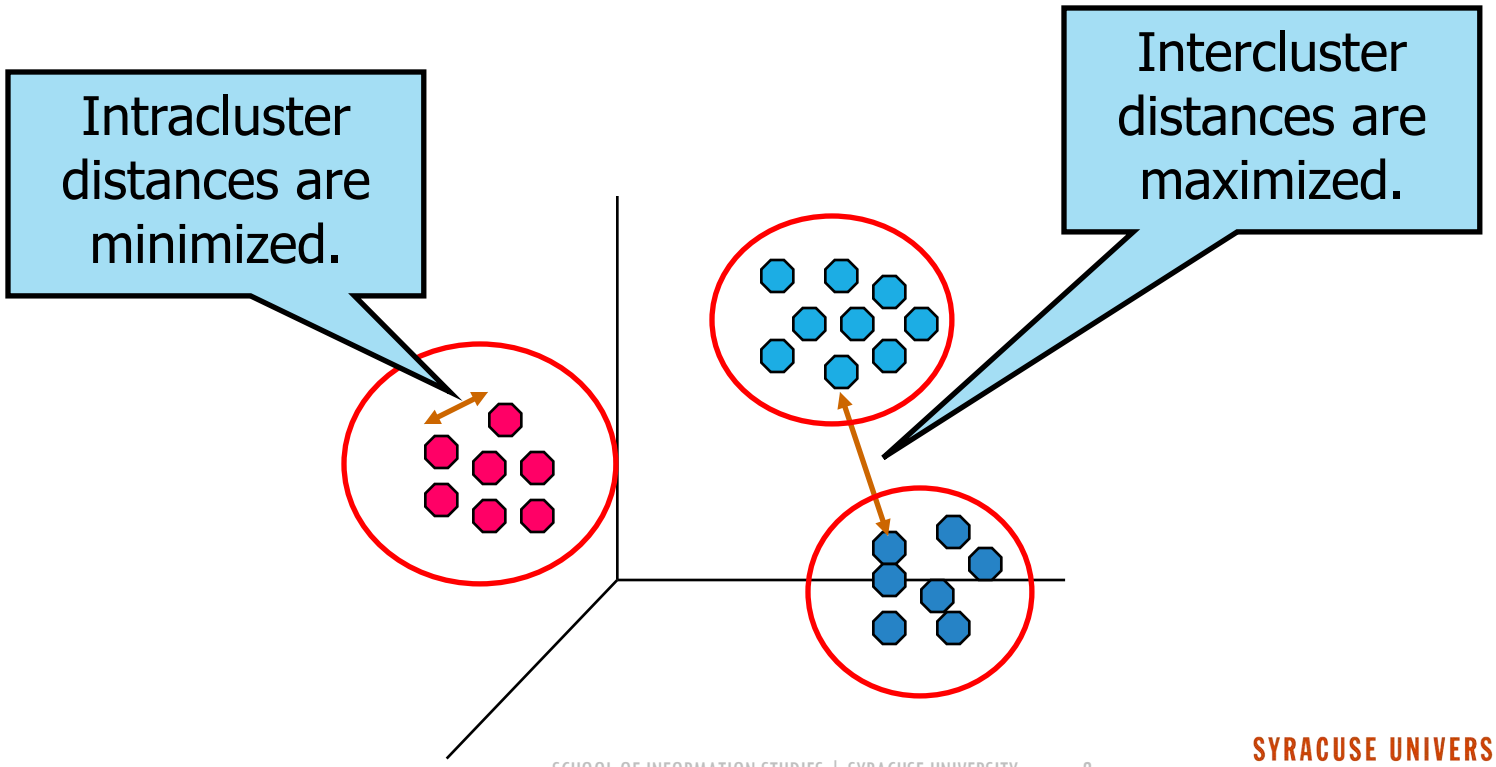
WHAT IS CLUSTERING ANALYSIS?

SYRACUSE UNIVERSITY
School of Information Studies

WHAT IS CLUSTER ANALYSIS?

Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



WHAT IS CLUSTER ANALYSIS?

Unsupervised learning: No predefined classes

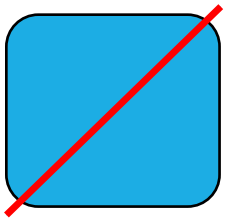
Typical applications:

- Explore a large data set without prior knowledge about it
  - Customer segmentation, document clustering, etc.
- Classification without training data
  - Usually less accurate than supervised learning methods
- Outlier detection
  - E.g., identify plagiarism cases

REQUIREMENTS OF CLUSTERING IN DATA MINING

Scalability

Ability to explore large data set



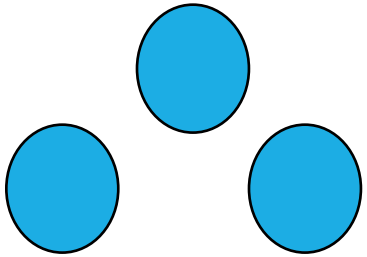
Difficult to cluster because the two clusters are overlapped

Ability to deal with different types of attributes

Nominal, ordinal, numeric

Discovery of clusters with arbitrary shape

Spherical vs. other shapes



Easy to cluster using distance-based methods

REQUIREMENTS OF CLUSTERING IN DATA MINING

Minimal requirements for domain knowledge to determine input parameters

- The number of desired clusters
- E.g., how many topics in congressional speeches

Able to deal with noise and outliers

Insensitive to order of input records

High dimensionality

- Sparse data

Interpretability and usability

TYPES OF CLUSTERINGS

A clustering is a set of clusters

Important distinction between hierarchical and partitional sets of clusters

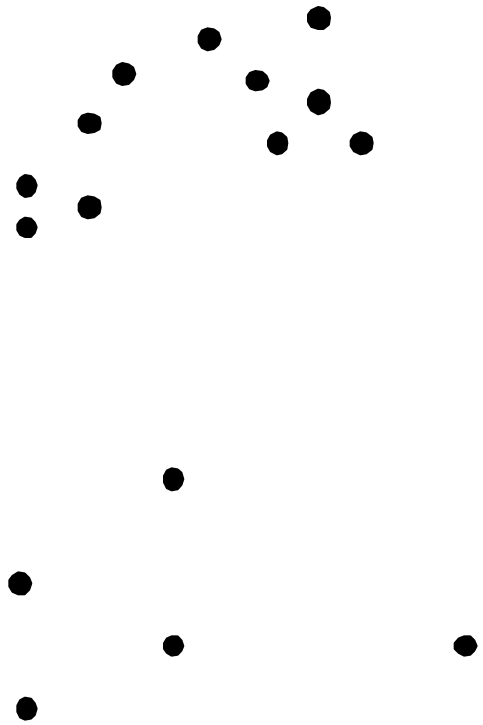
Partitional (flat) clustering:

A division of data objects into nonoverlapping subsets (clusters) such that each data object is in exactly one subset

Hierarchical clustering:

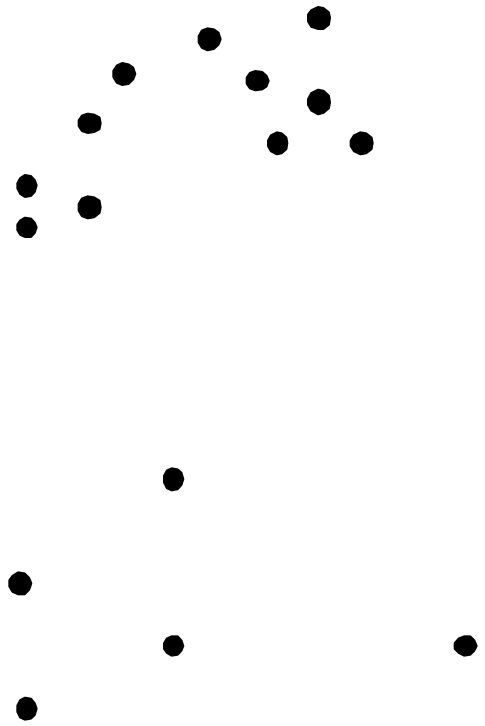
A set of nested clusters organized as a hierarchical tree

