

CS 780

Data Mining for Multimedia Data

Dr. Jessica Lin

Clustering

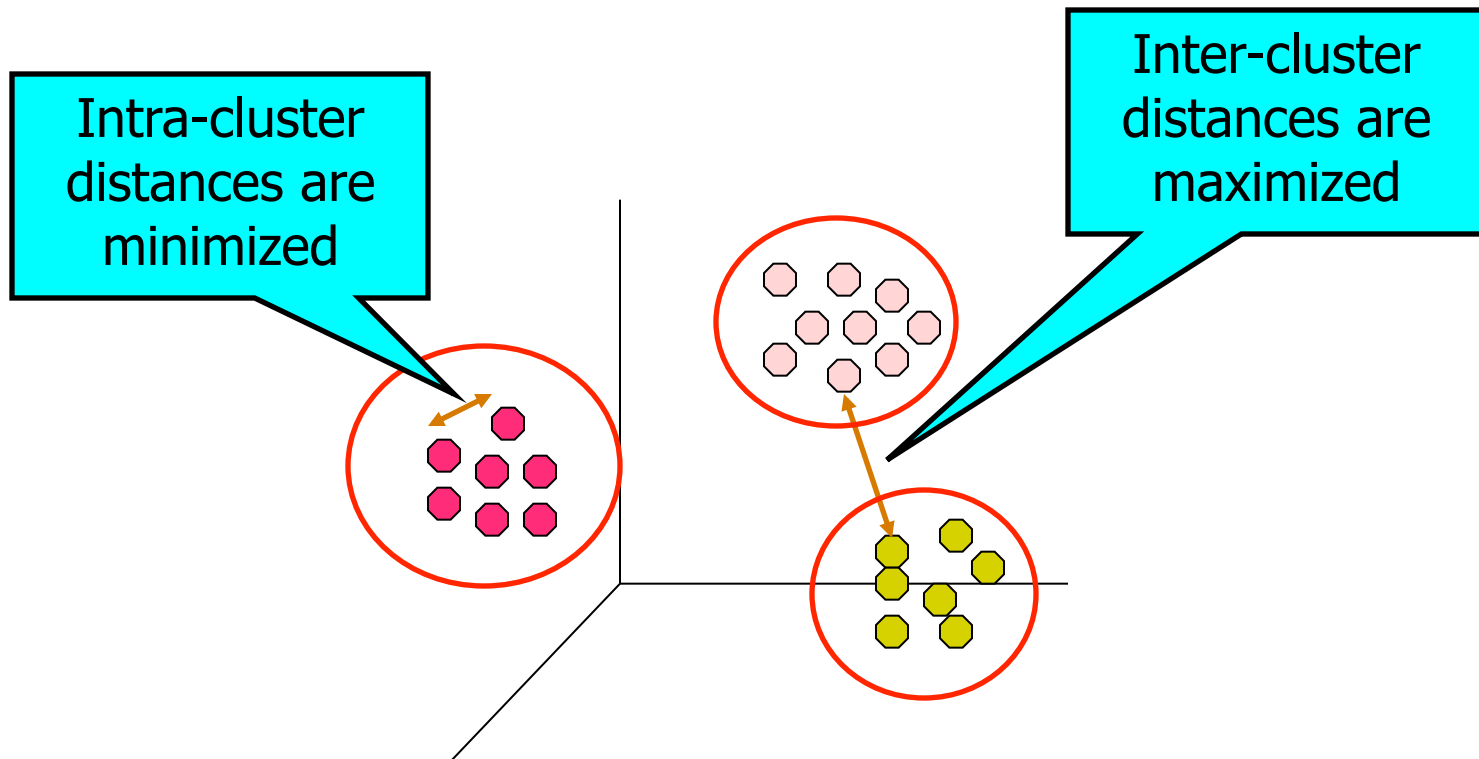
A thick, wavy orange line that spans the width of the slide, positioned below the title 'Clustering'.

Clustering

- Given a set of data points, each having a set of attributes, and a similarity measure among them, find clusters such that
 - ★ Data points in one cluster are more similar to one another.
 - ★ Data points in separate clusters are less similar to one another.
- More informally, finding natural groupings among objects. (i.e. east coast cities, west coast cities)
- Similarity Measures:
 - ★ Euclidean Distance if attributes are continuous.
 - ★ Other Problem-specific Measures.

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



Applications of Cluster Analysis

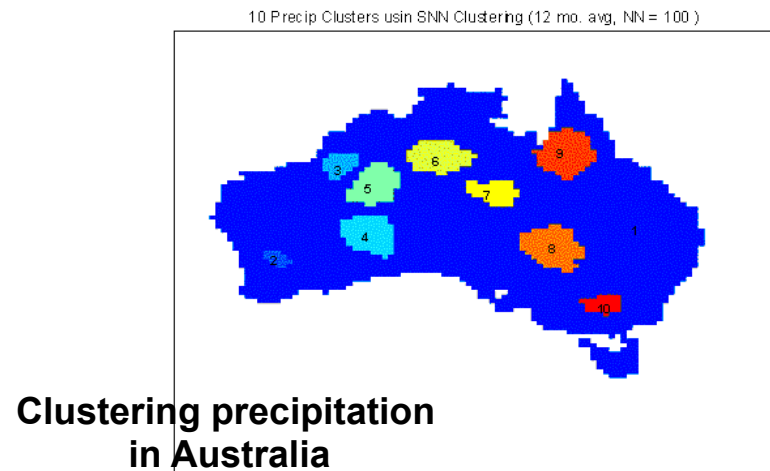
■ Understanding

- ★ Group related documents for browsing, group genes and proteins that have similar functionality, or group stocks with similar price fluctuations

	<i>Discovered Clusters</i>	<i>Industry Group</i>
1	Applied-Matl-DOWN,Bay-Network-DOWN,3-COM-DOWN,Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN,DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN,Micron-Tech-DOWN,Texas-Inst-DOWN,Tellabs-Inc-DOWN,Natl-Semiconduct-DOWN,OracI-DOWN,SGL-DOWN,Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN,ADV-Micro-Device-DOWN,Andrew-Corp-DOWN,Computer-Assoc-DOWN,Circuit-City-DOWN,Compaq-DOWN,EMC-Corp-DOWN,Gen-Inst-DOWN,Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN,MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP,Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP,Schlumberger-UP	Oil-UP

■ Summarization

- ★ Reduce the size of large data sets



Illustrating Document Clustering

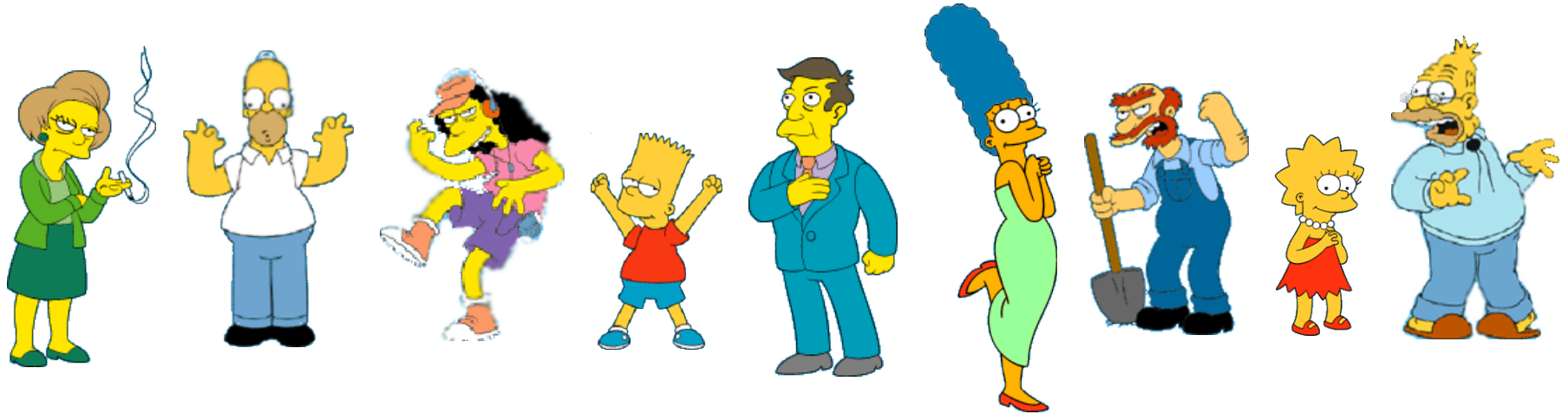
- Clustering Points: 3204 Articles of Los Angeles Times.
- Similarity Measure: How many words are common in these documents (after some word filtering).

<i>Category</i>	<i>Total Articles</i>	<i>Correctly Placed</i>
<i>Financial</i>	555	364
<i>Foreign</i>	341	260
<i>National</i>	273	36
<i>Metro</i>	943	746
<i>Sports</i>	738	573
<i>Entertainment</i>	354	278

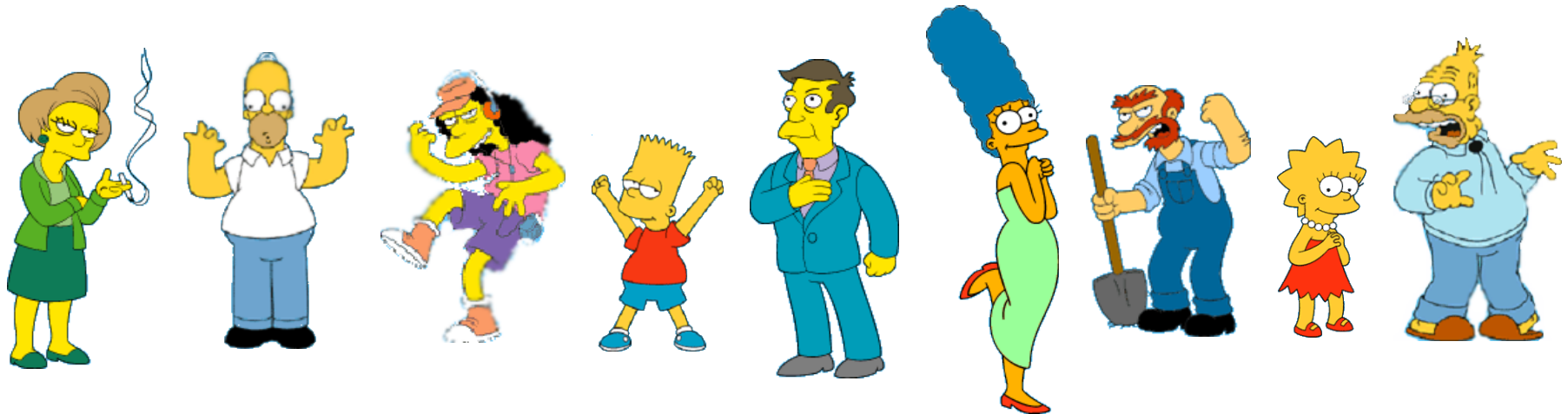
Think point ?

- Differences between classification and clustering?

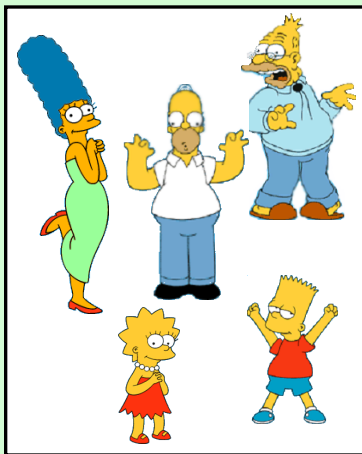
What is a natural grouping among these objects?



What is a natural grouping among these objects?



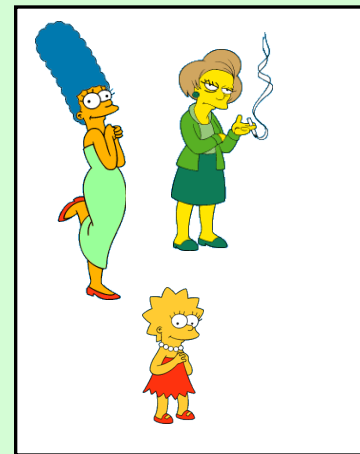
Clustering is subjective



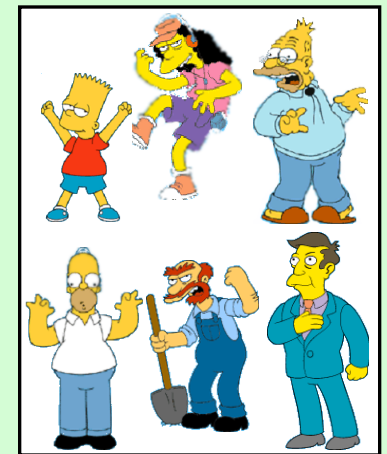
Simpson's Family



School Employees



Females

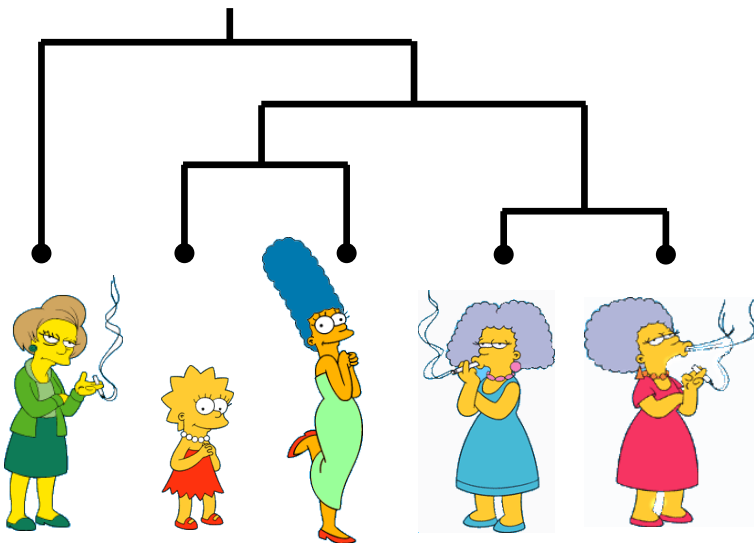


Males

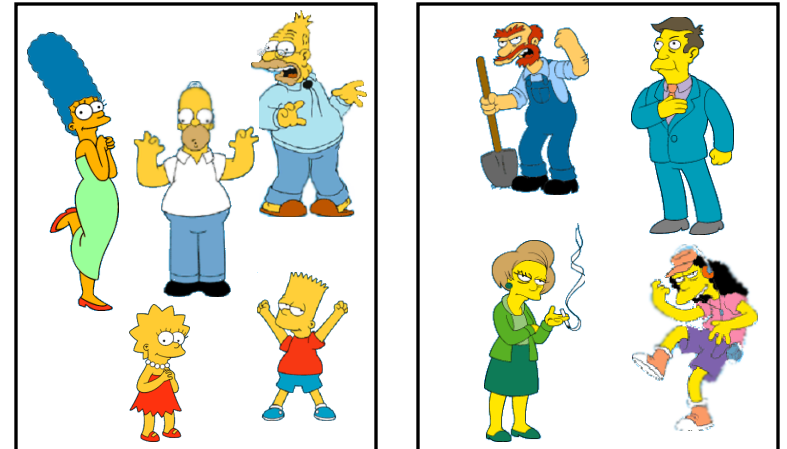
Two Types of Clustering

- **Partitional algorithms:** Construct various partitions and then evaluate them by some criterion
- **Hierarchical algorithms:** Create a hierarchical decomposition of the set of objects using some criterion

