_n, D_n) \end{pmatrix}$$

$$cov = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

Step 3: Eigenvectors and Eigenvalues

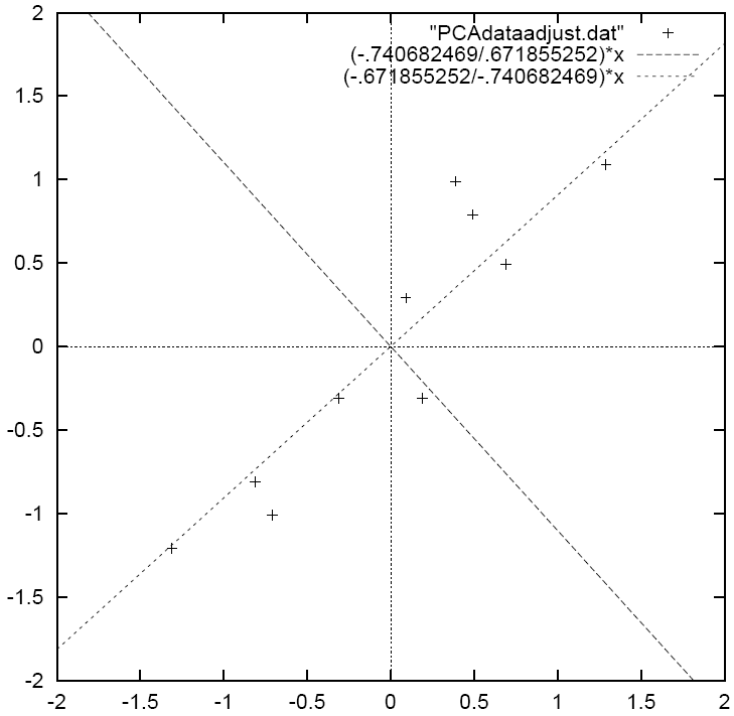
- Compute the eigenvectors and the eigenvalues of the covariance matrix
  - Intuition: find those direction invariant vectors as candidates of new attributes
  - Eigenvalues indicate how much the direction invariant vectors are scaled – the larger the better for manifest the data variance

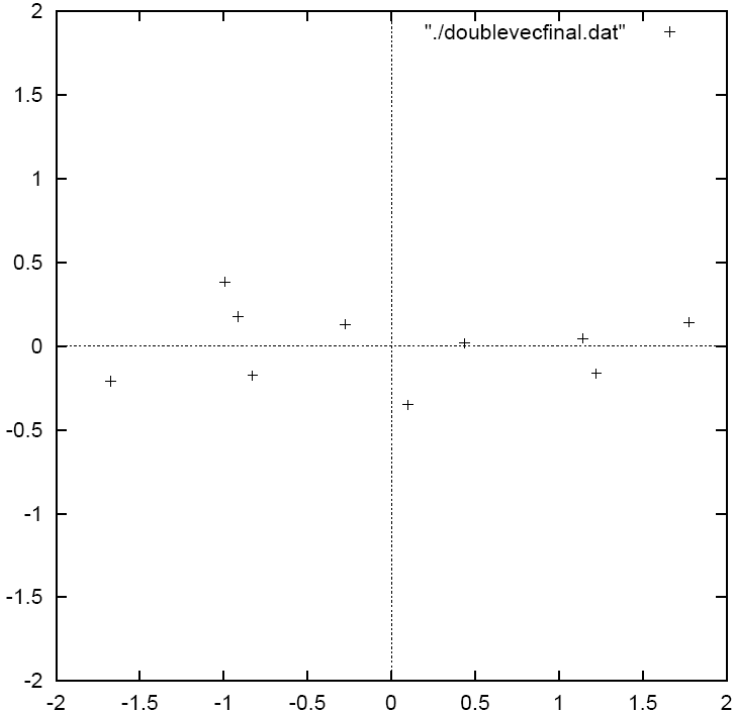
$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

Step 4: Forming New Features

- Choose the principal components and forme new features
  - Typically, choose the top-k components