PARTITIONAL CLUSTERING

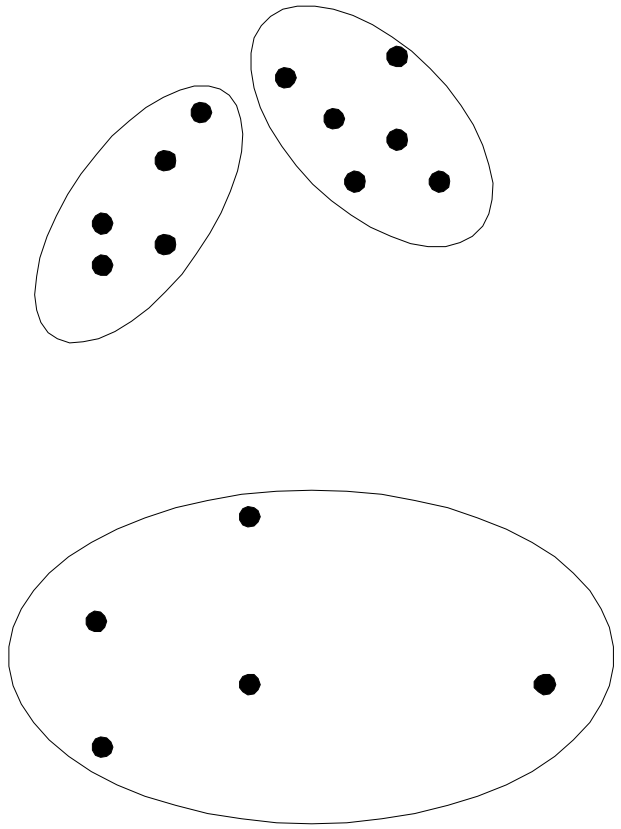


Original Points

PARTITIONAL CLUSTERING

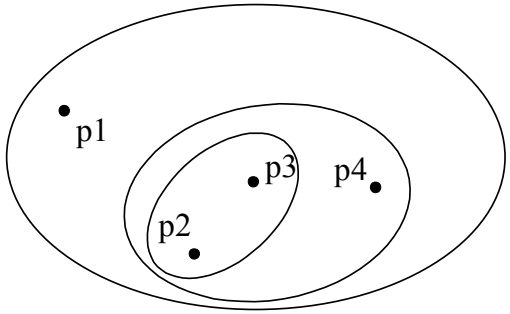


Original Points

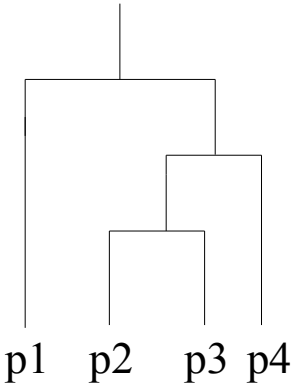


A Partitional Clustering

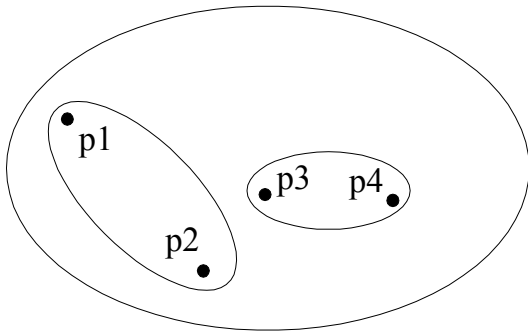
HIERARCHICAL CLUSTERING



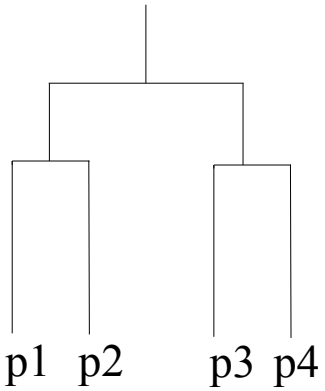
Traditional Hierarchical Clustering



Traditional Dendrogram



Nontraditional Hierarchical Clustering



Nontraditional Dendrogram

MAJOR CLUSTERING APPROACHES

Partitioning approach:

Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors.

Typical methods: k-means, k-medoids, CLARANS, EM

Hierarchical approach:

Create a hierarchical decomposition of the set of data (or objects) using some criterion

Typical methods: DIANA, AGNES, BIRCH, ROCK, CHAMELEON

DISTANCE MEASURE

SYRACUSE UNIVERSITY
School of Information Studies

DISTANCE MEASURES

Similarity and distance: Two opposite concepts

- Similarity measures how close or similar two examples are.
- Distance measures how far or different two examples are.

The definitions of distance functions are dependent on variable types: numeric, nominal. Many data sets contain mixed types of attributes.

- Example: How similar are these two people?
  - *i* = (Refund = No, Married, Income = 120K)
  - *j* = (Refund = Yes, Married, Income = 90K)

NUMERIC ATTRIBUTES

If the data have all numeric attributes, distance measures can compare the numeric values of the attributes.

Some popular ones include *Minkowski distance*

$$d(i,j) = \sqrt[q]{(|x_{i1} \quad x_{j1}|^q + |x_{i2} \quad x_{j2}|^q + ... + |x_{ip} \quad x_{jp}|^q)}$$

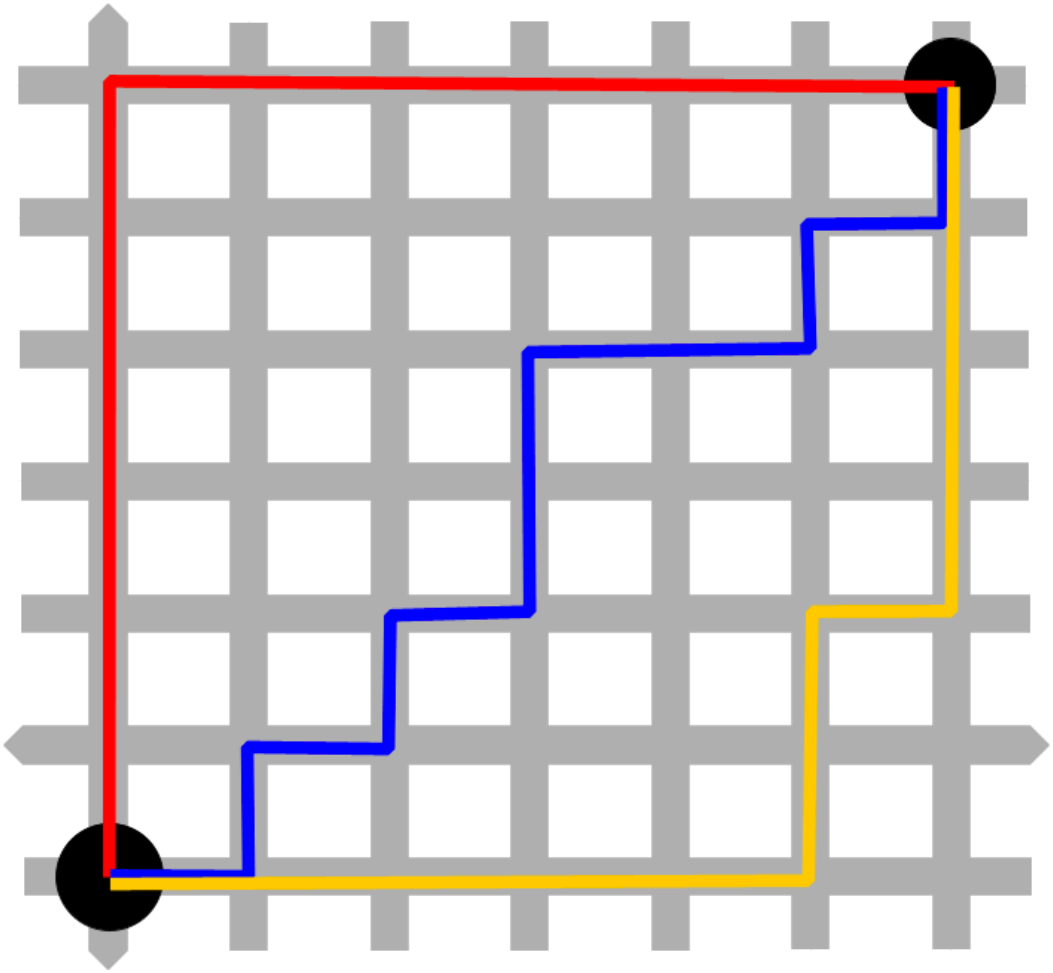
where $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two p-dimensional data instances, and q is a positive integer.

If $q = 1$, d is *Manhattan distance.*

Taking the absolute value of the differences between attribute values

$$d(i,j) = |x_{i1} \quad x_{j1}| + |x_{i2} \quad x_{j2}| + ... + |x_{ip} \quad x_{jp}|$$

MANHATTAN DISTANCE

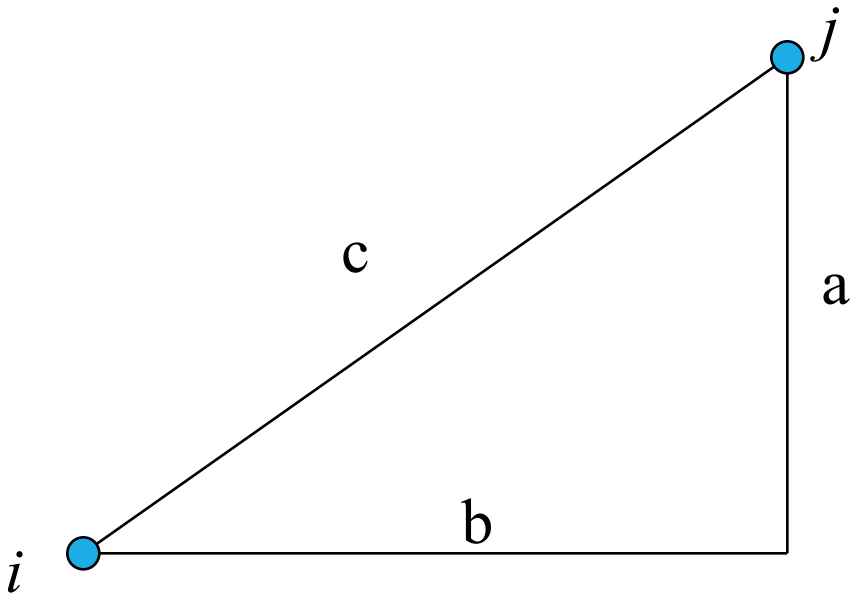


https://upload.wikimedia.org/wikipedia/commons/d/de/Manhattan_distance_bgiu.png

EUCLIDEAN DISTANCE

When $q = 2$, d is *Euclidean distance*:

$$d(i,j) = \sqrt{(|x_{i1} \quad x_{j1}|^2 + |x_{i2} \quad x_{j2}|^2 + ... + |x_{ip} \quad x_{jp}|^2)}$$



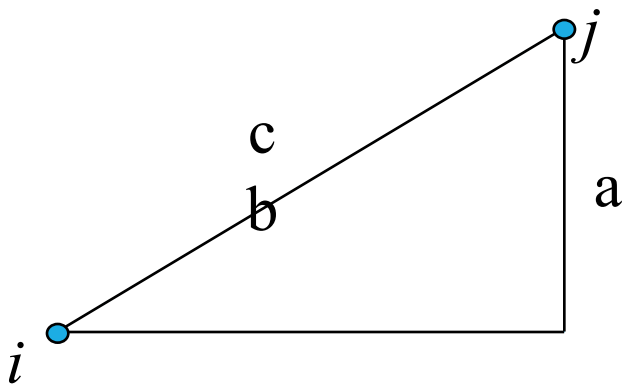
$$d_1(i,j) = a + b$$

$$d_2(i,j) = \sqrt{a^2 + b^2} = c$$

PROPERTIES OF DISTANCE MEASURE

A distance measure should satisfy the following requirements:

- $d(i, j) \geq 0$ (nonnegative value)
- $d(i, i) = 0$ (zero distance to itself)
- $d(i, j) = d(j, i)$ (symmetric measure)
- $d(i, j) \leq d(i, k) + d(k, j)$ (shortest distance between two points)



DISTANCE BETWEEN NOMINAL VALUES

Example: How similar are these two people?

i = (Refund = Yes, Married, Income = 120K)

j = (Refund = No, Divorced, Income = 90K)

Taxpayer	Refund	Marital Status	Income in Thousands
i	Yes	Married	120
j	No	Divorced	90

METHOD 1: SIMPLE MATCHING

Taxpayer	Refund	Marital Status	Income in Thousands
i	Yes	Married	120
j	No	Divorced	90

m: Number of matches; *p*: Total number of nominal variables

$$d(i,j) = \frac{p \quad m}{p}$$

METHOD 2: CONVERT NOMINAL TO BINARY VARIABLES

Taxpayer	Refund	Marital Status	Income in Thousands
i	Yes	Married	120
j	No	Divorced	90

Convert a nominal attribute to multiple binary attributes, and treat binary attributes as numeric (0 or 1).

Taxpayer	Refund	Married?	Divorced?	Single?	Income
1	1	1	0	0	120
2	0	0	1	0	90

BINARY VARIABLES: SYMMETRIC OR ASYMMETRIC

All patients run through many tests.

How different are their test results?

Patient	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
Jack	1	0	1	0	0	0
Mary	1	0	1	0	1	0

BINARY VARIABLES: SYMMETRIC OR ASYMMETRIC

A contingency table for binary data

Gives the number of attributes of each pair of values

		Mary		
		1	0	*sum*
Jack	1	a	b	$a+b$
	0	c	d	$c+d$
	sum	$a+c$	$b+d$	p

Patient	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
Jack	1	0	1	0	0	0
Mary	1	0	1	0	1	0

SYMMETRIC BINARY ATTRIBUTES

Distance measure for symmetric binary attributes:

		Object j		
		1	0	*sum*
Object i	1	a	b	$a+b$
	0	c	d	$c+d$
	sum	$a+c$	$b+d$	p

$$d(i,j) = \frac{b+c}{a+b+c+d}$$

ASYMMETRIC BINARY ATTRIBUTES

If most test results are negative, *d* will be much greater than *a*, *b*, and *c*. Sharing many negative test results is not that informative to doctors.

		Object *j*		
		1	0	*sum*
Object *i*	1	*a*	*b*	*a*+*b*
	0	*c*	*d*	*c*+*d*
	sum	*a*+*c*	*b*+*d*	*p*

Distance measure for asymmetric binary attributes:

$$d(i,j) = \frac{b+c}{a+b+c}$$

DISTANCE BETWEEN ORDINAL VALUES

Method 1: Treat as nominal.

Method 2: Treat as numeric.

ATTRIBUTES OF MIXED TYPES

A database may contain different types of attributes: Symmetric binary, asymmetric binary, nominal, ordinal, numerical

How to compute the distance between examples with heterogeneous attributes?

Calculate distance for each type of attribute and aggregate.

SIMILARITY MEASURE

If defining a distance measured in [0,1] range, similarity can be defined as 1 – d.

Other similarity measures:

- Cosine similarity measure

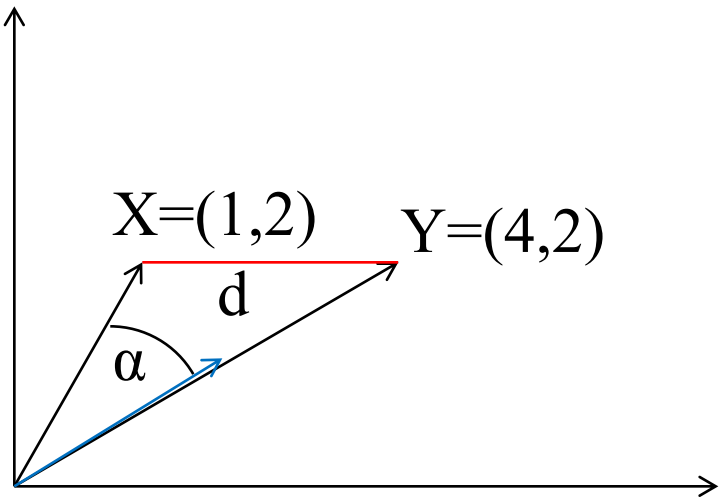
VECTOR SPACE REPRESENTATION AND COSINE SIMILARITY

Distance and similarity measures

Euclidean distance

$$d = x \quad y = \sqrt{(x_1 \quad y_1)^2 + (x_2 \quad y_2)^2}$$

$$= \sqrt{(1 \quad 4)^2 + (2 \quad 2)^2} = 3$$



Cosine similarity

$$\cos(\quad) = \frac{x \quad y}{|x \| y|} = \frac{x_1 y_1 + x_2 y_2}{\sqrt{x_1^2 + x_2^2}\ \sqrt{y_1^2 + y_2^2}}$$

$$= \frac{1 \cdot 4 + 2 \cdot 2}{\sqrt{1^2 + 2^2}\ \sqrt{4^2 + 2^2}} = \frac{8}{\sqrt{5}\ \sqrt{20}} = 0.8$$

COSINE SIMILARITY

In the range of [0,1]:

- "0" means two vectors are perpendicular to each other.
- "1" means same vector direction and length.

Commonly used in information retrieval and text mining to compare document similarity

- High-dimensional space
  - Each word in the vocabulary is a dimension.

IMPORTANCE OF NORMALIZATION

SYRACUSE UNIVERSITY
School of Information Studies

IMPORTANCE OF NORMALIZATION 1

Different variables might use different scales

Age: [0,120]

Income: [0,2M]

If averaging the difference on these two variables, income would weigh much more than age.

Solution: Normalize both variables to the same scale, e.g., [0,1].

IMPORTANCE OF NORMALIZATION 2

Different examples or vectors might differ greatly in length.

E.g., for text documents, the vector lengths of long documents are much greater than for short documents.

AN EXAMPLE OF NORMALIZATION IN INFORMATION RETRIEVAL

	a	against	but	camera	gallery	hit	husband	images	imagined
music.1	13	0	3	0	0	0	0	0	0
music.2	18	0	7	0	0	2	0	0	0
music.3	33	0	2	0	3	1	0	0	0
music.4	28	0	11	0	0	1	0	0	0
music.5	10	0	0	0	1	0	0	0	0
art.1	20	0	3	2	0	0	1	0	0
art.2	51	0	9	1	4	0	0	2	1
art.3	55	1	6	11	1	0	2	8	0
art.4	64	2	7	0	0	0	0	0	2
art.5	11	1	1	0	0	0	0	2	0

	instruments	melody	new	old	photographs	photography	songs	wife
music.1	3	1	0	0	0	0	0	0
music.2	0	0	1	1	0	0	0	0
music.3	0	0	2	1	0	0	3	0
music.4	0	0	2	0	0	0	0	1
music.5	0	1	2	1	0	0	1	0
art.1	0	0	1	0	0	1	0	1
art.2	0	0	3	3	1	4	0	1
art.3	1	0	5	2	0	3	0	2
art.4	0	0	1	0	0	0	0	2
art.5	0	0	0	0	1	1	0	0

Table 2: Bag-of-words vectors for five randomly selected stories classified as "music", and five classified as "art" (but not music), from the *Times* corpus. The table shows a selection of the 700 features.

http://www.stat.cmu.edu/~cshalizi/350/lectures/01/lecture-01.pdf

LONGER DOCS TEND TO BE FAR AWAY FROM SHORT ONES BASED ON RAW EUCLIDEAN DISTANCE

	a	against	but	camera	gallery	hit	husband	images	imagined
music.1	13	0	3	0	0	0	0	0	0
music.2	18	0	7	0	0	2	0	0	0
music.3	33	0	2	0	3	1	0	0	0
music.4	28	0	11	0	0	1	0	0	0
music.5	10	0	0	0	1	0	0	0	0
art.1	20	0	3	2	0	0	1	0	0
art.2	51	0	9	1	4	0	0	2	1
art.3	55	1	6	11	1	0	2	8	0
art.4	64	2	7	0	0	0	0	0	2
art.5	11	1	1	0	0	0	0	2	0

	instruments	melody	new	old	photographs	photography	songs	wife
music.1	3	1	0	0	0	0	0	0
music.2	0	0	1	1	0	0	0	0
music.3	0	0	2	1	0	0	3	0
music.4	0	0	2	0	0	0	0	1
music.5	0	1	2	1	0	0	1	0
art.1	0	0	1	0	0	1	0	1
art.2	0	0	3	3	1	4	0	1
art.3	1	0	5	2	0	3	0	2
art.4	0	0	1	0	0	0	0	2
art.5	0	0	0	0	1	1	0	0

Table 2: Bag-of-words vectors for five randomly selected stories classified as "music", and five classified as "art" (but not music), from the *Times* corpus. The table shows a selection of the 700 features.

http://www.stat.cmu.edu/~cshalizi/350/lectures/01/lecture-01.pdf

NORMALIZATION BY DOC LENGTH (L-1)

2.1 Normalization

Just looking at the Euclidean distances between document vectors doesn't work, at least if the documents are at all different in size. Instead, we need to **normalize** by document size, so that we can fairly compare short texts with long ones. There are (at least) two ways of doing this.