Hierarchical

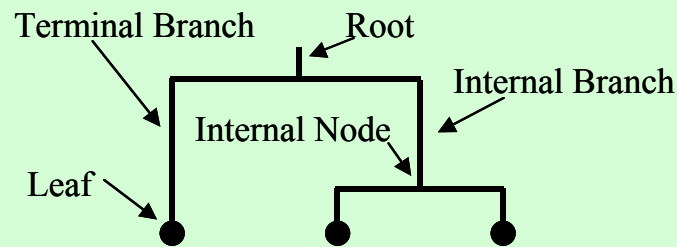


Partitional

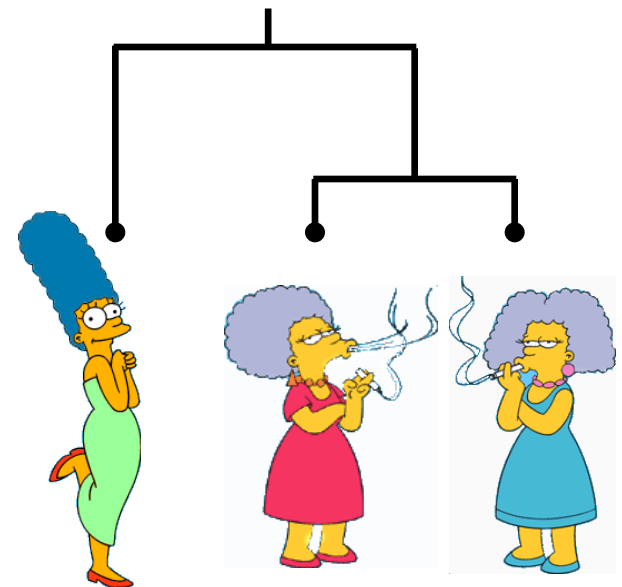
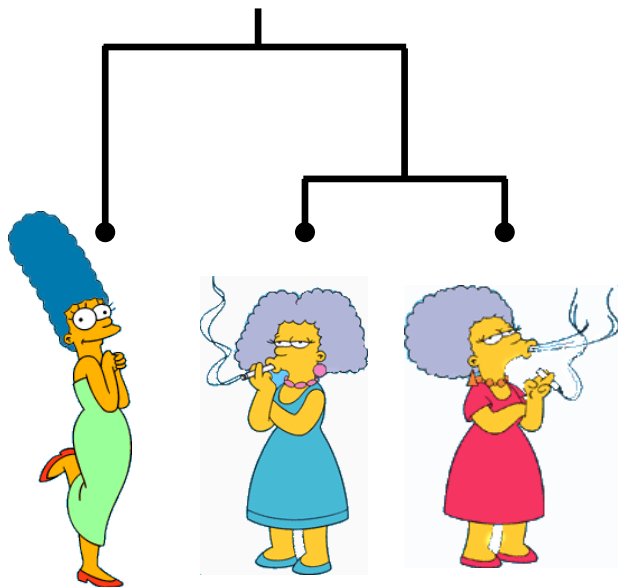


A Useful Tool for Summarizing Similarity Measurements

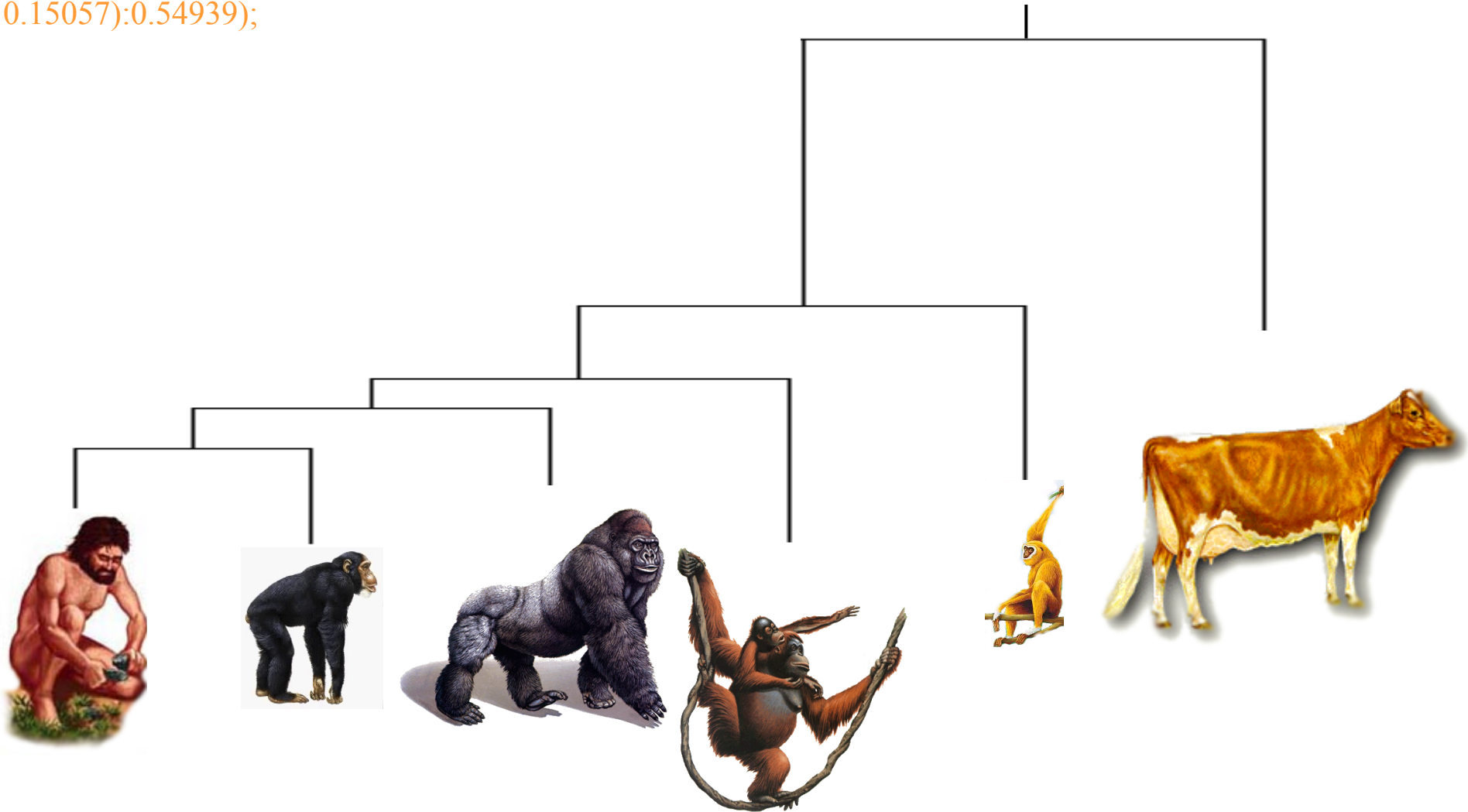
In order to better appreciate and evaluate the examples given in the early part of this talk, we will now introduce the *dendrogram*.



The similarity between two objects in a dendrogram is represented as the height of the lowest internal node they share.



(Bovine:0.69395,(Gibbon:0.36079,(Orangutan:
0.33636,(Gorilla:0.17147,(Chimp:
0.19268,Human:0.11927):0.08386):0.06124):
0.15057):0.54939);



Note that hierarchies are commonly used to organize information, for example in a web portal.

Yahoo's hierarchy is manually created, we will focus on automatic creation of hierarchies in data mining.

Web Site Directory - Sites organized by subject

[Suggest your site](#)

Business & Economy

[B2B](#), [Finance](#), [Shopping](#), [Jobs](#)...

Regional

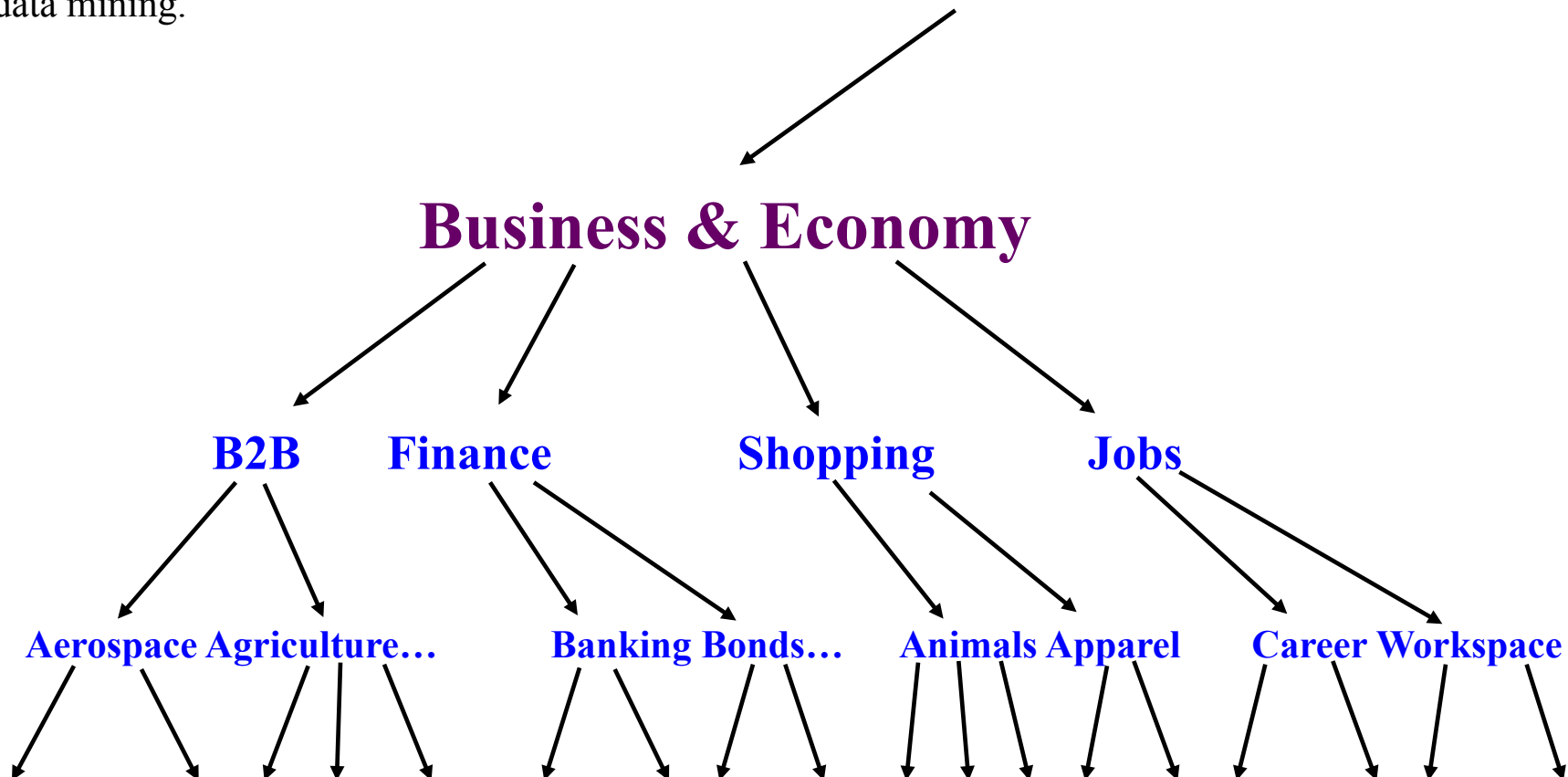
[Countries](#), [Regions](#), [US States](#)...

Computers & Internet

[Internet](#), [WWW](#), [Software](#), [Games](#)...

Society & Culture

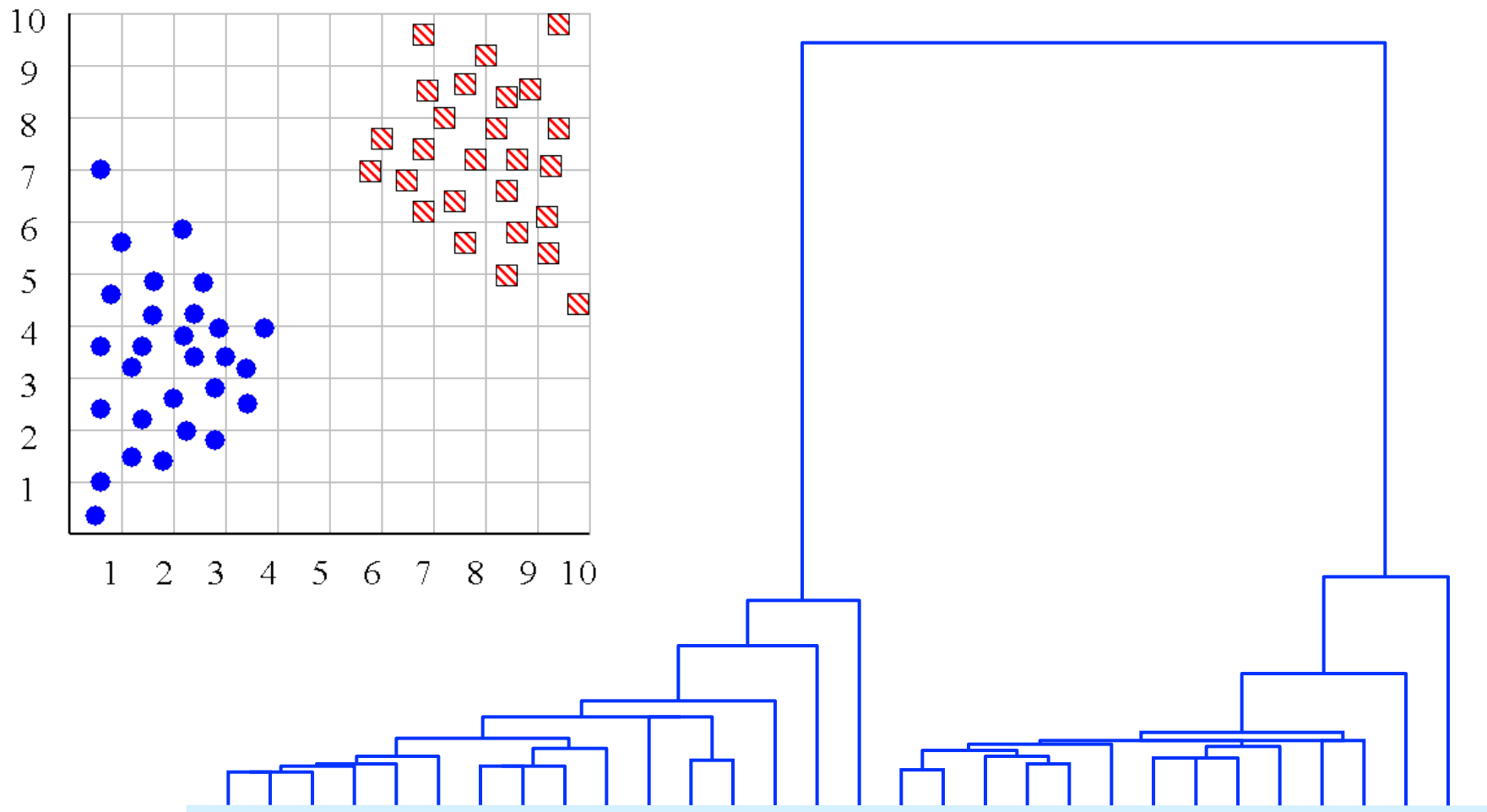
[People](#), [Environment](#), [Religion](#)...