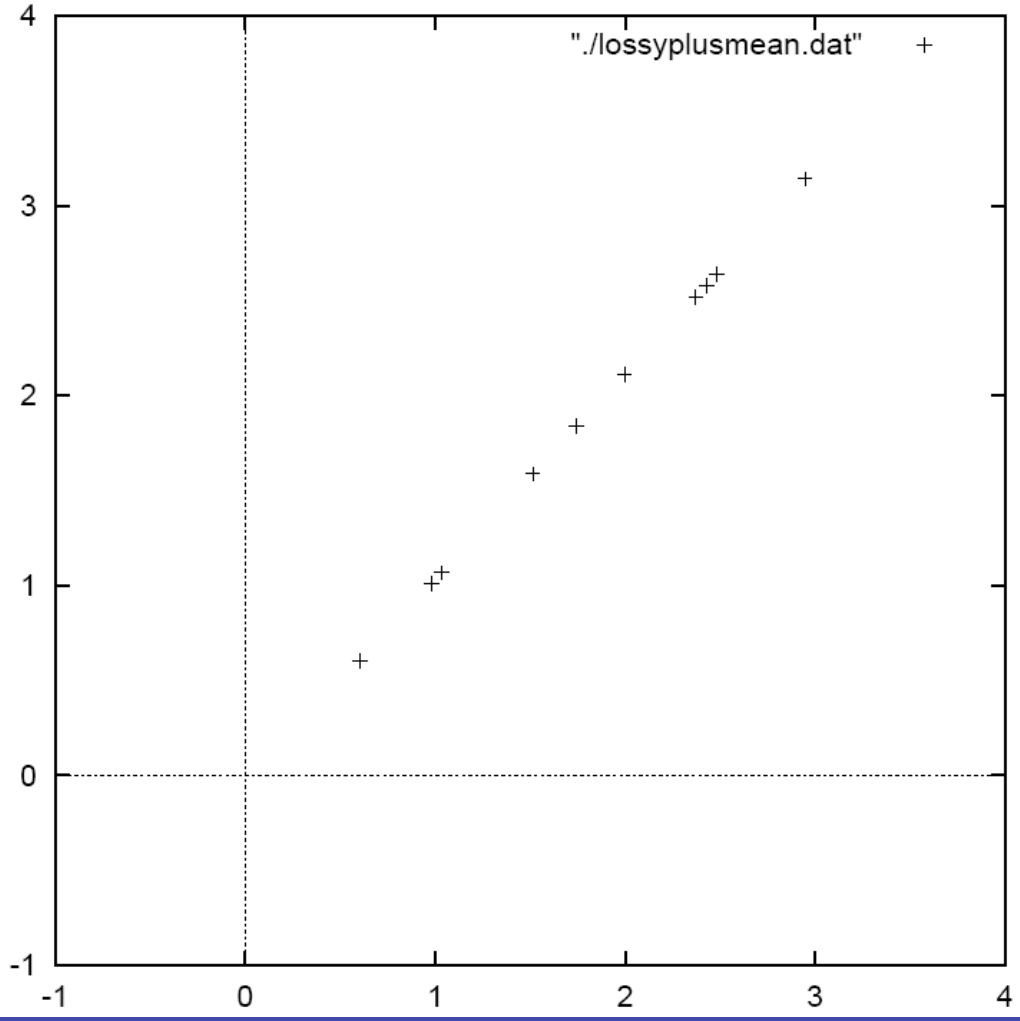




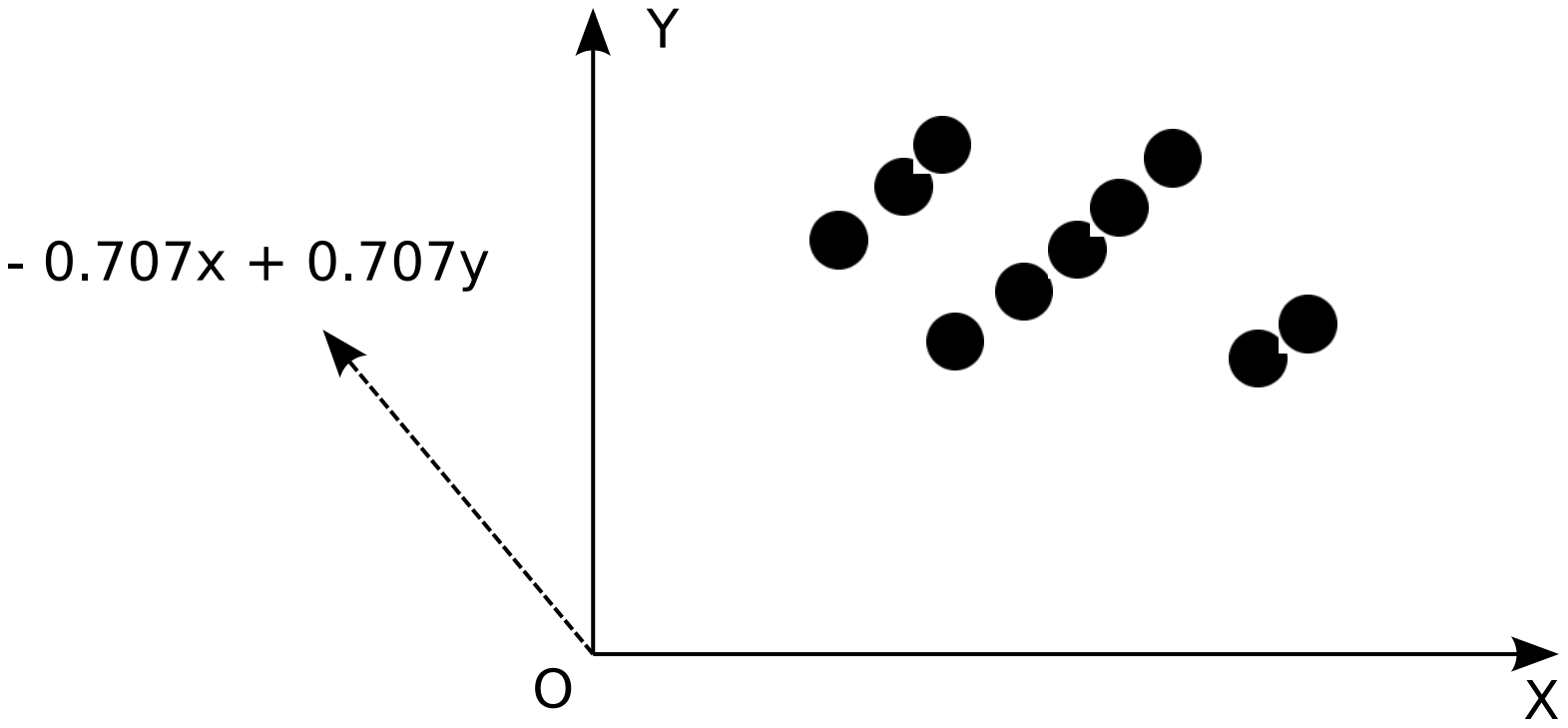
New Features

NewData = RowFeatureVector x RowDataAdjust

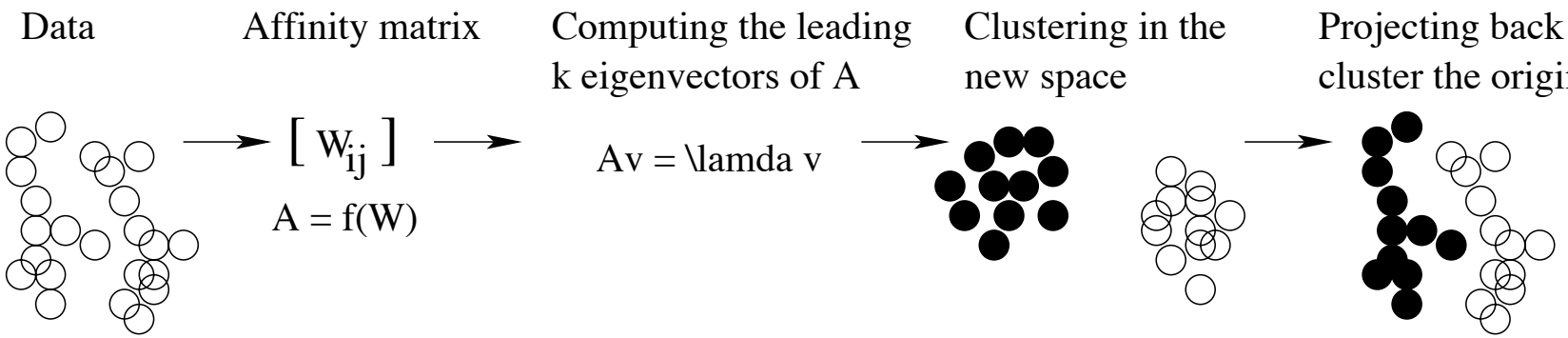
The first principal component is used



Clustering in Derived Space



Spectral Clustering



Affinity Matrix

- Using a distance measure

$$W_{ij} = e^{-\frac{dist(o_i, o_j)}{\sigma^w}}$$

  where σ is a scaling parameter controling how fast the affinity W_{ij} decreases as the distance increases

- In the Ng-Jordan-Weiss algorithm, W_{ii} is set to 0

Clustering

- In the Ng-Jordan-Weiss algorithm, we define a diagonal matrix such that

$$D_{ii} = \sum_{j=1}^{n} W_{ij}$$

- Then, $A = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$
- Use the k leading eigenvectors to form a new space
- Map the original data to the new space and conduct clustering

Is a Clustering Good?