Document length normalization Divide the word counts by the total number of words in the document. In symbols,

$$\vec{x} \mapsto \frac{\vec{x}}{\sum_{i=1}^{p} x_i}$$

Notice that all the entries in the normalized vector are non-negative fractions, which sum to 1. The i^{th} component is thus the probability that if we pick a word out of the bag at random, it's the i^{th} entry in the lexicon.

NORMALIZATION BY EUCLIDEAN LENGTH (L-2)

Euclidean length normalization Divide the word counts by the Euclidean length of the document vector:

$$\vec{x} \mapsto \frac{\vec{x}}{\|\vec{x}\|}$$

For search, normalization by Euclidean length tends to work a bit better than normalization by word-count, apparently because the former de-emphasizes words which are rare in the document.

Cosine "distance" is actually a similarity measure, not a distance:

$$d_{\cos}\,\vec{x}, \vec{y} = \frac{\sum_i x_i y_i}{\|\vec{x}\|\|\vec{y}\|}$$

It's the cosine of the angle between the vectors $\vec{x}$ and $\vec{y}$.

COMPARE RESULTS WITH AND WITHOUT NORMALIZATION

	Best match by similarity measure		
	Euclidean	Euclidean + word-count	Euclidean + length
music.1	art.5	art.4	art.4
music.2	art.1	music.4	music.4
music.3	music.4	music.4	art.3
music.4	music.2	art.1	art.3
music.5	art.5	music.3	music.3
art.1	music.1	art.4	art.3
art.2	music.4	art.4	art.4
art.3	art.4	art.4	art.4
art.4	art.3	art.3	art.3
art.5	music.1	art.3	art.3
error count	6	2	3

Table 3: Closest matches for the ten documents, as measured by the distances between bag-of-words vectors, and the total error count (number of documents whose nearest neighbor is in the other class).

WEIGHTED DISTANCE AND SIMILARITY

For some data, some dimensions are more important than others, and thus their similarity or distance carries more weight. In these cases, we can assign different weights to individual dimensions or attributes.

E.g.: $d(i,j)=2\cdot|x_{i1} \quad x_{j1}|+|x_{i2} \quad x_{j2}|$

$$similarity(p,q) = \frac{\sum_{k=1}^{n} w_k \delta_k s_k}{\sum_{k=1}^{n} \delta_k}$$

$$distance(p,q) = \left(\sum_{k=1}^{n} w_k |p_k - q_k|^r \right)^{1/r}$$

SIMILARITY AND DISTANCE MEASURE IN R

```
# define a function that calculates the Euclidean distance between two vectors a and b
ED = function(a,b) sqrt(sum((a-b)^2))

# define a function that calculates the cosine similarity between two vectors a and b
CS = function(a,b) a%*% b/sqrt(a%*%a*b%*%b)

# given two vectors a and b
A = c(1,2,3)
B = c(4,5,6)

# call functions to calculate distance
ED(a,b) = 5.196
CS(a,b) = 0.975
```

SUMMARY OF DISTANCE AND SIMILARITY

Manhattan and Euclidean distance for numeric variables

Properties of distance measure

Distance for nominal variables: Count matches or convert to binary

Symmetric vs. asymmetric binary variables

Convert ordinal to either numeric or nominal

Calculate distance or similarity on each attribute or attribute group and then average over all

Cosine similarity

Importance of normalization

K-MEANS ALGORITHM

SYRACUSE UNIVERSITY
School of Information Studies

CENTROID OF A CLUSTER

Centroid: The “gravity center” of a cluster, which is calculated as the average of all data examples in a cluster

For numeric variable, use the mean as the average.

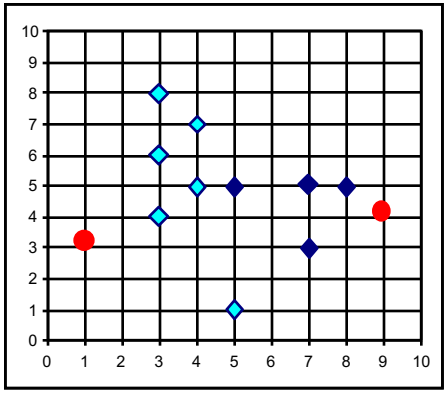
For nominal data, use the mode as the average.

THE K-MEANS CLUSTERING METHOD

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

THE K-MEANS CLUSTERING METHOD

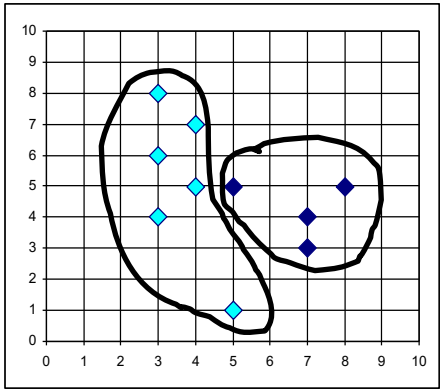
Example:



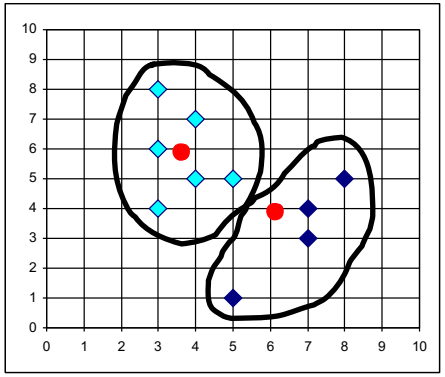
k = 2

Arbitrarily choose k object as initial cluster center.

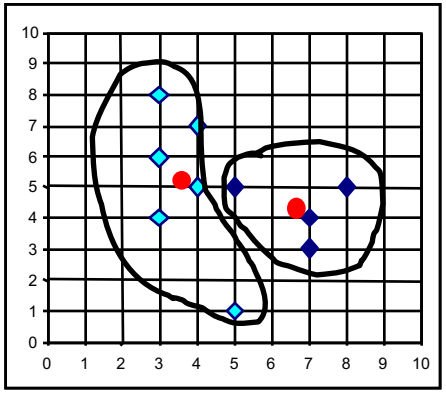
Assign each object to most similar center.



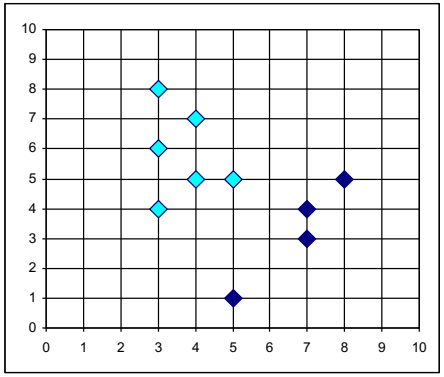
Reassign.



Update the cluster means.

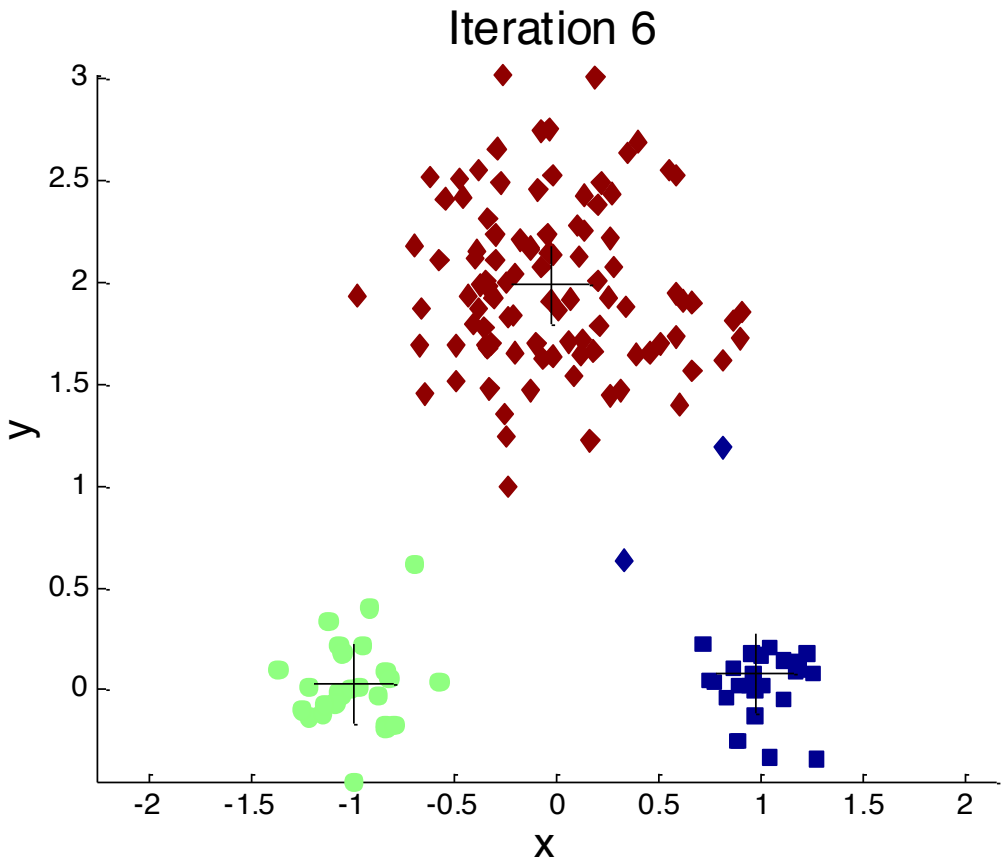


Reassign.