Desirable Properties of a Clustering Algorithm

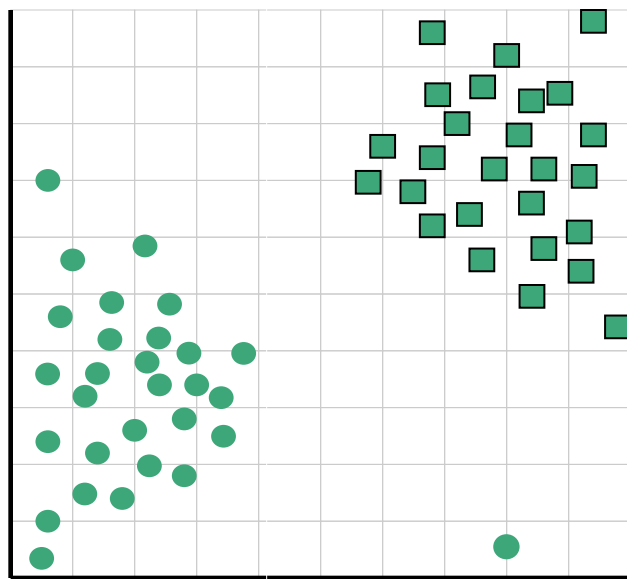
- Scalability (in terms of both time and space)
- Ability to deal with different data types
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- Incorporation of user-specified constraints
- Interpretability and usability

We can look at the dendrogram to determine the “correct” number of clusters. In this case, the two highly separated subtrees are highly suggestive of two clusters. (Things are rarely this clear cut, unfortunately)

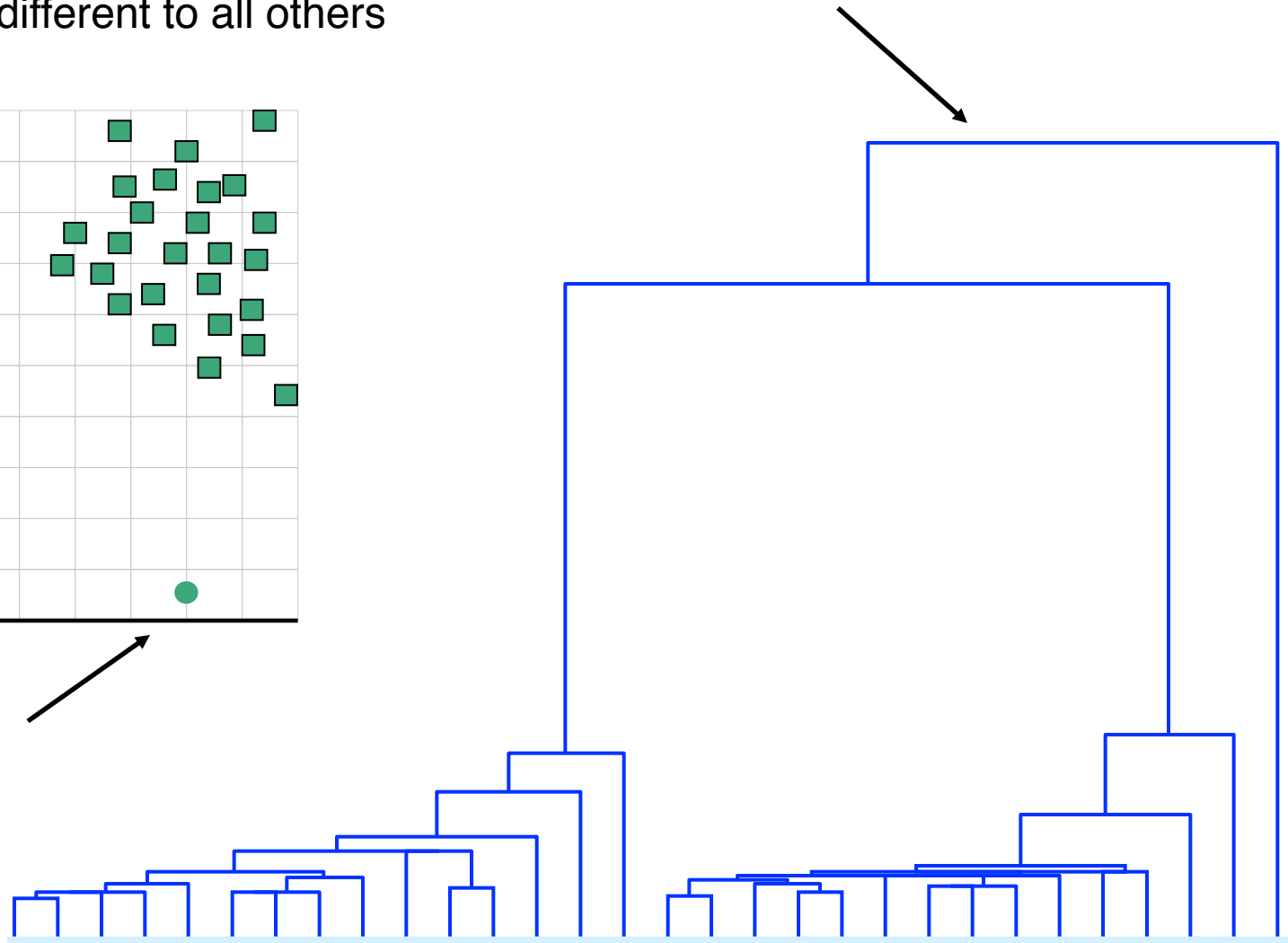


One potential use of a dendrogram is to detect outliers

The single isolated branch is suggestive of a data point that is very different to all others



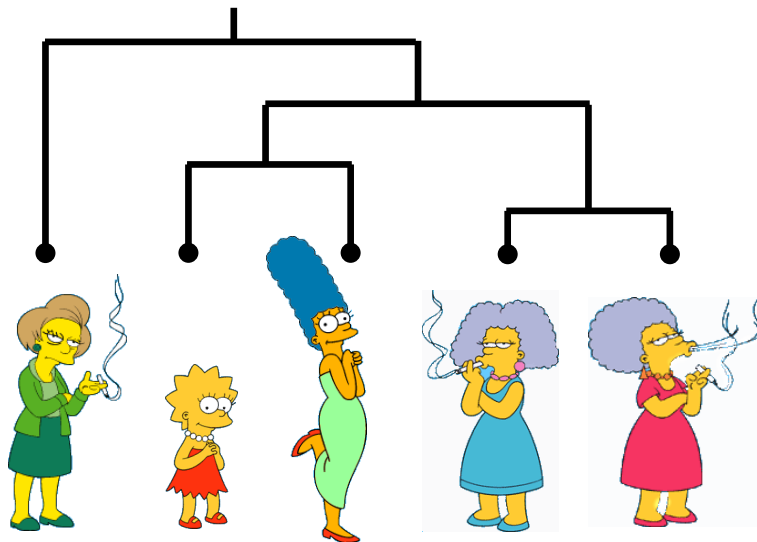
Outlier



Hierarchical Clustering

The number of dendrograms with n leafs = $(2n - 3)! / [(2^{n-2}) (n - 2)!]$

Number of Leafs	Number of Possible Dendrograms
2	1
3	3
4	15
5	105
...	...
10	34,459,425



Since we cannot test all possible trees we will have to heuristic search of all possible trees. We could do this..

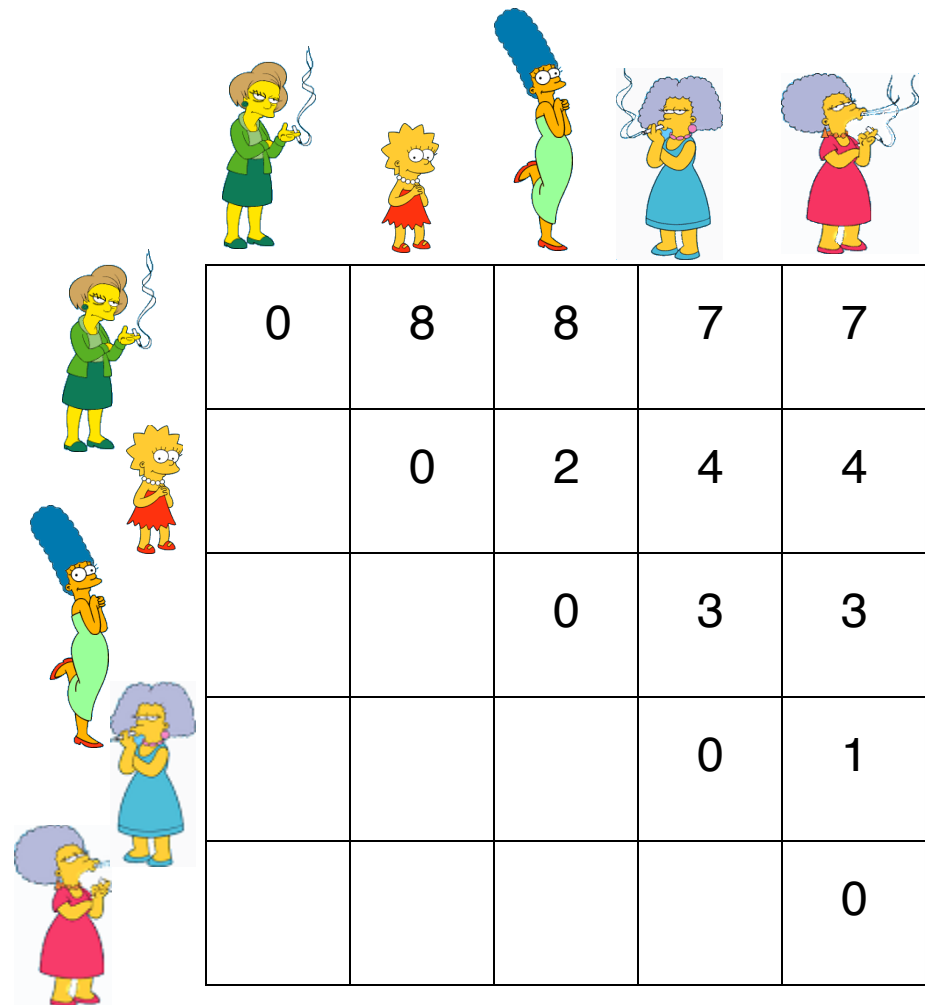
Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.











Top-Down (divisive): Starting with all the data in a single cluster, consider every possible way to divide the cluster into two. Choose the best division and recursively operate on both sides.

We begin with a distance matrix which contains the distances between every pair of objects in our database.

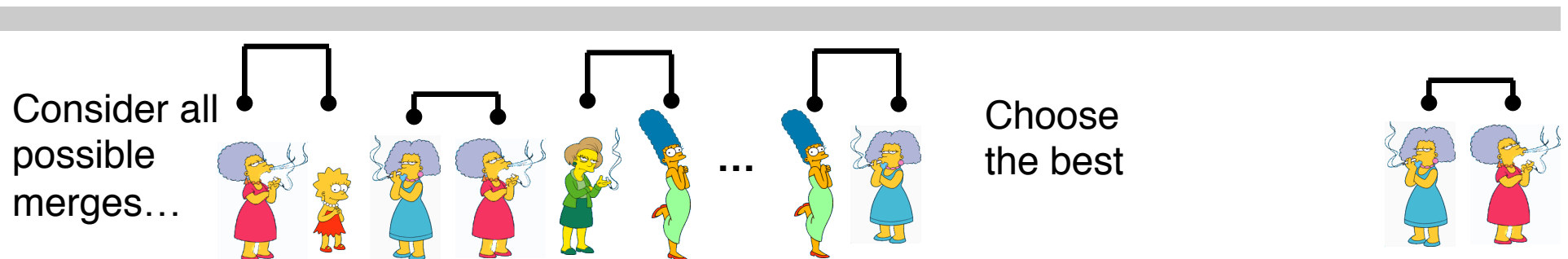
$$D(\text{Mrs. Simpson, Lisa Simpson}) = 8$$

$$D(\text{Marge Simpson, Lisa Simpson}) = 1$$

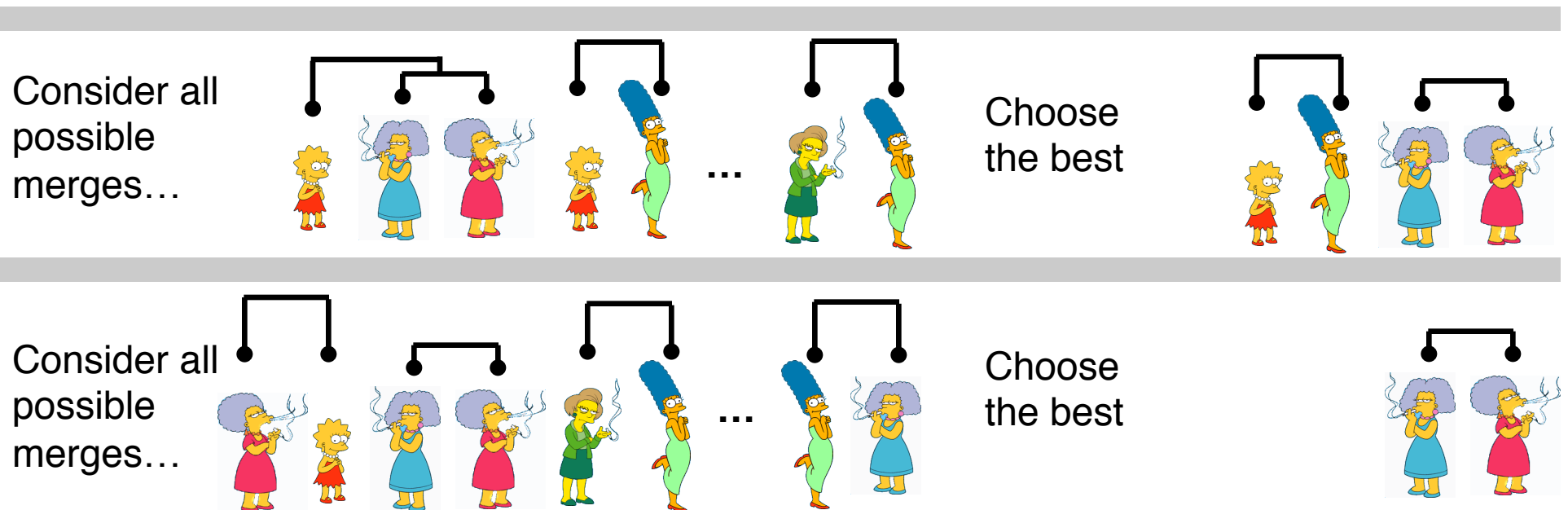


					
	0	8	8	7	7
		0	2	4	4
			0	3	3
				0	1
					0

Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

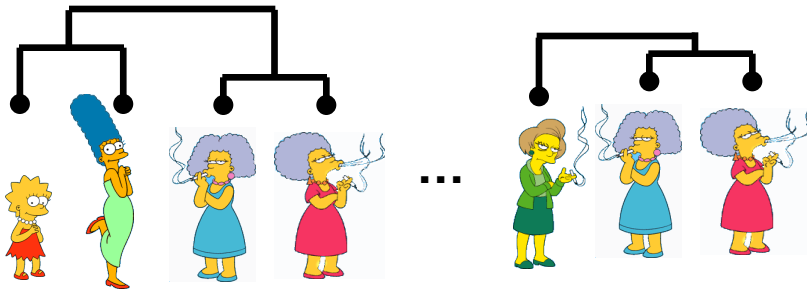


Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

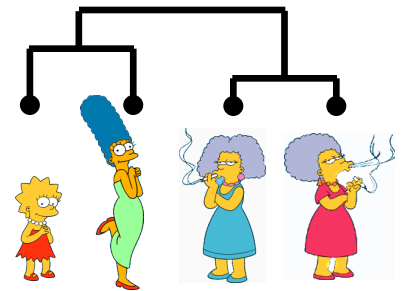


Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

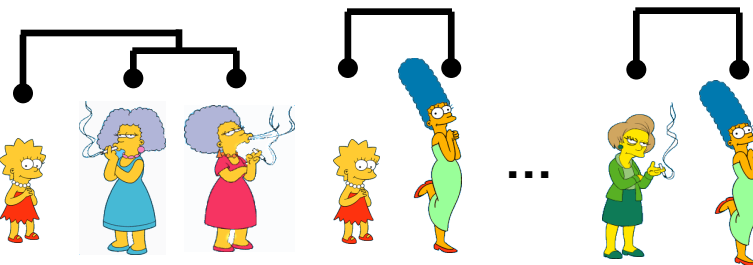
Consider all possible merges...



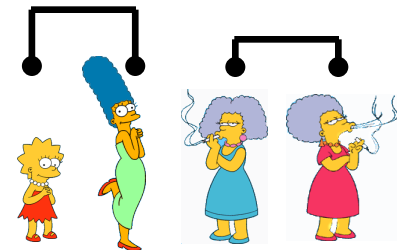
Choose the best



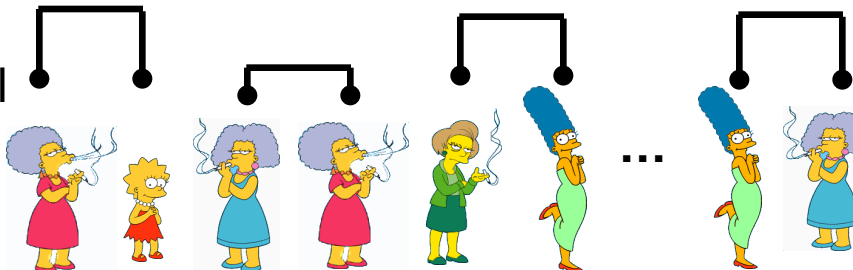
Consider all possible merges...



Choose the best



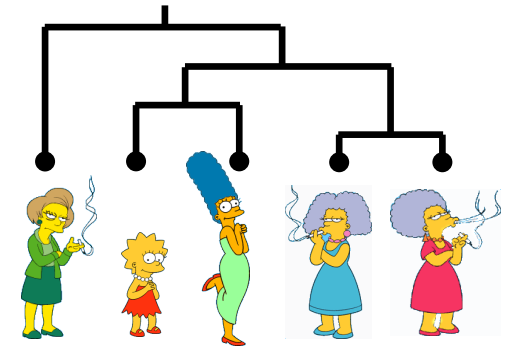
Consider all possible merges...