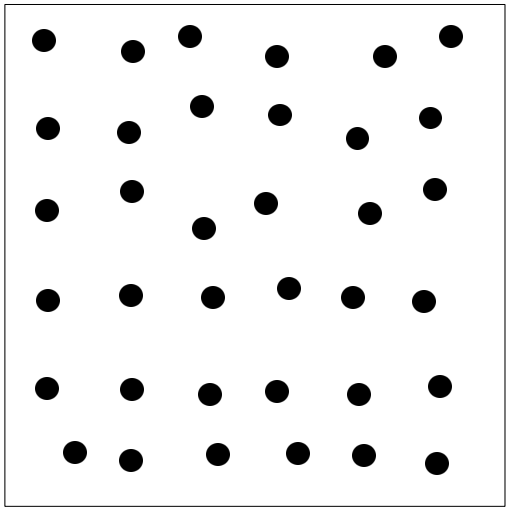
- Feasibility
  - Applying any clustering methods on a uniformly distributed data set is meaningless
- Quality
  - Are the clustering results meeting users' interest?
  - Clustering patients into clusters corresponding various disease or sub-phenotypes is meaningful
  - Clustering patients into clusters corresponding to male or female is not meaningful

Major Tasks

- Assessing clustering tendency
  - Are there non-random structures in the data?
- Determining the number of clusters or other critical parameters
- Measuring clustering quality

Uniformly Distributed Data

- Clustering uniformly distributed data is meaningless
- A uniformly distributed data set is generated by a uniform data distribution



Hopkins Statistic

- Hypothesis: the data is generated by a uniform distribution in a space
- Sample n points, p_1, …, p_n, uniformly from the space of D
- For each point p_i, find the nearest neighbor of p_i in D, let x_i be the distance between p_i and its nearest neighbor in D

$$x_i = \min_{v \in D}\{dist(p_i, v)\}$$

Hopkins Statistic

- Sample n points, q_1, …, q_n, uniformly from D
- For each qi, find the nearest neighbor of q_i in D – {q_i}, let y_i be the distance between q_i and its nearest neighbor in D – {q_i}

$$y_i = \min_{v \in D, v \neq q_i} \{dist(q_i, v)\}$$

- Calculate the Hopkins Statistic H

$$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}$$

Explanation

- If D is uniformly distributed, then $\sum_{i=1}^{n} x_i$ and $\sum_{i=1}^{n} y_i$ would be close to each other, and thus

  H would be round 0.5
- If D is skewed, then $\sum_{i=1}^{n} y_i$ would be substantially smaller, and thus H would be close to 0
- If H > 0.5, then it is unlikely that D has statistically significant clusters