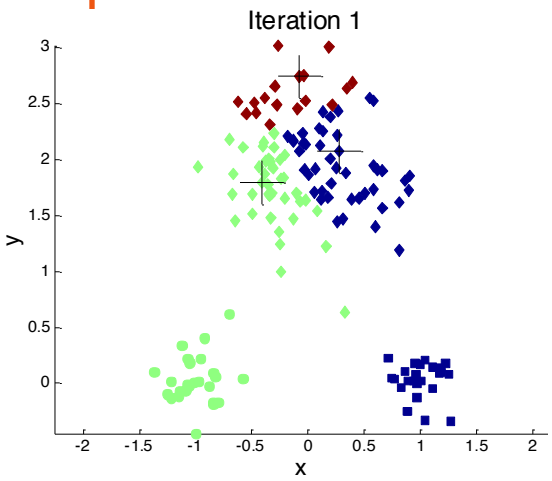
Update the cluster means.

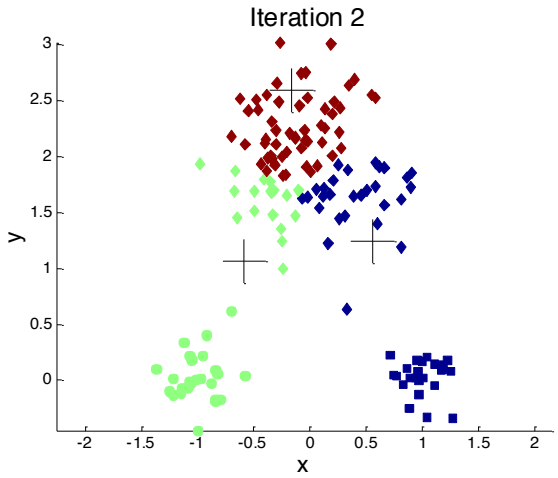
IMPORTANCE OF CHOOSING INITIAL CENTROIDS

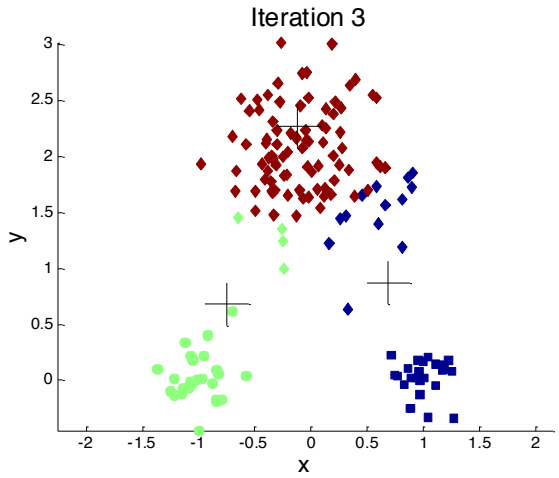


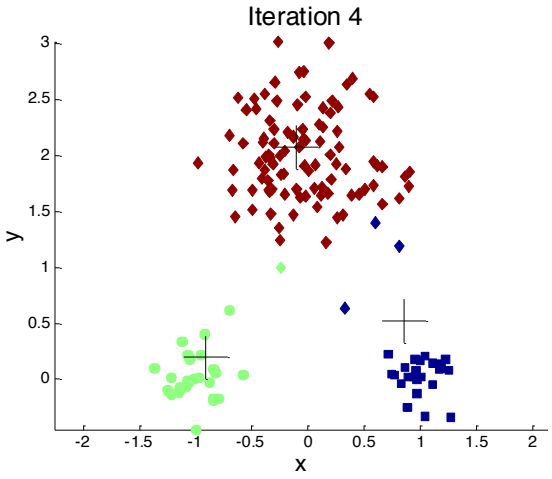
A good clustering result

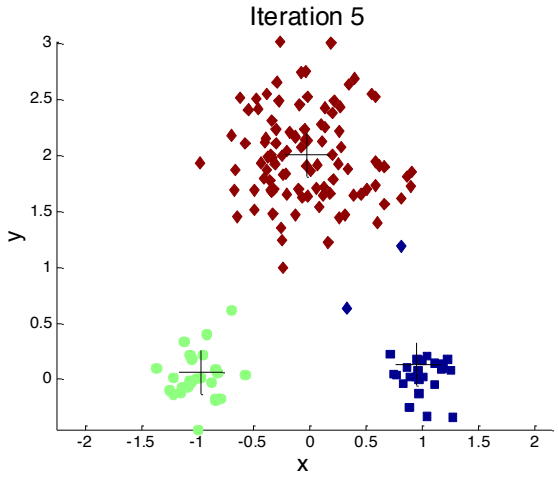
IMPORTANCE OF CHOOSING INITIAL CENTROIDS

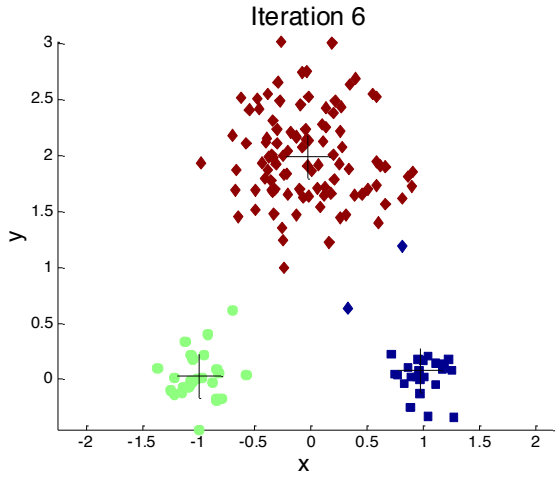




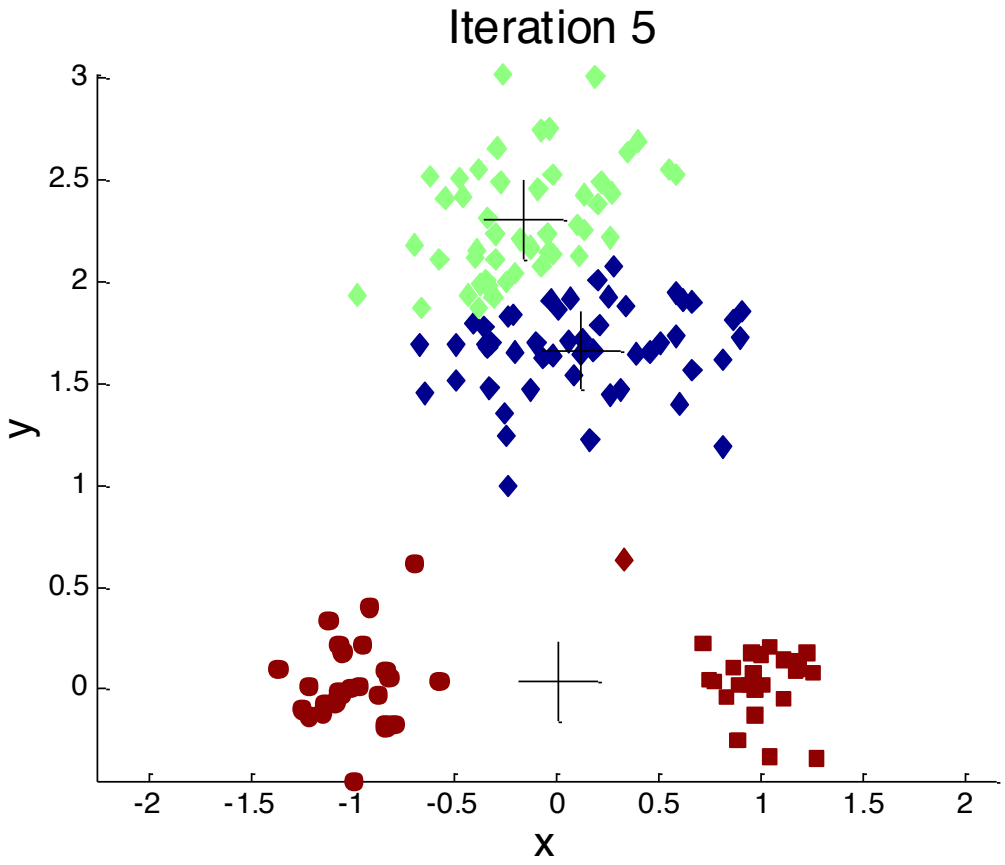






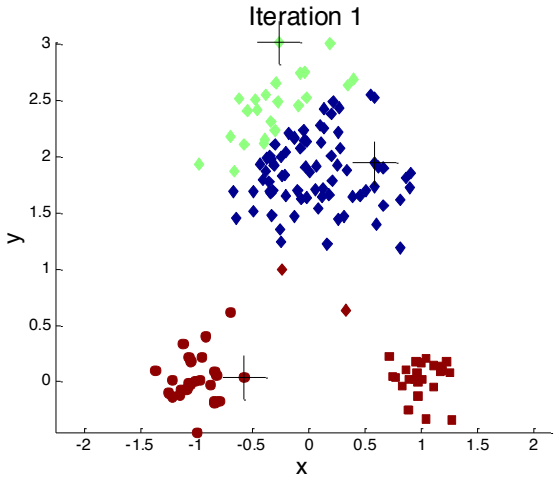


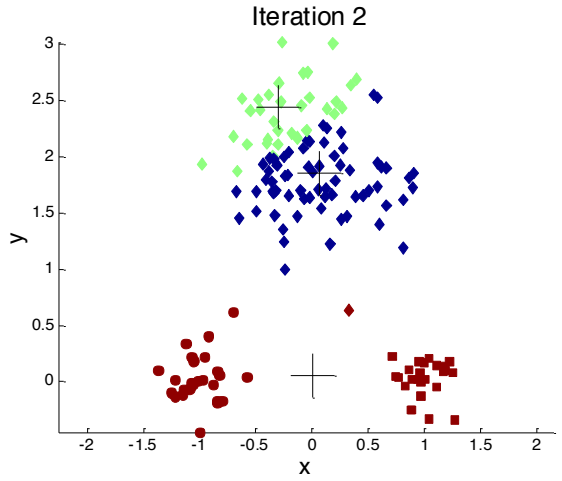
IMPORTANCE OF CHOOSING INITIAL CENTROIDS