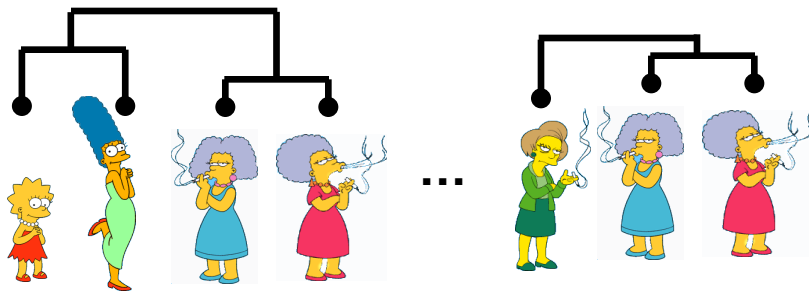
Choose the best



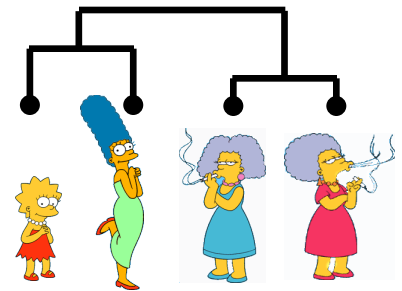
Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



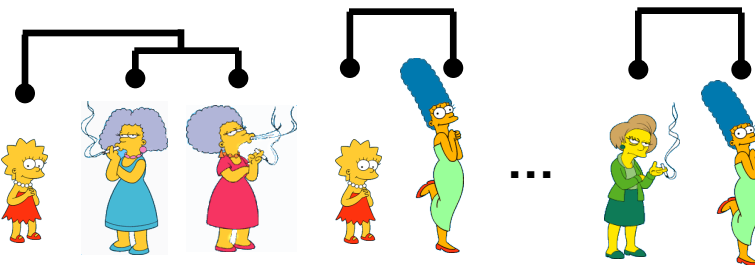
Consider all possible merges...



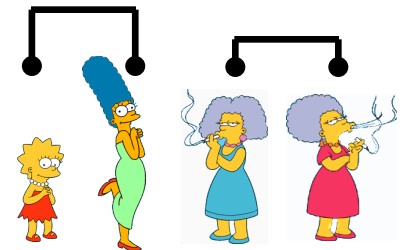
Choose the best



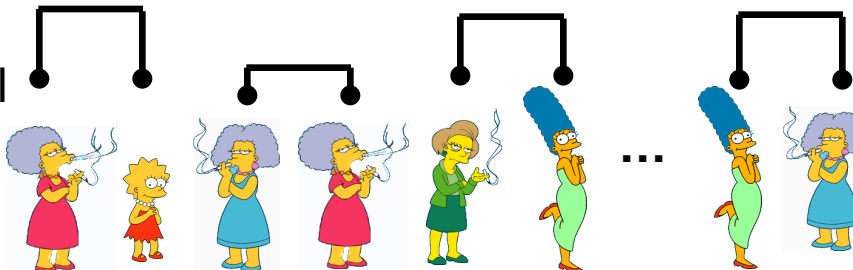
Consider all possible merges...



Choose the best



Consider all possible merges...

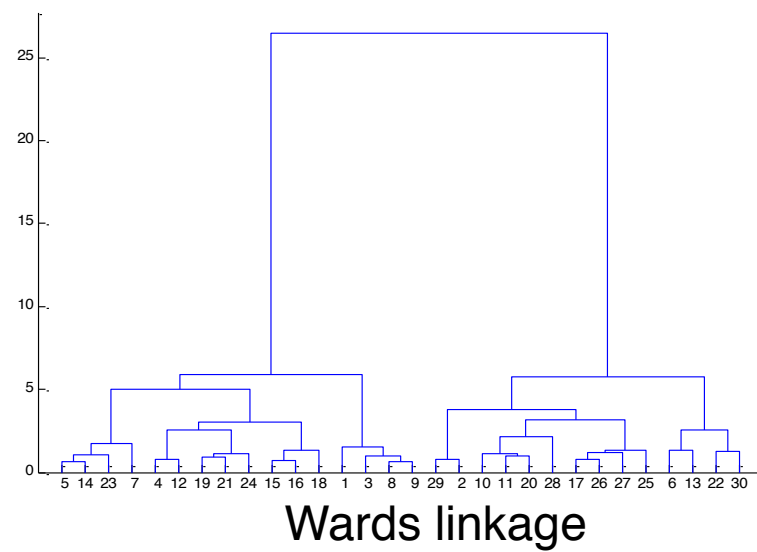
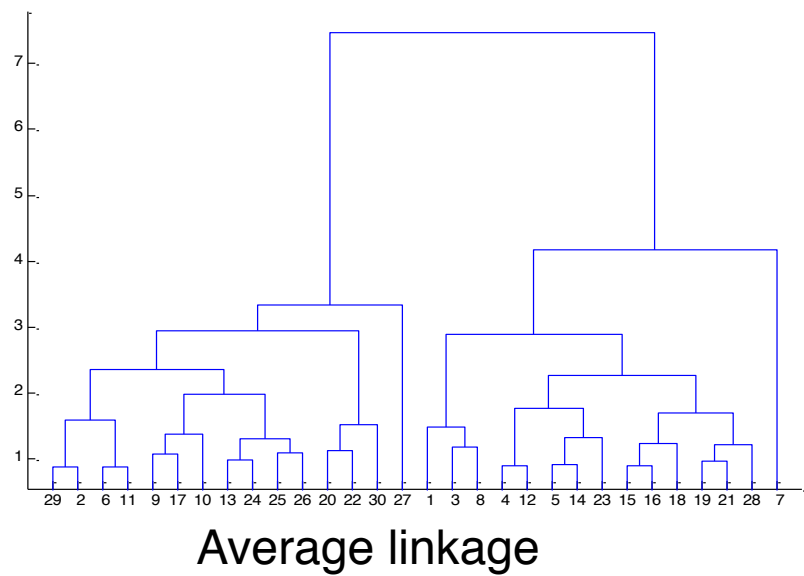
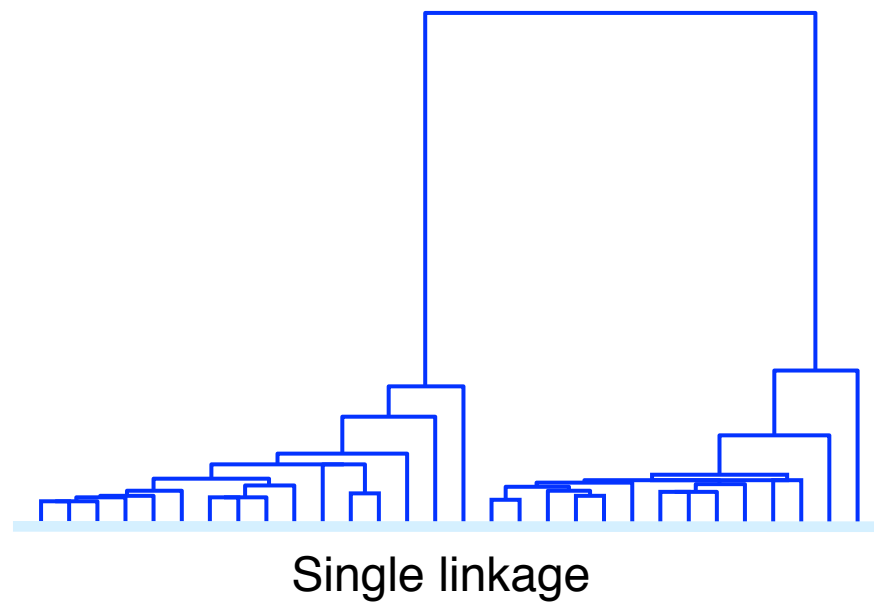
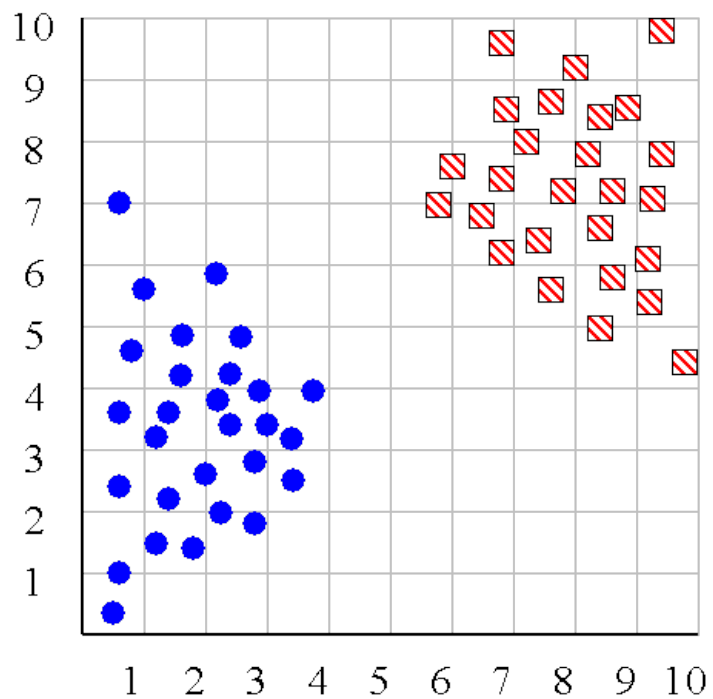


Choose the best



We know how to measure the distance between two objects, but defining the distance between an object and a cluster, or defining the distance between two clusters is non obvious.

- **Single linkage (nearest neighbor):** In this method the distance between two clusters is determined by the distance of the two closest objects (nearest neighbors) in the different clusters.
- **Complete linkage (furthest neighbor):** In this method, the distances between clusters are determined by the greatest distance between any two objects in the different clusters (i.e., by the "furthest neighbors").
- **Group average linkage:** In this method, the distance between two clusters is calculated as the average distance between all pairs of objects in the two different clusters.
- **Wards Linkage:** In this method, we try to minimize the variance of the merged clusters

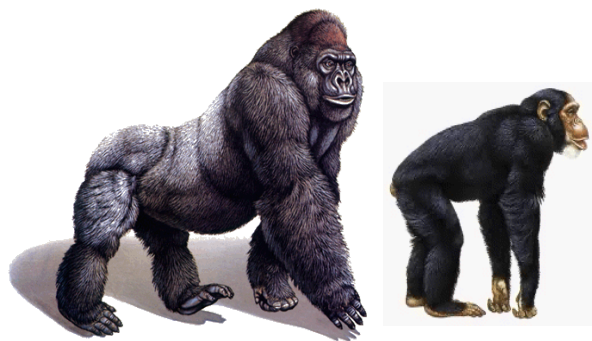


Summary of Hierarchical Clustering Methods

- No need to specify the number of clusters in advance.
- Hierarchical nature maps nicely onto human intuition for some domains
- They do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects.
- Like any heuristic search algorithms, local optima are a problem.
- Interpretation of results is (very) subjective.

Up to this point we have simply assumed that we
can measure similarity, but

How do we measure similarity?



0.23

Peter Piotr



3



342.7

What is Similarity?

The quality or state of being similar; likeness; resemblance; as, a similarity of features.

Webster's Dictionary



Similarity is hard to define, but...
"We know it when we see it"

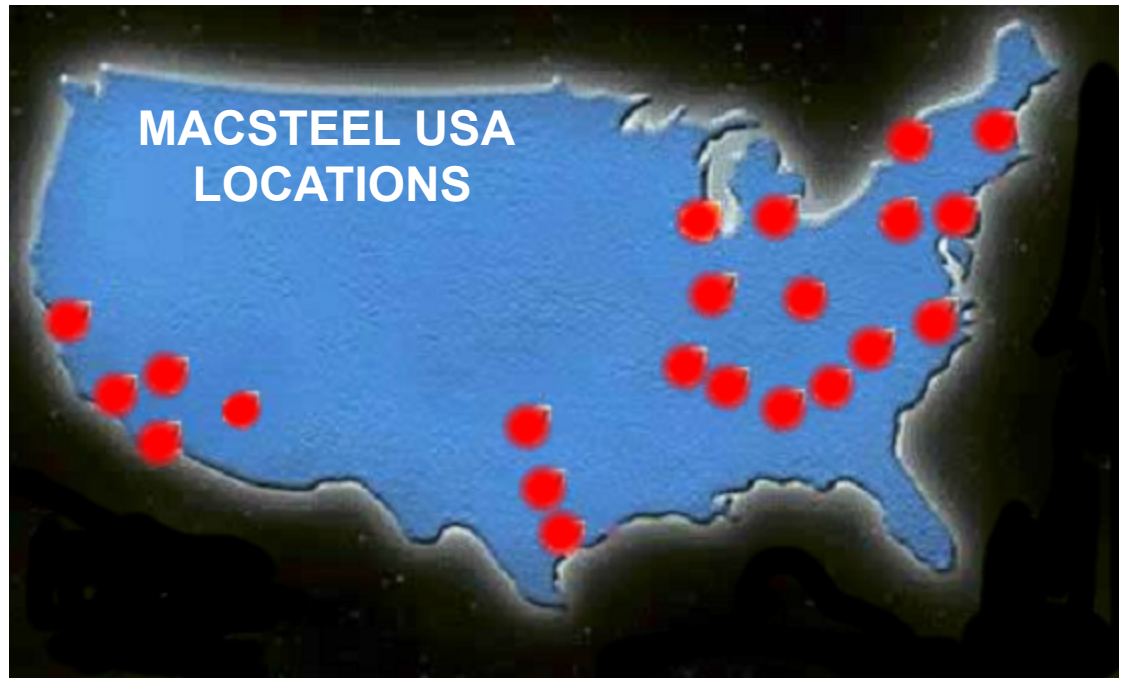
The real meaning of similarity is a philosophical question. We will take a more pragmatic approach.

Similarity Measures

For the moment assume that we can measure the similarity between any two objects. (we will cover this in detail later).

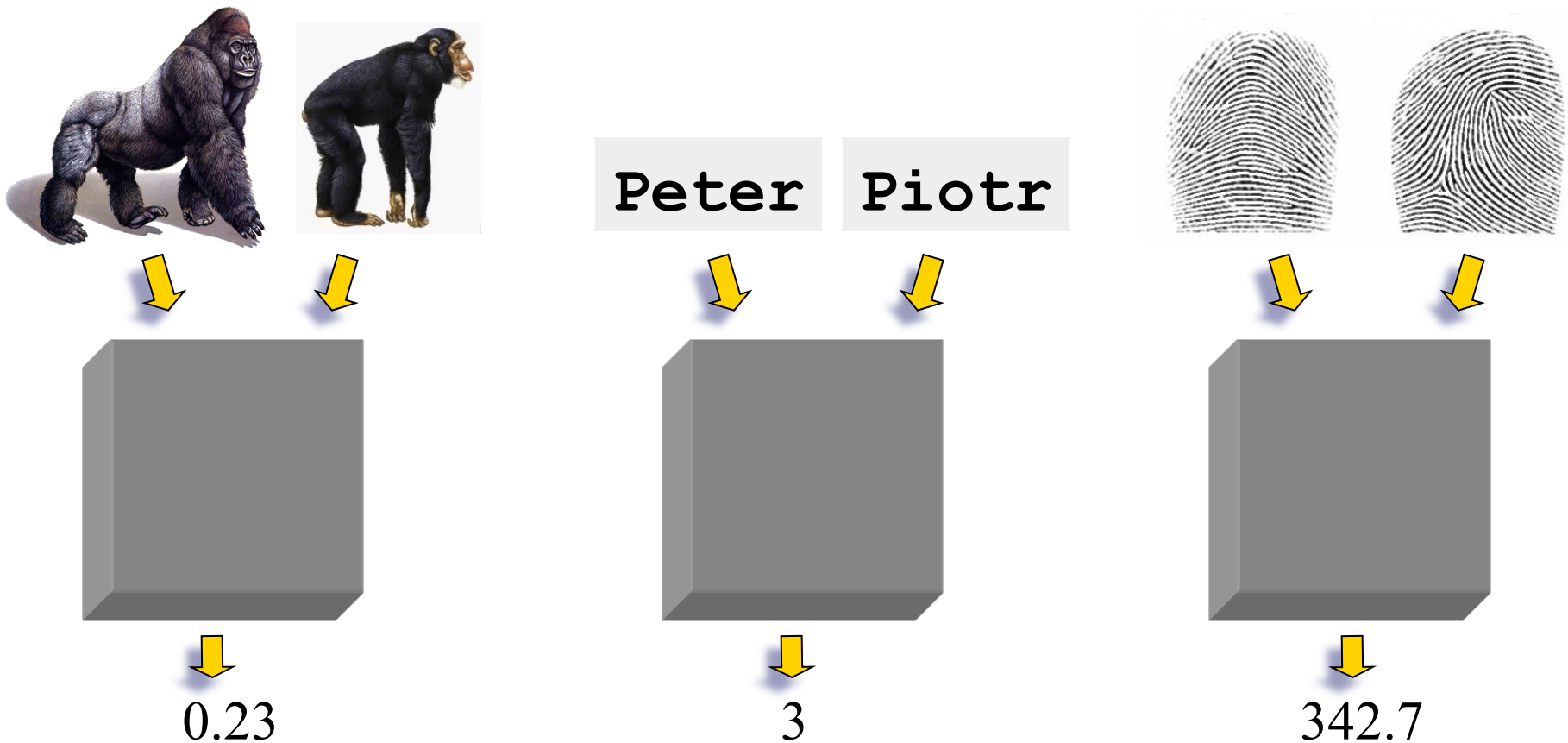
One intuitive example is to measure the distance between two cities and call it the **similarity (or rather, *dissimilarity*)**. For example we have $D(\text{LA}, \text{San Diego}) = 110$, and $D(\text{LA}, \text{New York}) = 3,000$.