Finding the Number of Clusters

- Depending on many factors
  - The shape and scale of the distribution in the data set
  - The clustering resolution required by the user
- Many methods exist
  - Set $k = \sqrt{\frac{n}{2}}$, each cluster has $\sqrt{2n}$ points on average
  - Plot the sum of within-cluster variances with respect to k, find the first (or the most significant turning point)

A Cross-Validation Method

- Divide the data set D into m parts
- Use m – 1 parts to find a clustering
- Use the remaining part as the test set to test the quality of the clustering
  - For each point in the test set, find the closest centroid or cluster center
  - Use the squared distances between all points in the test set and the corresponding centroids to measure how well the clustering model fits the test set
- Repeat m times for each value of k, use the average as the quality measure

Measuring Clustering Quality

- Ground truth: the ideal clustering determined by human experts
- Two situations
  - There is a known ground truth – the extrinsic (supervised) methods, comparing the clustering against the ground truth
  - The ground truth is unavailable – the intrinsic (unsupervised) methods, measuring how well the clusters are separated

Quality in Extrinsic Methods

- Cluster homogeneity: the more pure the clusters in a clustering are, the better the clustering
- Cluster completeness: objects in the same cluster in the ground truth should be clustered together
- Rag bag: putting a heterogeneous object into a pure cluster is worse than putting it into a rag bag
- Small cluster preservation: splitting a small cluster in the ground truth into pieces is worse than splitting a bigger one

Bcubed Precision and Recall

- $D = \{o_1, \ldots, o_n\}$
  - $L(o_i)$ is the cluster of o_i given by the ground truth
- C is a clustering on D
  - $C(o_i)$ is the cluster-id of o_i in C
- For two objects o_i and o_j, the correctness is 1 if $L(o_i) = L(o_j) \leftrightarrow C(o_i) = C(o_j)$, 0 otherwise

Bcubed Precision and Recall

- Precision

$$\text{Precision BCubed} = \frac{\sum_{i=1}^{n} \frac{\sum_{\boldsymbol{o_j}:i\neq j, C(\boldsymbol{o_i})=C(\boldsymbol{o_j})} \text{Correctness}(\boldsymbol{o_i}, \boldsymbol{o_j})}{\|\{\boldsymbol{o_j} | i \neq j, C(\boldsymbol{o_i}) = C(\boldsymbol{o_j})\}\|}}{n}.$$

- Recall

$$\text{Recall BCubed} = \frac{\sum_{i=1}^{n} \frac{\sum_{\boldsymbol{o_j}:i\neq j, L(\boldsymbol{o_i})=L(\boldsymbol{o_j})} \text{Correctness}(\boldsymbol{o_i}, \boldsymbol{o_j})}{\|\{\boldsymbol{o_j} | i \neq j, L(\boldsymbol{o_i}) = L(\boldsymbol{o_j})\}\|}}{n}$$

Silhouette Coefficient

- No ground truth is assumed
- Suppose a data set D of n objects is partitioned into k clusters, C_1, …, C_k
- For each object o,
  - Calculate a(o), the average distance between o and every other object in the same cluster – compactness of a cluster, the smaller, the better
  - Calculate b(o), the minimum average distance from o to every objects in a cluster that o does not belong to – degree of separation from other clusters, the larger, the better

Silhouette Coefficient

$$a(o) = \frac{\sum_{o,o' \in C_i, o' \neq o} dist(o, o')}{|C_i| - 1}$$

$$b(o) = \min_{C_j : o \notin C_j} \{ \frac{\sum_{o' \in C_j} dist(o, o')}{|C_j|} \}$$

- Then

$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}$$

- Use the average silhouette coefficient of all objects as the overall measure

Multi-Clustering

- A data set may be clustered in different ways
  - In different subspaces, that is, using different attributes
  - Using different similarity measures
  - Using different clustering methods
- Some different clusterings may capture different meanings of categorization
  - Orthogonal clusterings
- Putting users in the loop

To-Do List

- Read Chapters 10.5, 10.6, and 11.1
- Find out how Gaussian mixture can be used in SPARK MLlib
- (for thesis-based graduate students only) Learn LDA (Latent Dirichlet allocation) by yourself