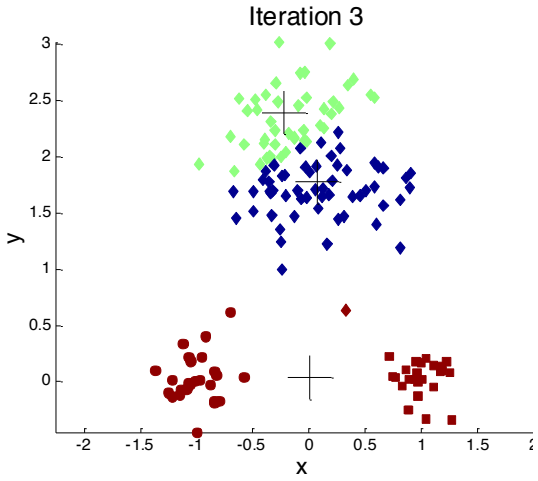


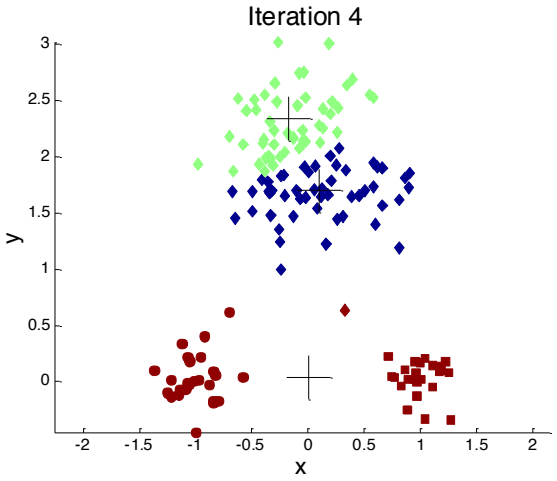
A less meaningful result

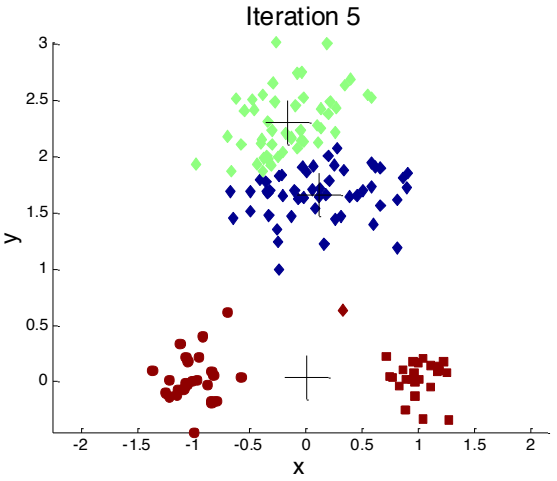
IMPORTANCE OF CHOOSING INITIAL CENTROIDS











TUNING K-MEANS

SYRACUSE UNIVERSITY
School of Information Studies

SOLUTIONS TO INITIAL CENTROIDS PROBLEM

Perform multiple runs, changing random seeds every time.

- 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.

Select most widely separated.

COMPARE SSE OF DIFFERENT INITIAL CENTROIDS

Most common measure is **sum of squared error (SSE).**

x is a data point in cluster C_i and m_i is the centroid or medoid for cluster C_i.

For each point, the error is the distance to the centroid or medoid.

To get SSE, we square these errors and sum them.

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

Given two clustering results with **same** number of clusters, we can choose the one with the smallest SSE.

BE CAREFUL WITH SSE

Attention! One easy way to reduce SSE is to increase *k*, the number of clusters. When *k* equals the data set size, meaning, each data point is in its own cluster, then SSE = 0.

Don't simply use *k* to reduce SSE. *k* should have a reasonable range of value in real applications.

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

WHAT IF THE ITERATION NEVER STOPS?

Set maximum number of iterations.

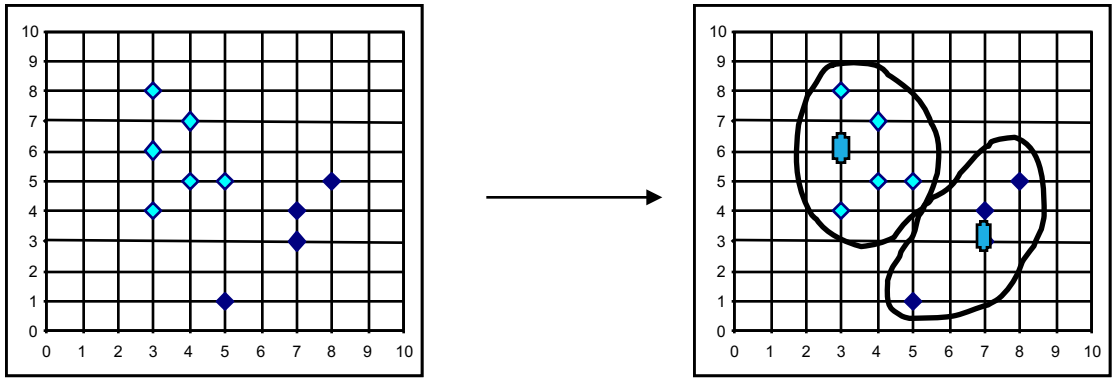
Set minimum value of SSE change.

USE MEDOIDS TO RESIST OUTLIERS IN K-MEANS

The k-means algorithm is sensitive to outliers!

Since an object with an extremely large value may substantially distort the distribution of the data

k-medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.



PAM: A K-MEDOID ALGORITHM

http://www.cs.umb.edu/cs738/pam1.pdf

PAM: Partition Around Medoids

The goal of the algorithm is to minimize the average dissimilarity of objects to their closest selected object. Equivalently, we can minimize the sum of the dissimilarities between object and their closest selected object.

The algorithm has two phases:

(i) In the first phase, BUILD, a collection of k objects are selected for an initial set S.

(ii) In the second phase, SWAP, one tries to improve the quality of the clustering by exchanging selected objects with unselected objects.

VARIATIONS OF THE K-MEANS METHOD

One variation comprises mixture models (soft clustering).

Estimates clusters from probability distributions

Includes the **expectation-maximization** (EM) algorithm

CLUSTER VALIDITY

For supervised classification, we have a variety of measures to evaluate how good our model ism

- Accuracy, precision, recall

For cluster analysis, the analogous question is how to evaluate the "goodness" of the resulting clusters.

But "clusters are in the eye of the beholder"!

DIFFERENT METHODS FOR CLUSTER VALIDATION

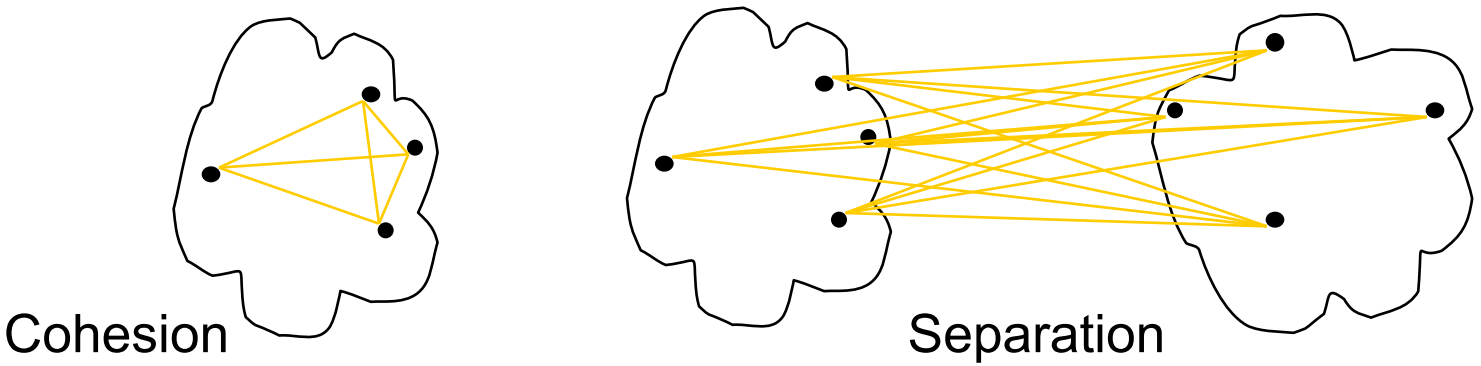
Cluster cohesion: Measures how closely related objects are in a cluster

- High **intraclass** similarity
- SSE as a cohesion measure

Cluster separation: Measure how distinct or well separated a cluster is from other clusters

- Low **interclass** similarity

Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels

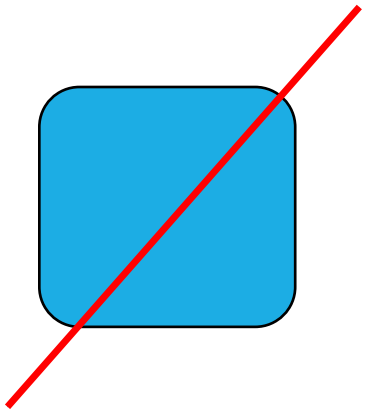


COMMENTS ON THE K-MEANS METHOD

Strength: *Relatively efficient*

Weaknesses:

- Need to specify *k*, the *number* of clusters, in advance
- Unable to handle noisy data and *outliers*
- Not suitable to discover clusters with *nonconvex shapes*