This would allow use to make (subjectively correct) statements like “LA is more similar to San Francisco than it is to New York”.



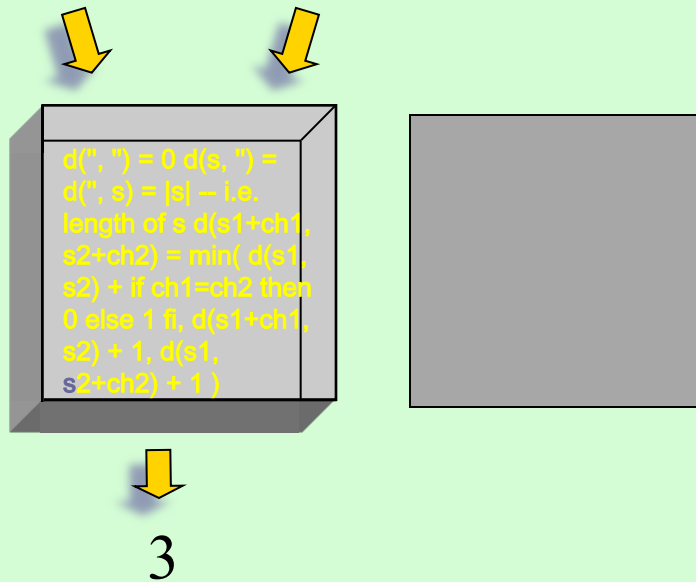
Defining Distance Measures

Definition: Let O_1 and O_2 be two objects from the universe of possible objects. The distance (dissimilarity) between O_1 and O_2 is a real number denoted by $D(O_1, O_2)$



Peter

Piotr



When we peek inside one of these black boxes, we see some function on two variables. These functions might very simple or very complex.

In either case it is natural to ask, what properties should these functions have?

What properties should a distance measure have?

- $D(A, B) = D(B, A)$

Symmetry

- $D(A, A) = 0$

Constancy of Self-Similarity

- $D(A, B) = 0$ iff $A = B$

Positivity (Separation)

- $D(A, B) \leq D(A, C) + D(B, C)$

Triangular Inequality

Intuitions behind desirable distance measure properties

$$D(A,B) = D(B,A)$$

Symmetry

Otherwise you could claim “Alex looks like Bob, but Bob looks nothing like Alex.”

$$D(A,A) = 0$$

Constancy of Self-Similarity

Otherwise you could claim “Alex looks more like Bob, than Bob does.”

$$D(A,B) = 0 \text{ iff } A=B$$

Positivity (Separation)

Otherwise there are objects in your world that are different, but you cannot tell apart.

$$D(A,B) \leq D(A,C) + D(B,C)$$

Triangular Inequality

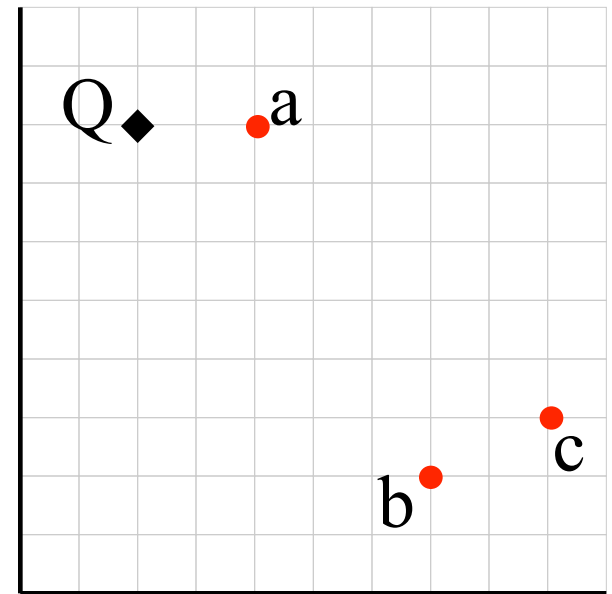
Otherwise you could claim “Alex is very like Bob, and Alex is very like Carl, but Bob is very unlike Carl.”

Why is the Triangular Inequality so Important?

Virtually all techniques to index data require the triangular inequality to hold.

Suppose I am looking for the closest point to Q, in a database of 3 objects.

Further suppose that the triangular inequality holds, and that we have precompiled a table of distance between all the items in the database.



	a	b	c
a		6.70	7.07
b			2.30
c			

Why is the Triangular Inequality so Important?

Virtually all techniques to index data require the triangular inequality to hold.
I find **a** and calculate that it is 2 units from Q,
it becomes my *best-so-far*. I find **b** and
calculate that it is **7.81** units away from Q.

I don't have to calculate the distance from Q
to **c**!

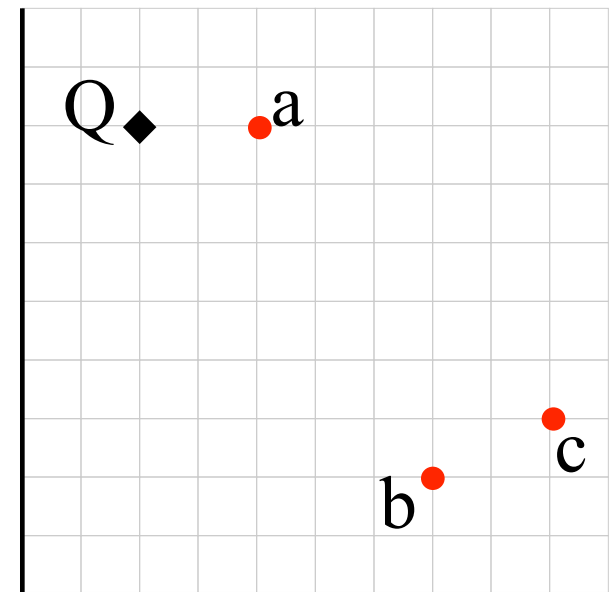
I know $D(Q, \mathbf{b}) \leq D(Q, \mathbf{c}) + D(\mathbf{b}, \mathbf{c})$

$$D(Q, \mathbf{b}) - D(\mathbf{b}, \mathbf{c}) \leq D(Q, \mathbf{c})$$

$$7.81 - 2.30 \leq D(Q, \mathbf{c})$$

$$5.51 \leq D(Q, \mathbf{c})$$

So I know that **c** is at least 5.51 units away,
but my *best so far* is only 2 units away

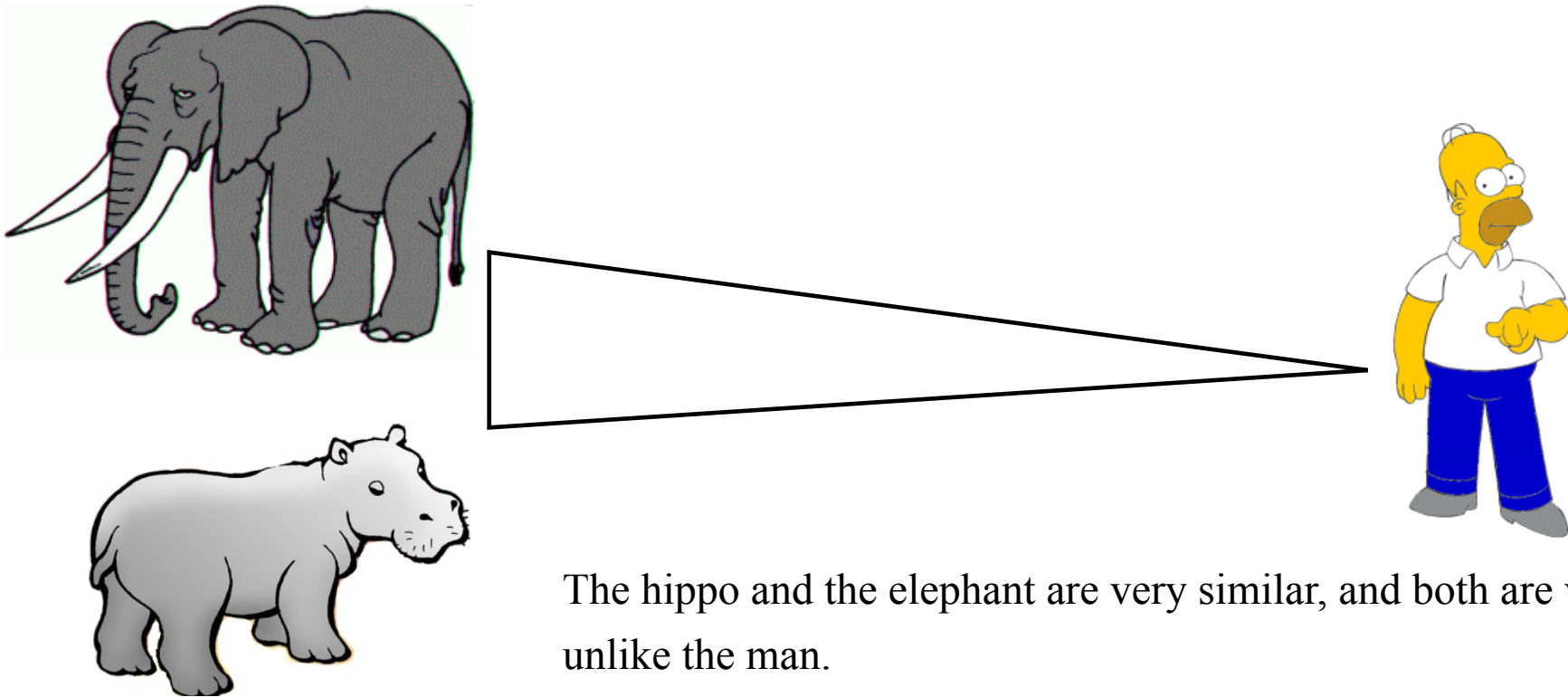


	a	b	c
a		6.70	7.07
b			2.30
c			

A Final Thought on the Triangular Inequality I

Sometimes the triangular inequality requirement maps nicely onto human intuitions.

Consider the similarity between a hippo, an elephant and a man.

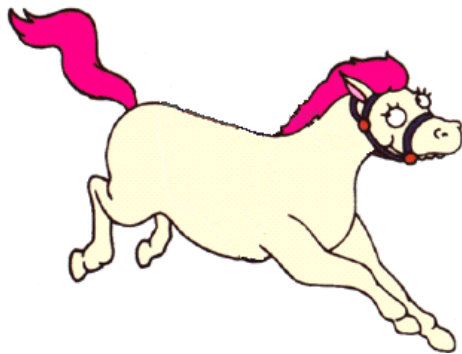


The hippo and the elephant are very similar, and both are very unlike the man.

A Final Thought on the Triangular Inequality II

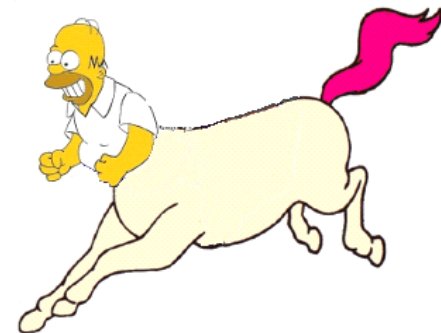
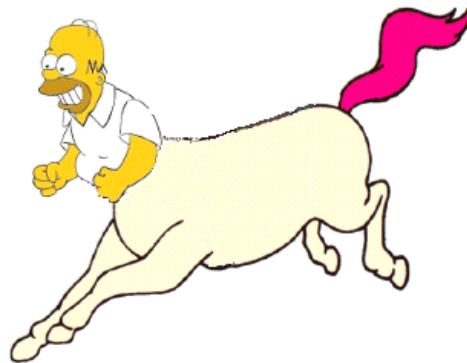
Sometimes the triangular inequality requirement *fails* to map onto human intuition.

Consider the similarity between the horse, a man and the centaur...



The **horse** and the **man** are very different, but both share many features with the **centaur**.

This relationship does not obey the triangular inequality.



This example due to Remco C. Veltkamp

A generic technique for measuring similarity

To measure the similarity between two objects, transform one of the objects into the other, and measure how much effort it took. The measure of effort becomes the distance measure.

The distance between Patty and Selma.

Change dress color, 1 point

Change earring shape, 1 point

Change hair part, 1 point

$D(\text{Patty}, \text{Selma}) = 3$

The distance between Marge and Selma.

Change dress color, 1 point

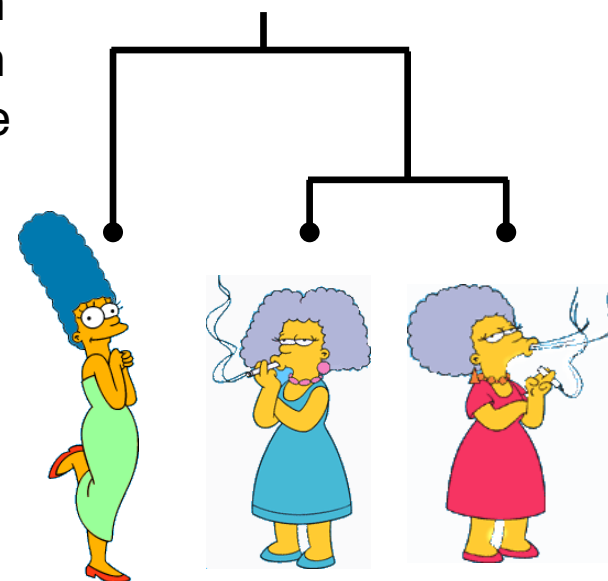
Add earrings, 1 point

Decrease height, 1 point

Take up smoking, 1 point

Lose weight, 1 point

$D(\text{Marge}, \text{Selma}) = 5$



This is called the “edit distance” or the “transformation distance”

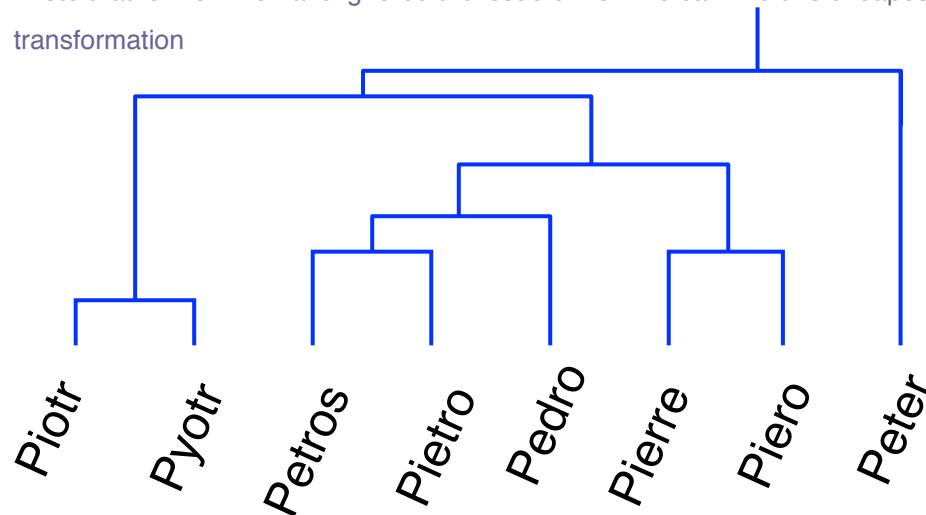
Edit Distance Example

It is possible to transform any string Q into string C , using only *Substitution*, *Insertion* and *Deletion*.

Assume that each of these operators has a cost associated with it.

The similarity between two strings can be defined as the cost of the cheapest transformation from Q to C .

Note that for now we have ignored the issue of how we can find this cheapest transformation

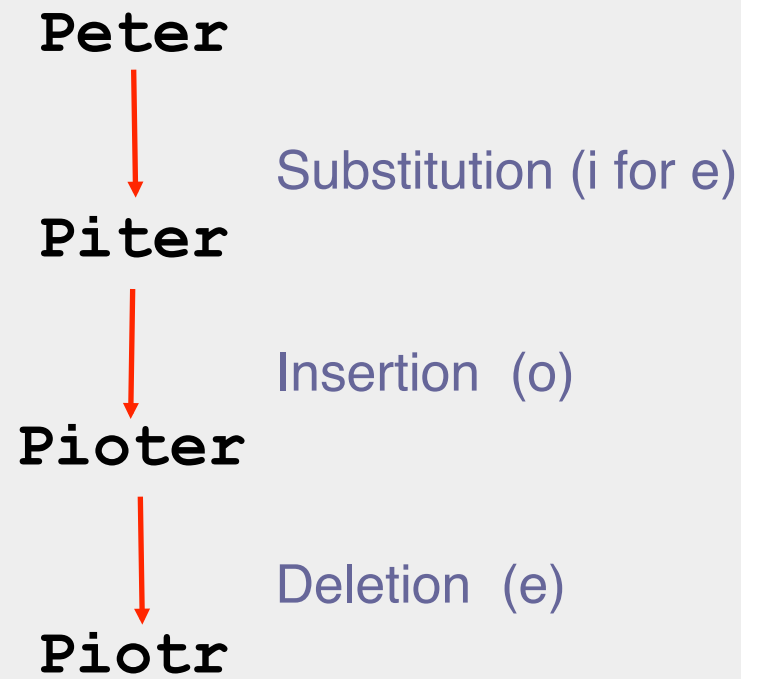


How similar are the names “Peter” and “Piotr”?

Assume the following cost function

<i>Substitution</i>	1 Unit
<i>Insertion</i>	1 Unit
<i>Deletion</i>	1 Unit

$D(\text{Peter}, \text{Piotr})$ is 3



A Demonstration of Hierarchical Clustering using String Edit Distance

Pedro (Portuguese)

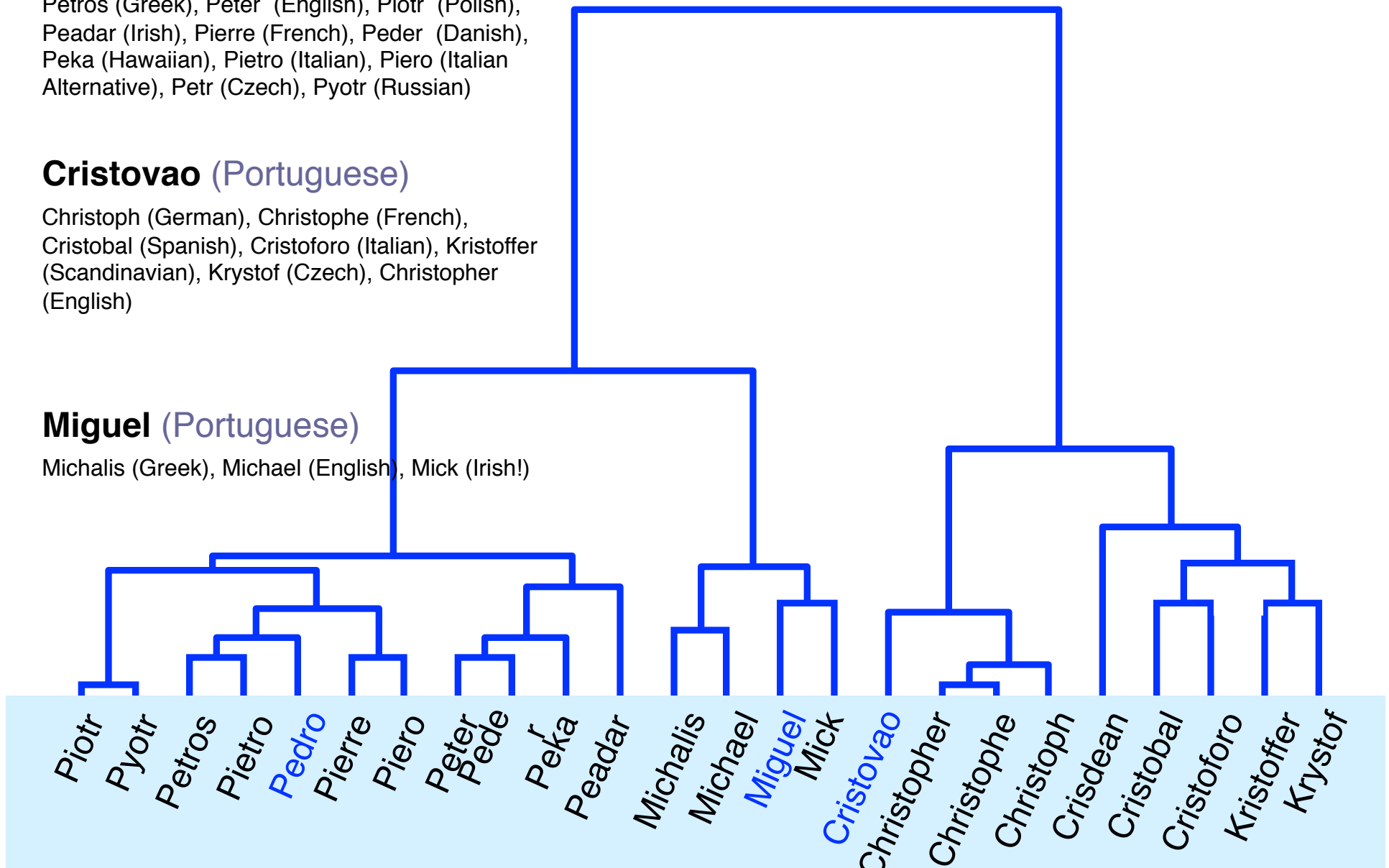
Petros (Greek), Peter (English), Piotr (Polish), Peadar (Irish), Pierre (French), Peder (Danish), Peka (Hawaiian), Pietro (Italian), Piero (Italian Alternative), Petr (Czech), Pyotr (Russian)

Cristovao (Portuguese)

Christoph (German), Christophe (French), Cristobal (Spanish), Cristoforo (Italian), Kristoffer (Scandinavian), Krystof (Czech), Christopher (English)

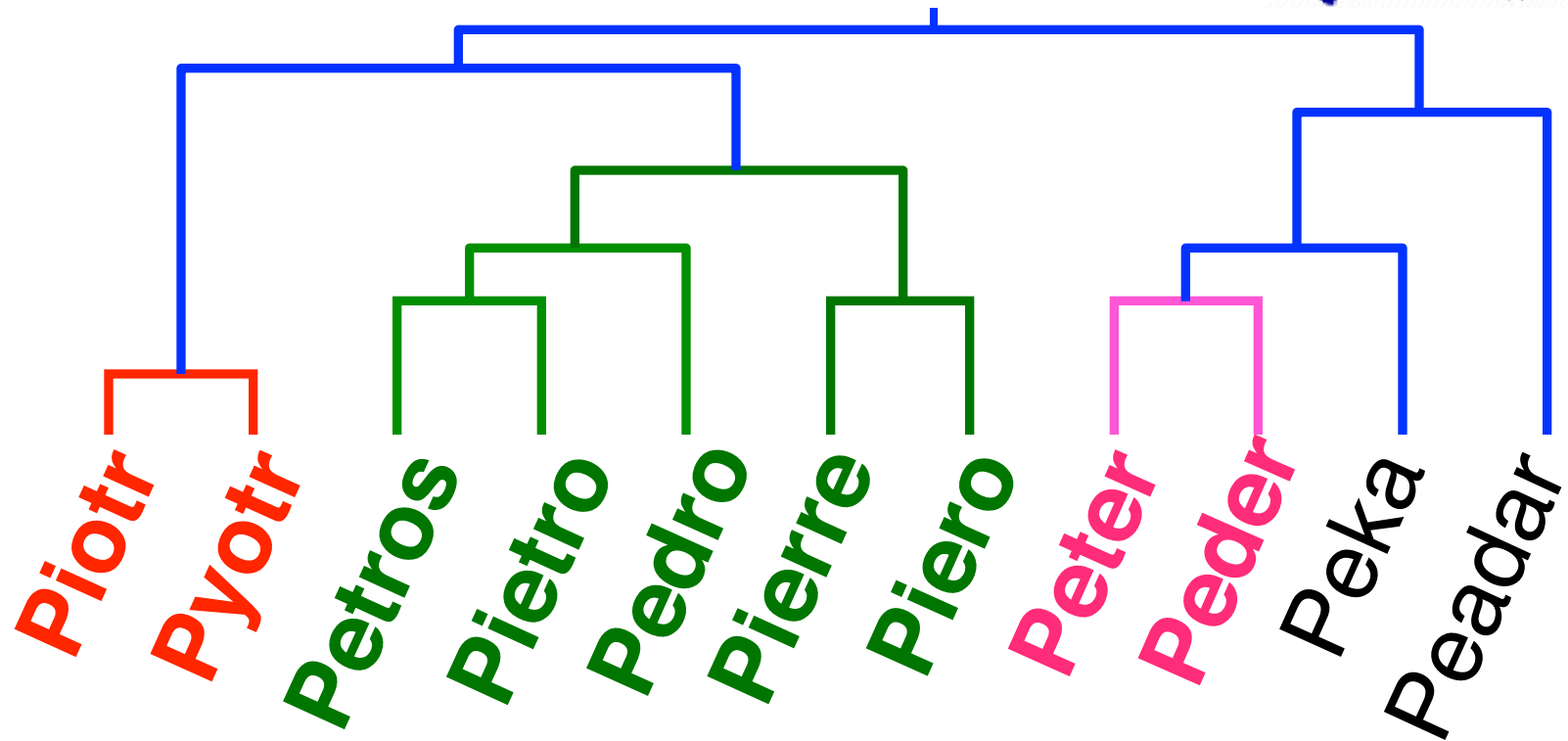
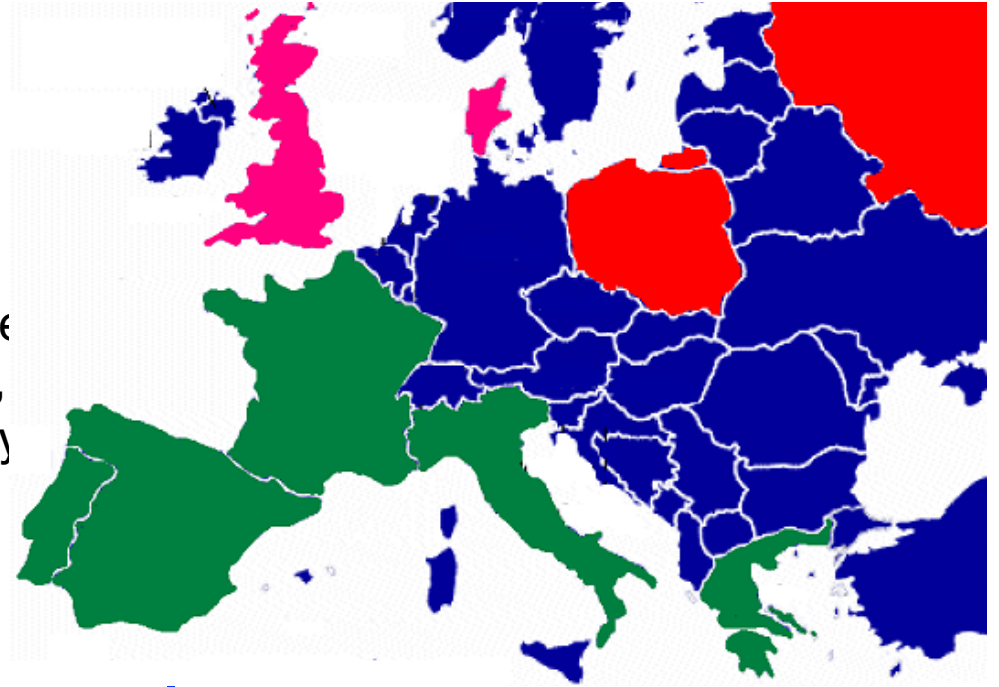
Miguel (Portuguese)

Michalis (Greek), Michael (English), Mick (Irish!)



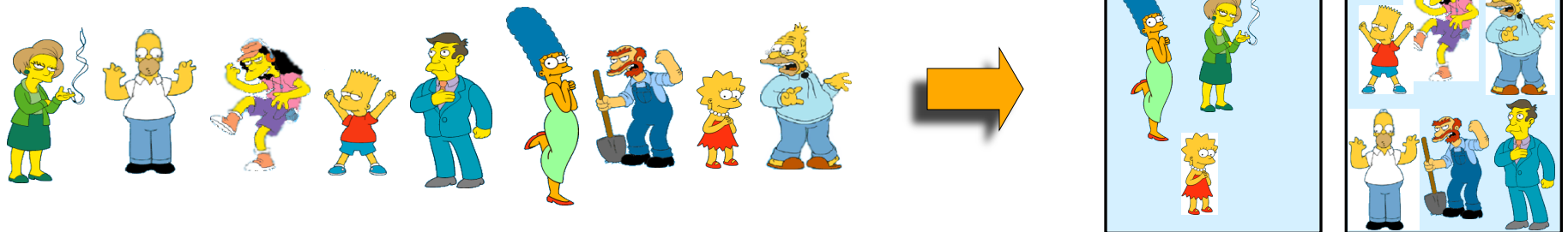
Pedro (Portuguese/ Spanish)

Petros (Greek), Peter (English), Piotr (Polish), Peadar (Irish), Pierre (French), Pēteris (Danish), Peka (Hawaiian), Pietro (Italian), Piero (Italian Alternative), Petr (Czech), Pyotr (Russian)



Partitional Clustering

- Nonhierarchical, each instance is placed in exactly one of K nonoverlapping clusters.
- Since only one set of clusters is output, the user normally has to input the desired number of clusters K .