FINAL COMMENT ON CLUSTER VALIDITY

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

Algorithms for Clustering Data (Jain & Dubes)

HAC ALGORITHM

SYRACUSE UNIVERSITY
School of Information Studies

K-MEANS

Process

Problem

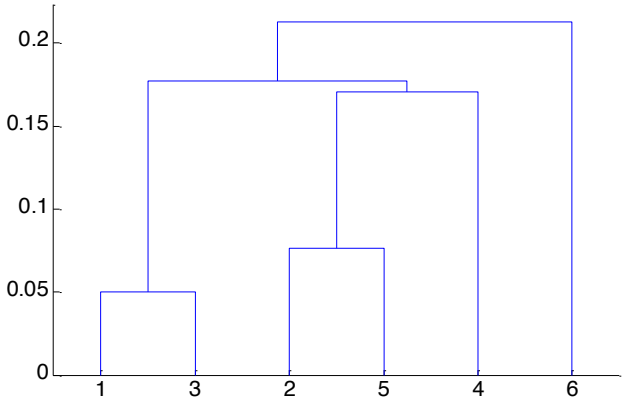
- Fixed number of clusters
- Initial choice of centroid

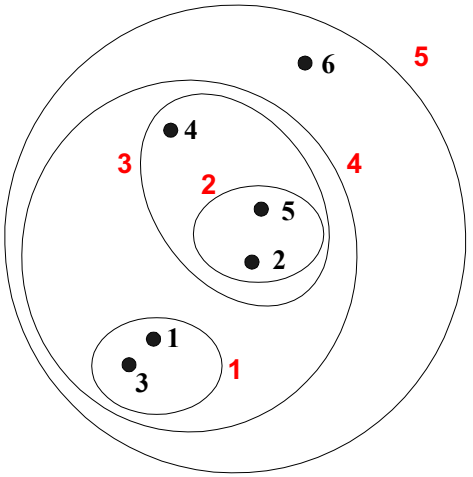
HIERARCHICAL CLUSTERING

Produces a set of nested clusters organized as a hierarchical tree

Can be visualized as a dendrogram:

A treelike diagram that records the sequences of merges or splits

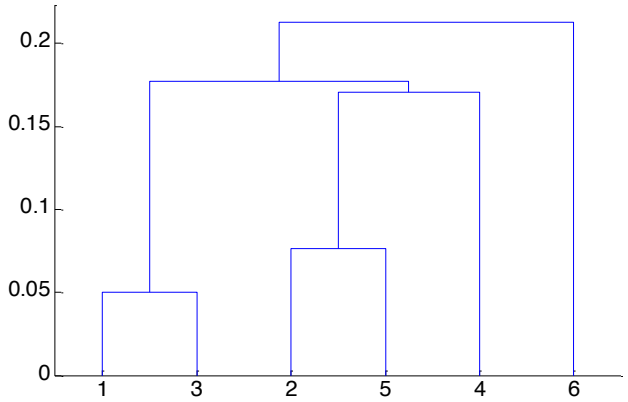




DENDROGRAM: SHOWS HOW THE CLUSTERS ARE MERGED

Decompose data objects into a several levels of nested partitioning (**tree** of clusters), called a **dendrogram**.

A **clustering** of the data objects is obtained by **cutting** the dendrogram at the desired level; then each **connected component** forms a cluster.



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.

AGGLOMERATIVE CLUSTERING ALGORITHM

More popular hierarchical clustering technique:

Basic algorithm is straightforward.

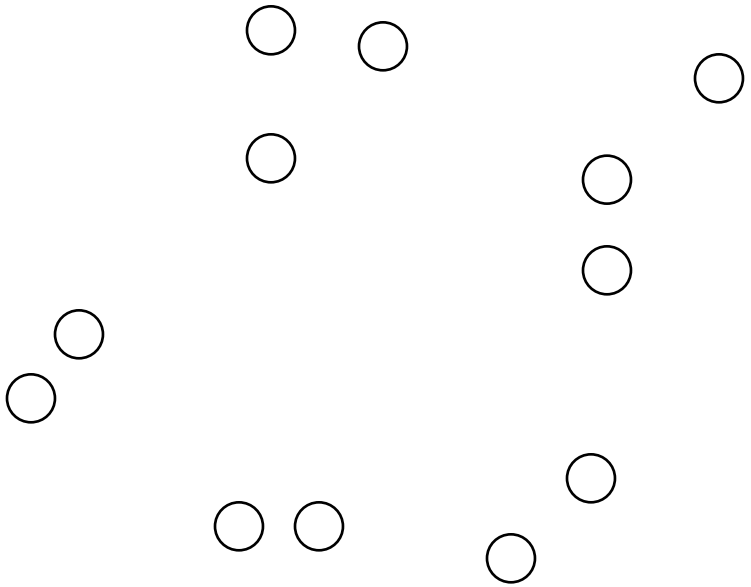
1. Let each data point be a cluster.
2. Compute the distance matrix.
3. **Repeat.**
4. Merge the two closest clusters.
5. Update the distance matrix …
6. **… until** only a single cluster remains.

Key operation is the computation of the distance of two clusters.

Different approaches to defining the distance between clusters distinguish the different algorithms.

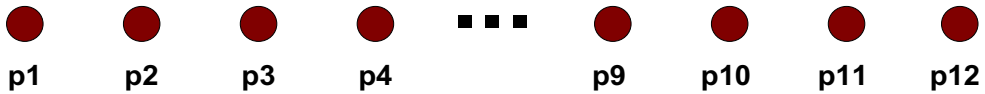
STARTING SITUATION

Start with clusters of individual points and a distance matrix.