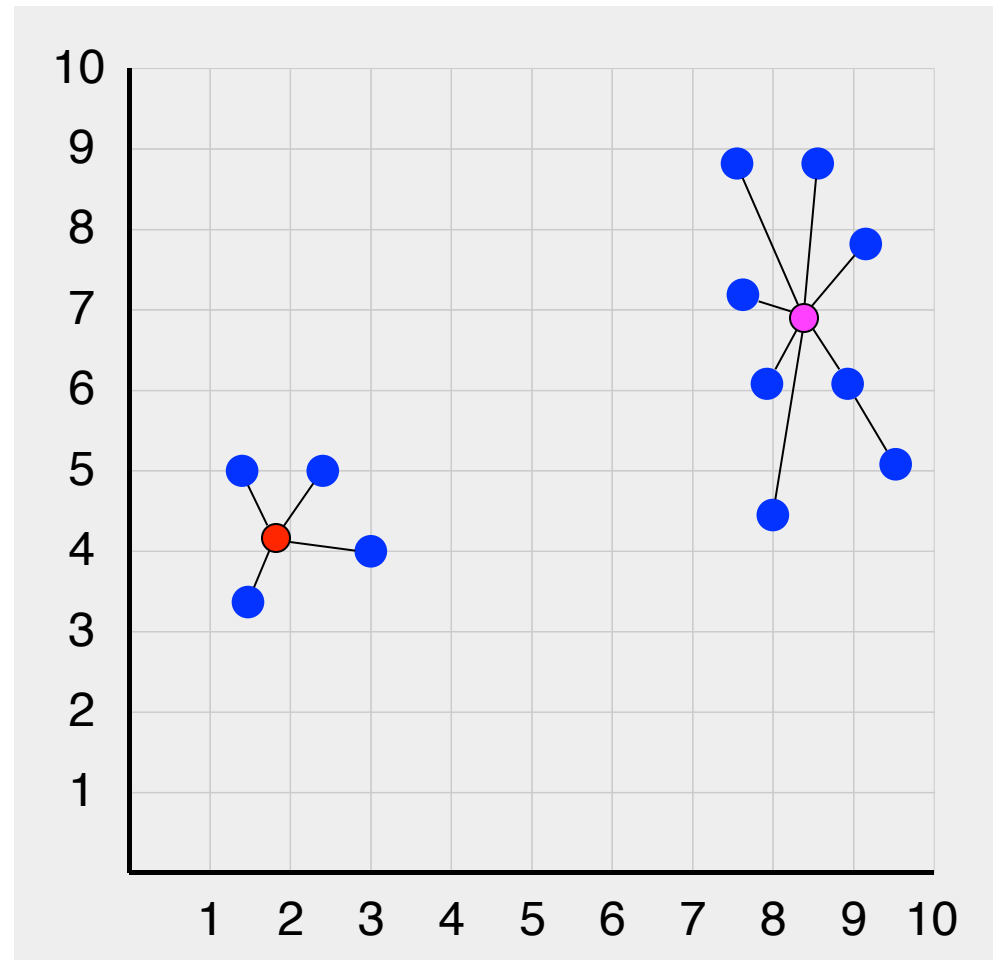


Squared Error

$$se_{K_i} = \sum_{j=1}^m \|t_{ij} - C_k\|^2$$

$$se_K = \sum_{j=1}^k se_{K_j}$$

Objective Function

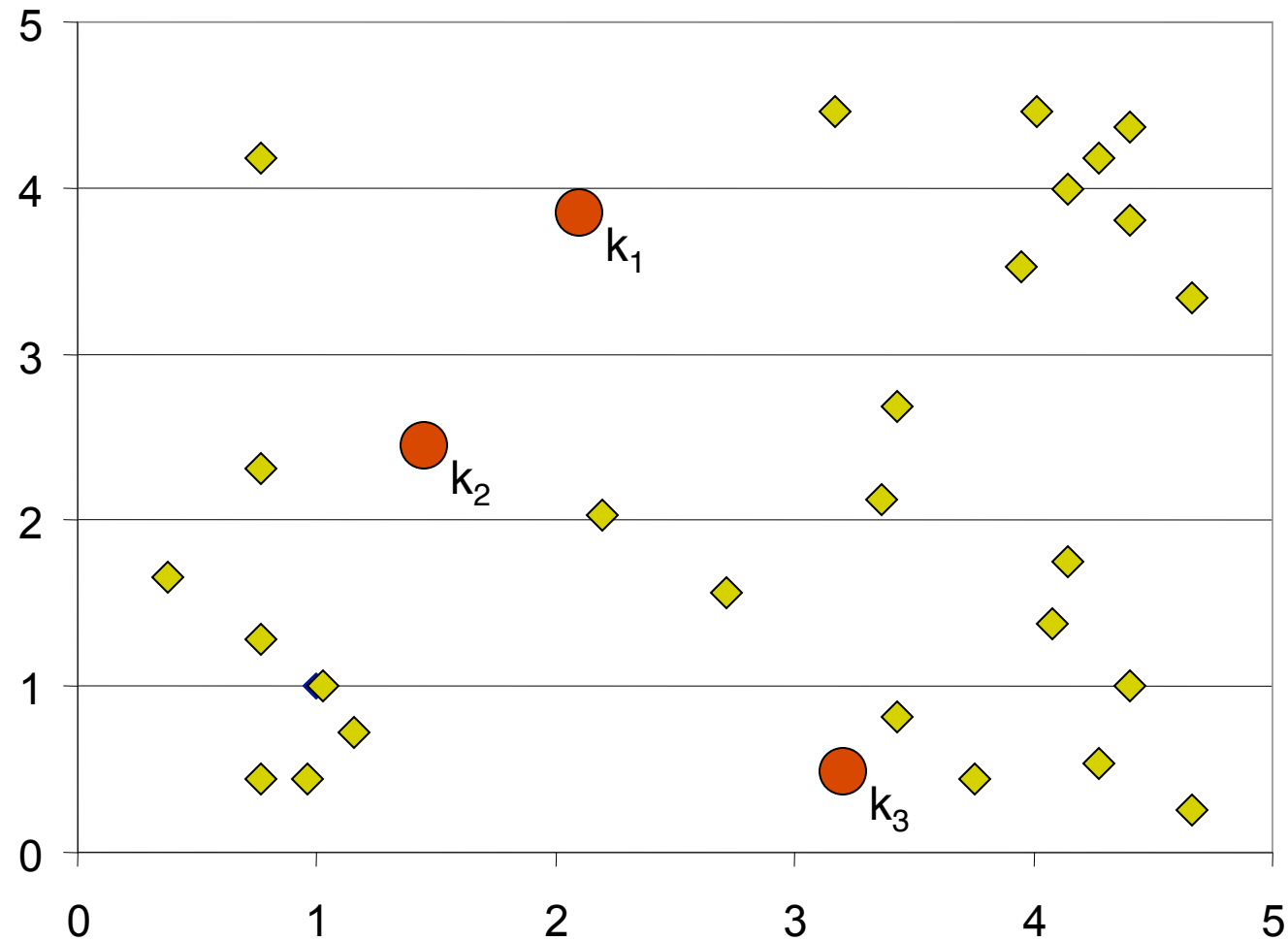


Algorithm *k-means*

- 1. Decide on a value for k .
- 2. Initialize the k cluster centers (randomly, if necessary).
- 3. Decide the class memberships of the N objects by assigning them to the nearest cluster center.
- 4. Re-estimate the k cluster centers, by assuming the memberships found above are correct.
- 5. If none of the N objects changed membership in the last iteration, exit. Otherwise goto 3.