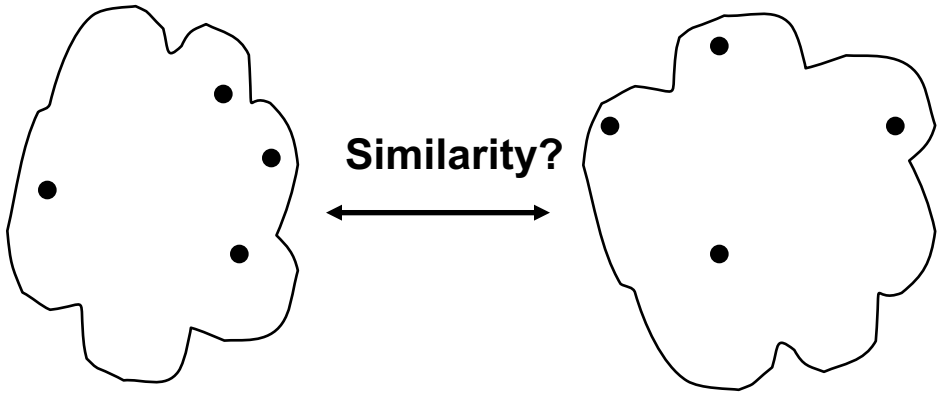


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Distance Matrix



HOW TO DEFINE INTERCLUSTER DISTANCE



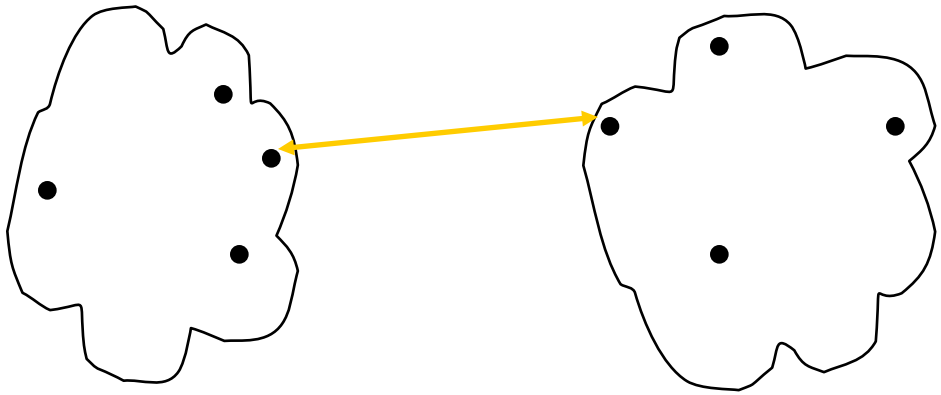
Single linkage

Complete linkage

Average linkage

Centroid linkage

HOW TO DEFINE INTERCLUSTER DISTANCE



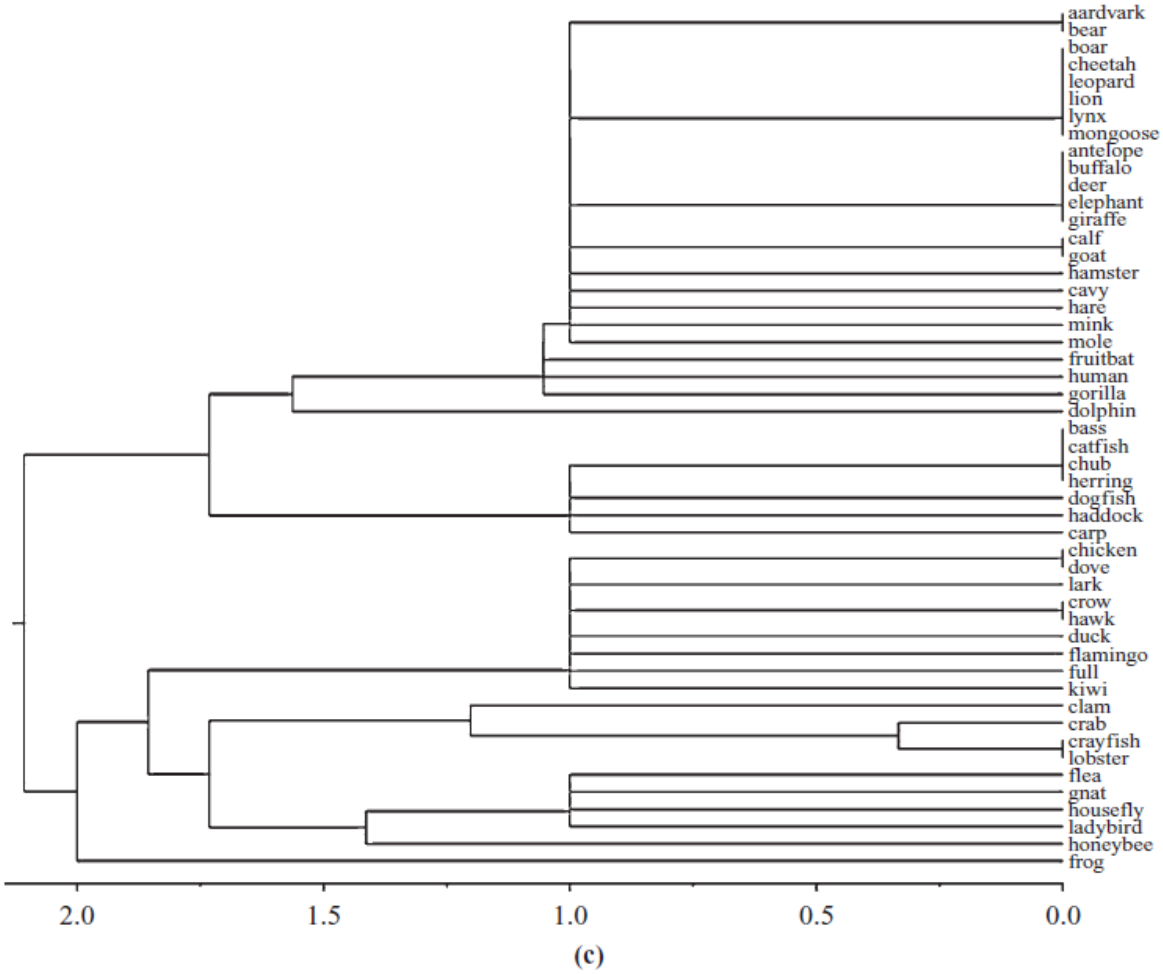
Single linkage:

- Minimum distance between two clusters
- The distance between a pair of closest members

Pros and cons:

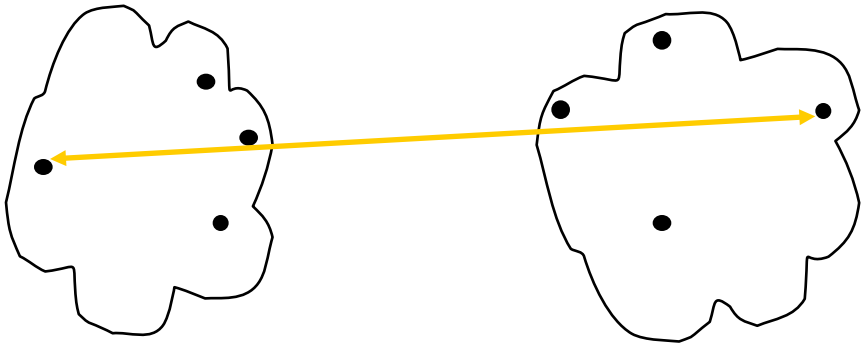
- Depends only on the distance ordering
- Sensitive to outliers
- Clusters with large diameters

SINGLE LINKAGE



(c)

HOW TO DEFINE INTERCLUSTER DISTANCE (CONT.)



Complete linkage:

- Maximum distance between clusters
- The distance between a pair of farthest members

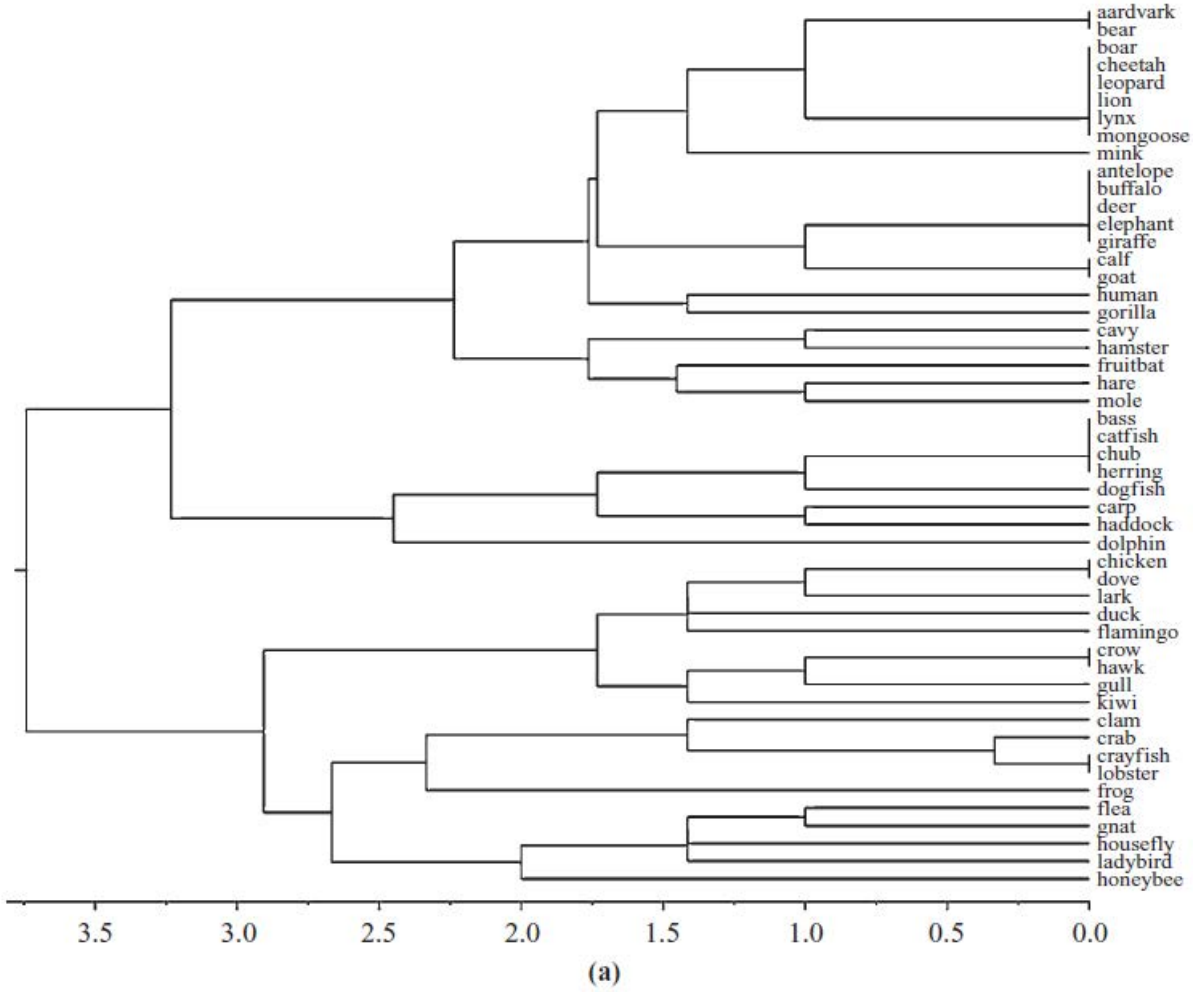
Pros and cons:

- Depends only on the distance ordering
- Sensitive to outliers
- Clusters with small diameters

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

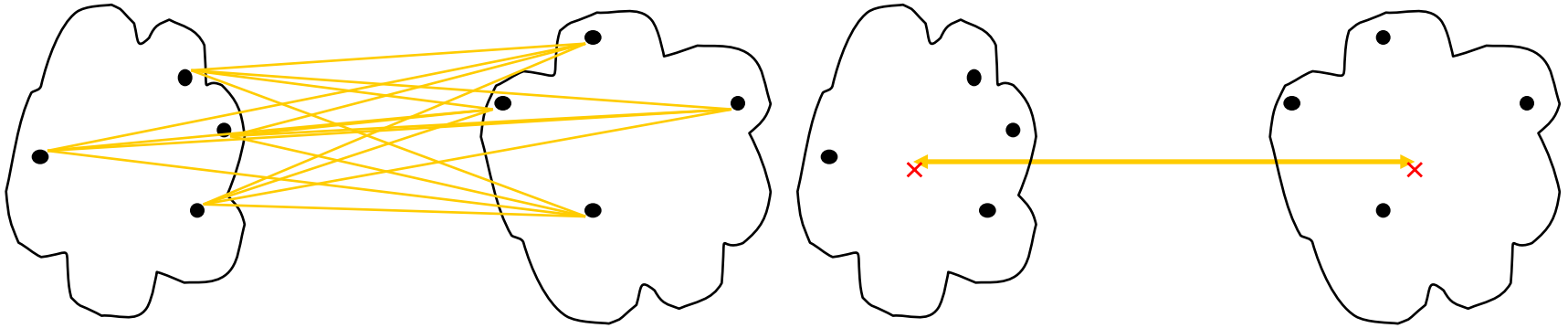
Distance Matrix

COMPLETE LINKAGE



(a)

HOW TO DEFINE INTERCLUSTER DISTANCE (CONT.)



To overcome sensitivity to outliers –

Average linkage: The average distance between each pair of members of the two clusters

Centroid linkage: Distance between two centroids

HIERARCHICAL CLUSTERING: PROBLEMS AND LIMITATIONS

Once a decision is made to combine two clusters, it cannot be undone.

No objective function is directly minimized.

Different linkage calculations have problems with one or more of the following:

- Sensitivity to noise and outliers
- Difficulty handling different-sized clusters and convex shapes
- Breaking large clusters