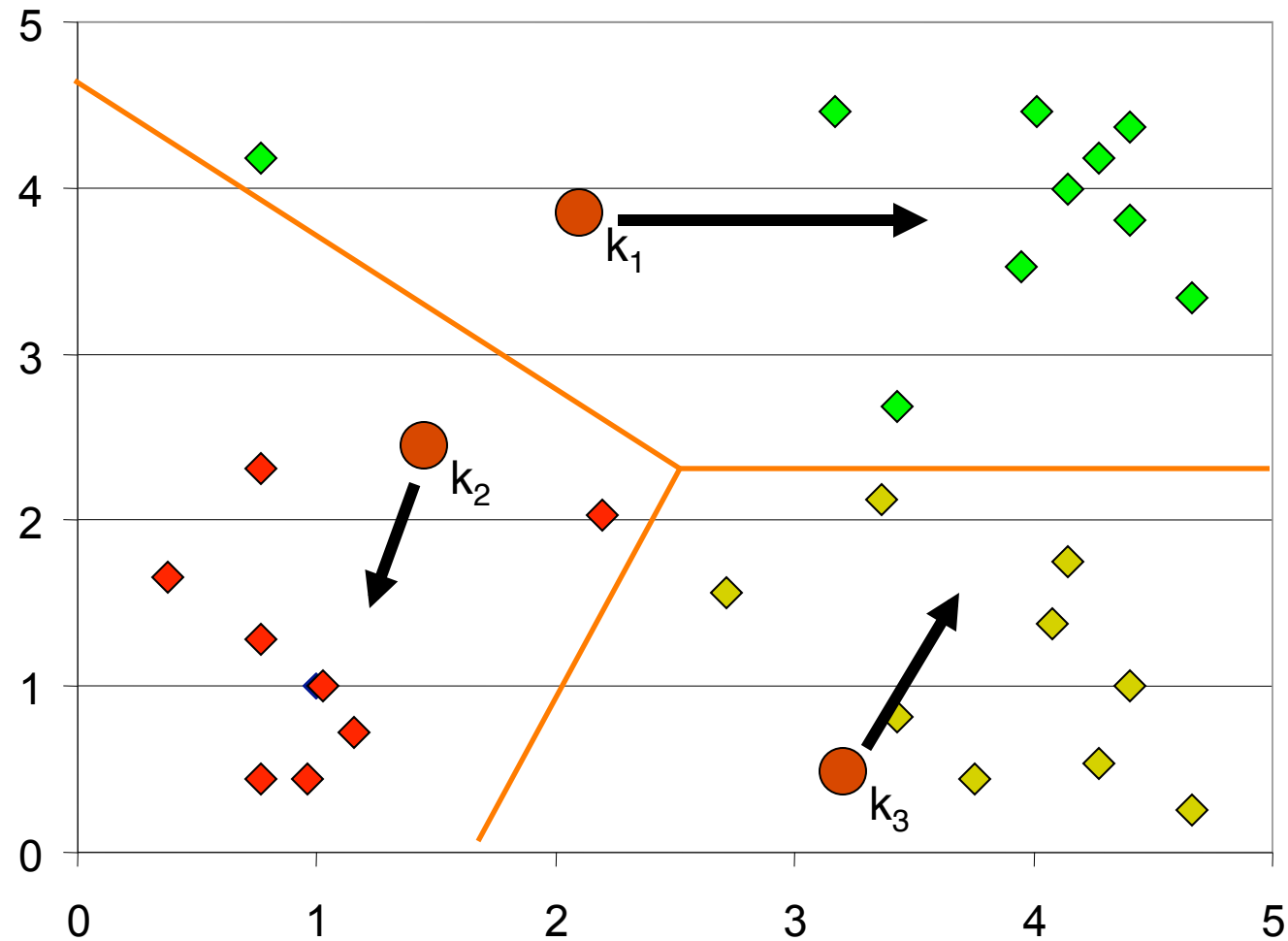
K-means Clustering: Step 1

Algorithm: k-means, Distance Metric: Euclidean Distance



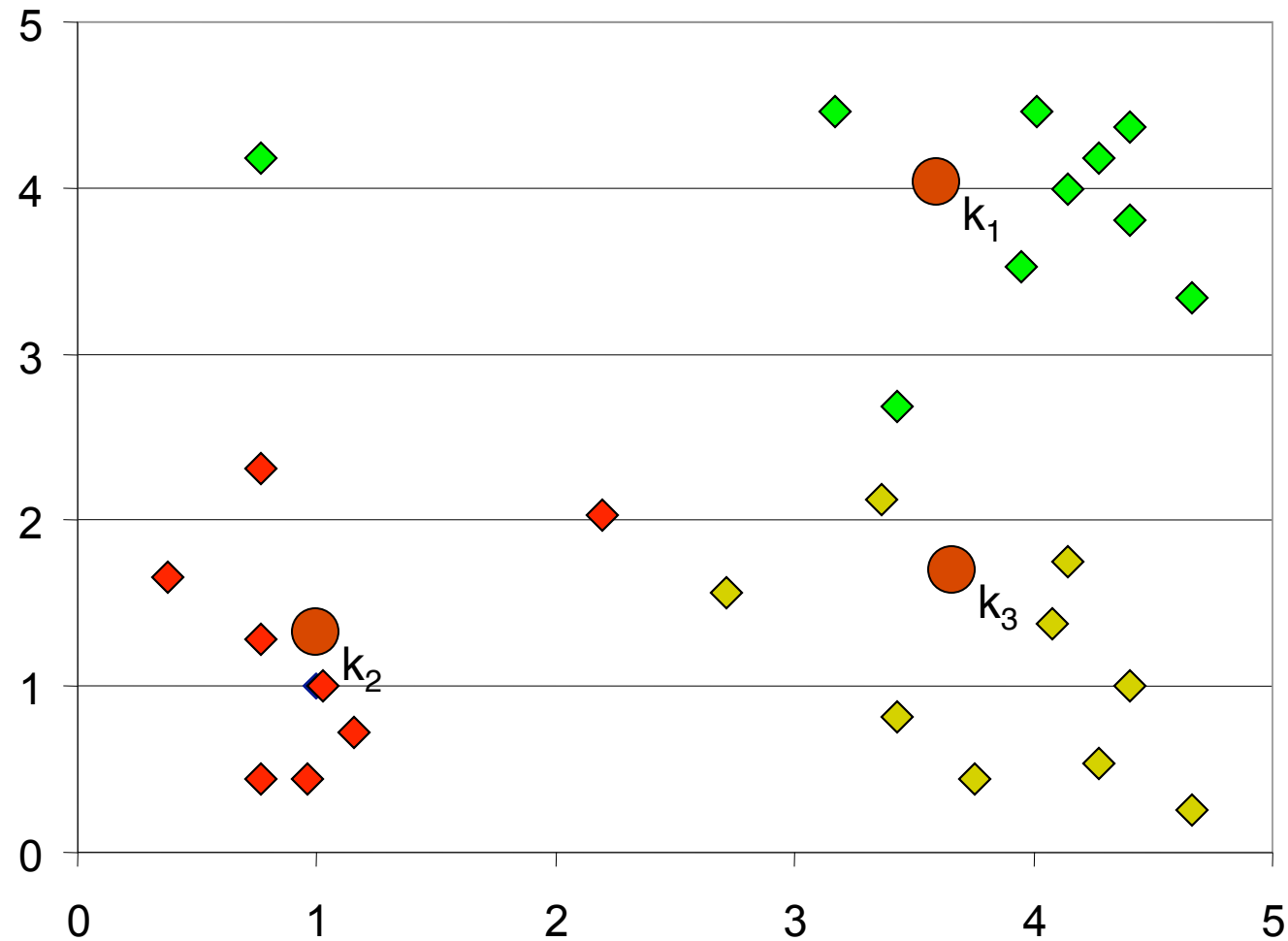
K-means Clustering: Step 2

Algorithm: k-means, Distance Metric: Euclidean Distance



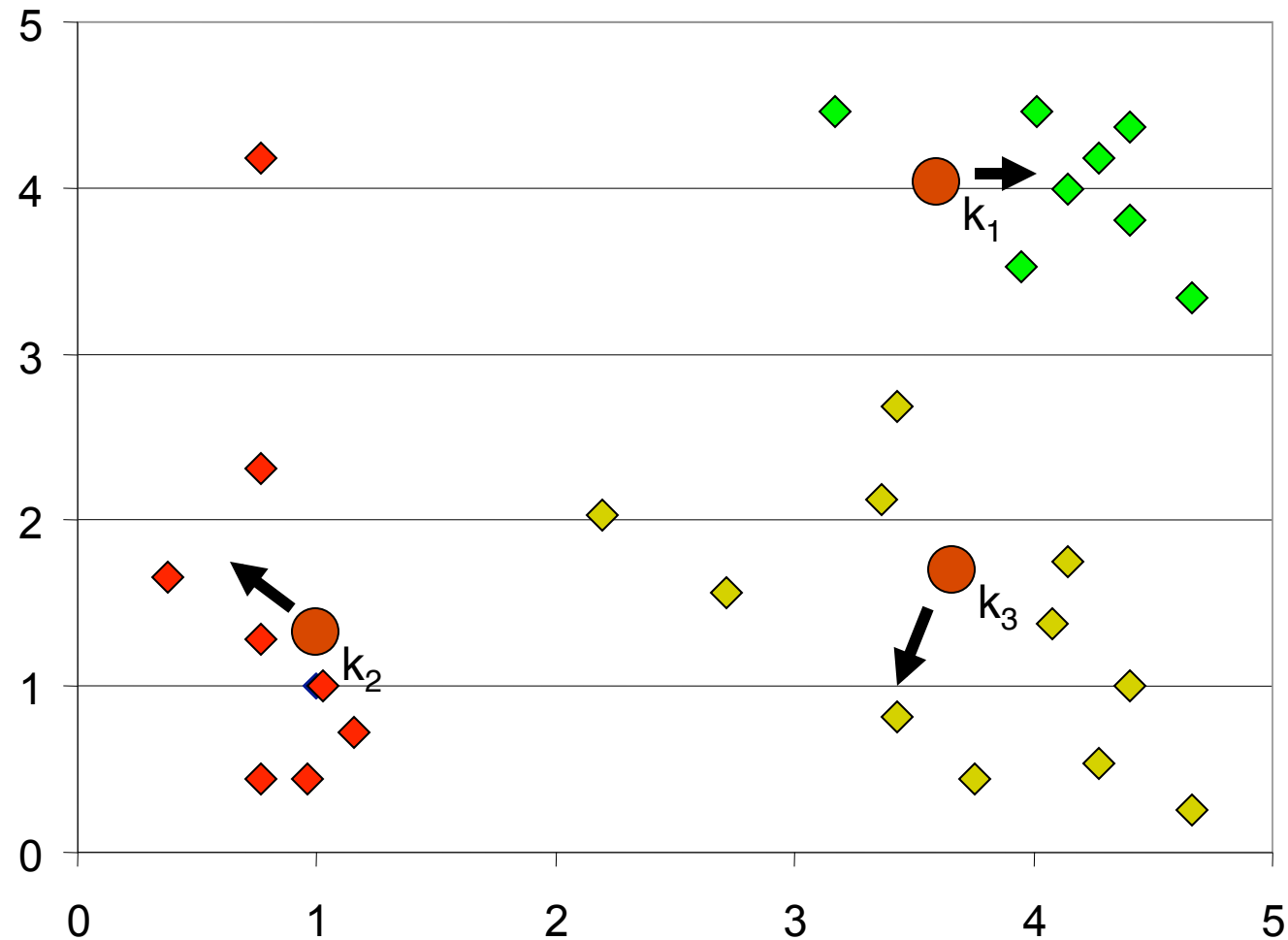
K-means Clustering: Step 3

Algorithm: k-means, Distance Metric: Euclidean Distance



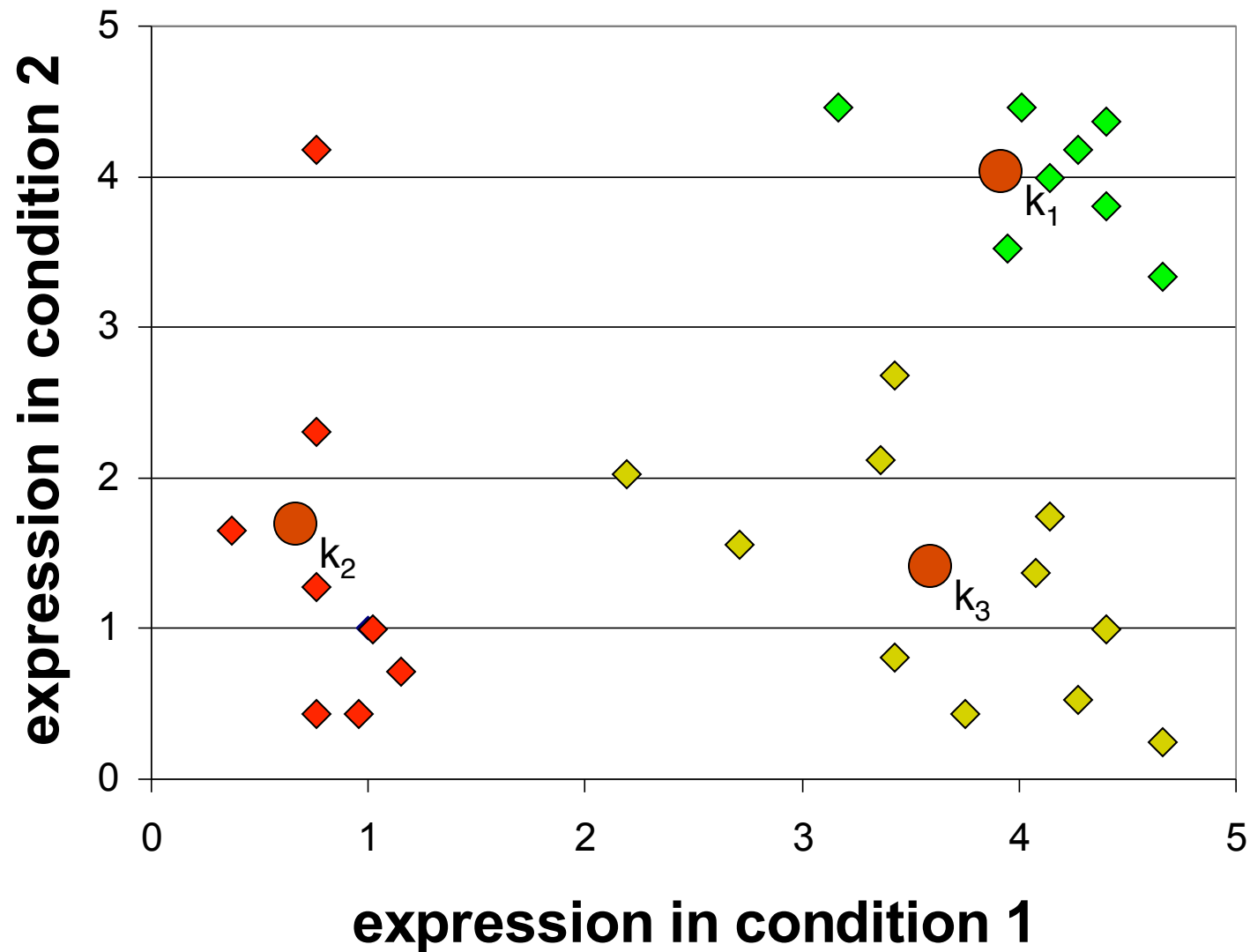
K-means Clustering: Step 4

Algorithm: k-means, Distance Metric: Euclidean Distance

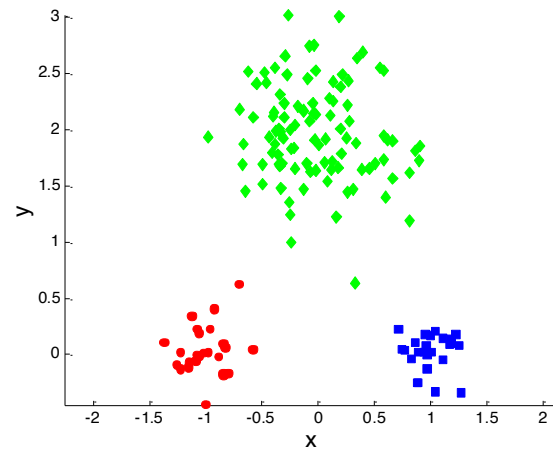


K-means Clustering: Step 5

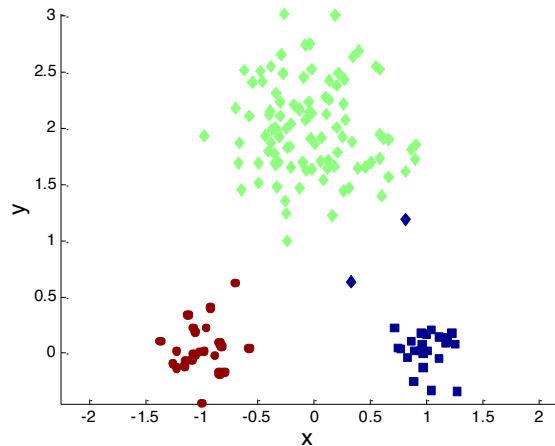
Algorithm: k-means, Distance Metric: Euclidean Distance



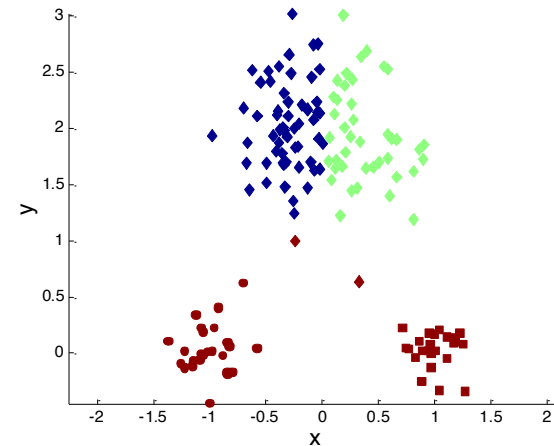
Two different K-means Clusterings



Original Points

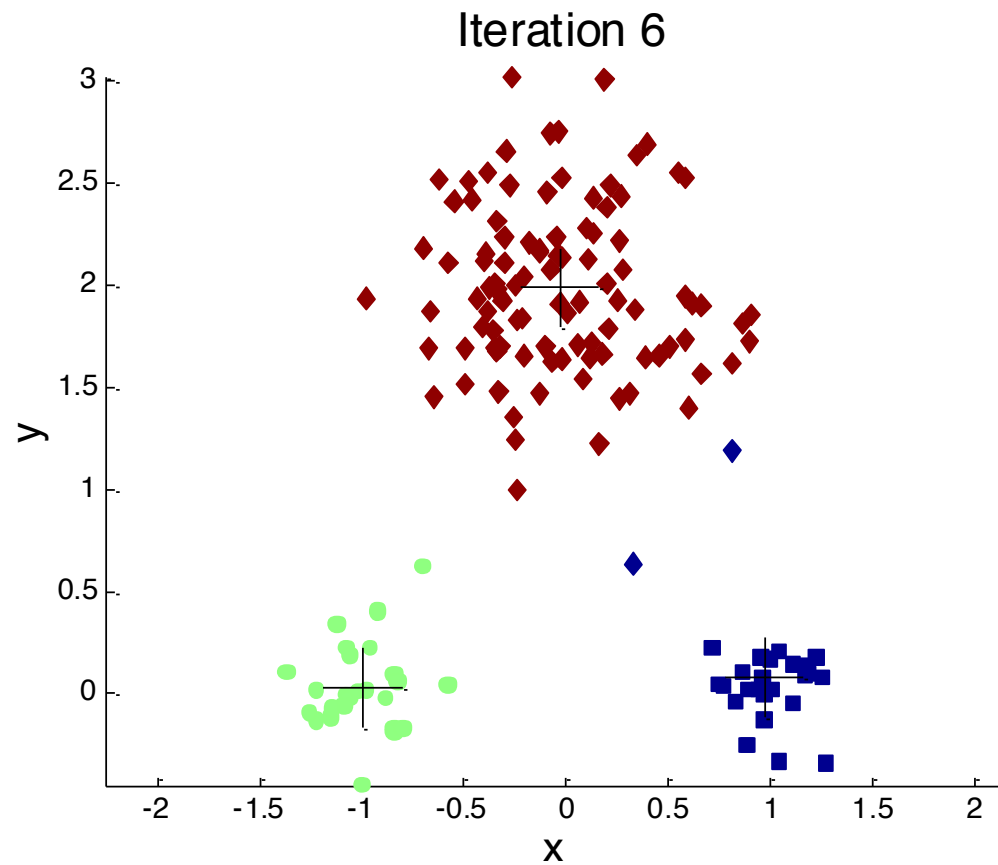


Optimal Clustering

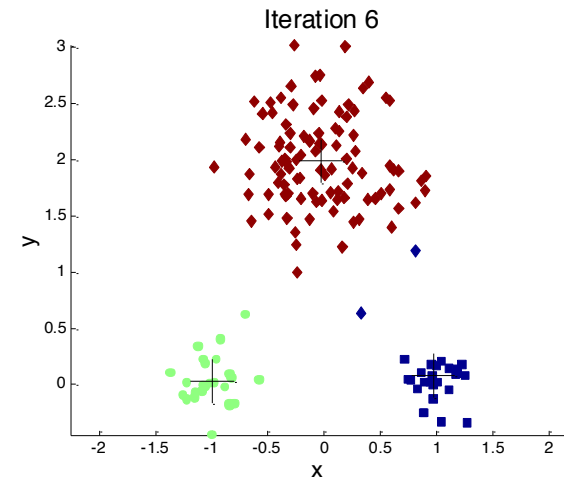
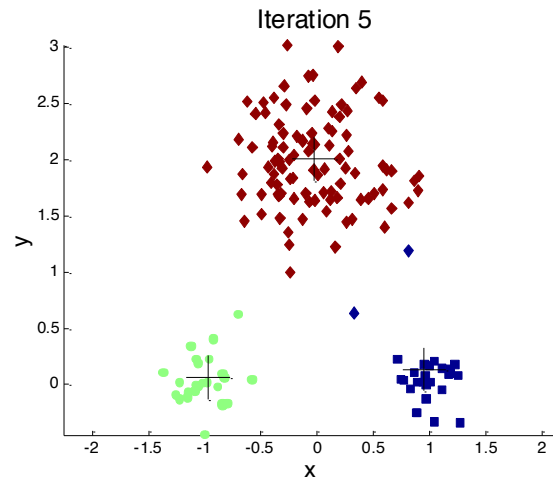
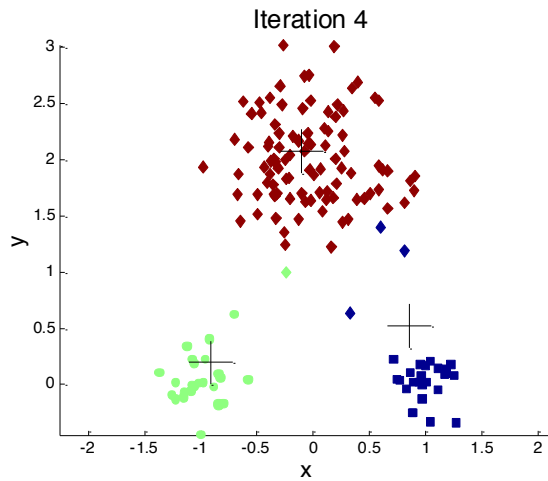
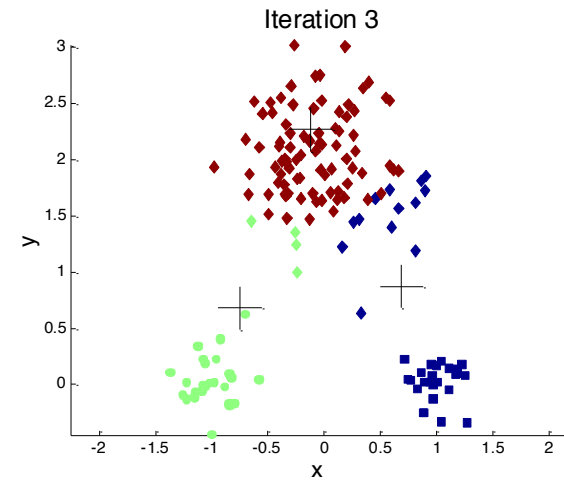
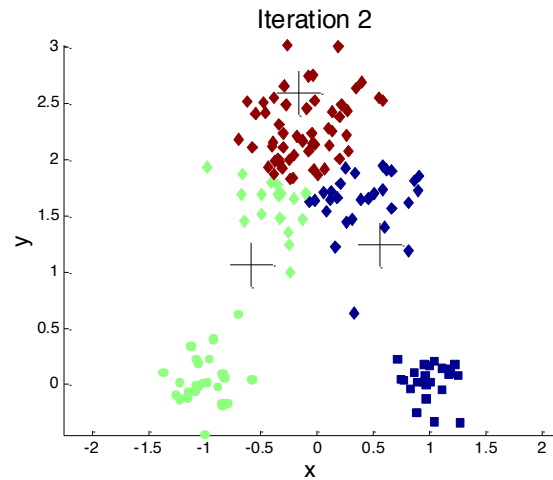
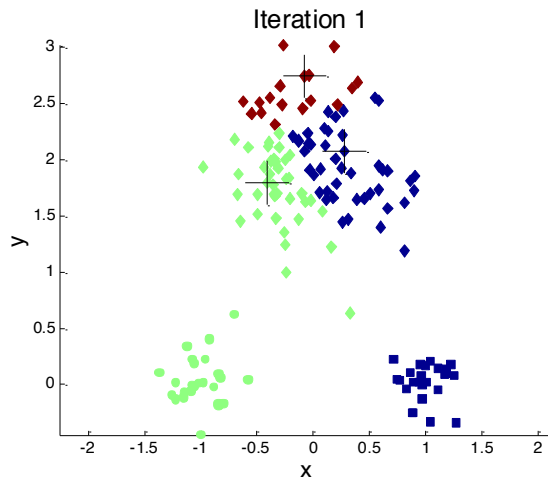


Sub-optimal Clustering

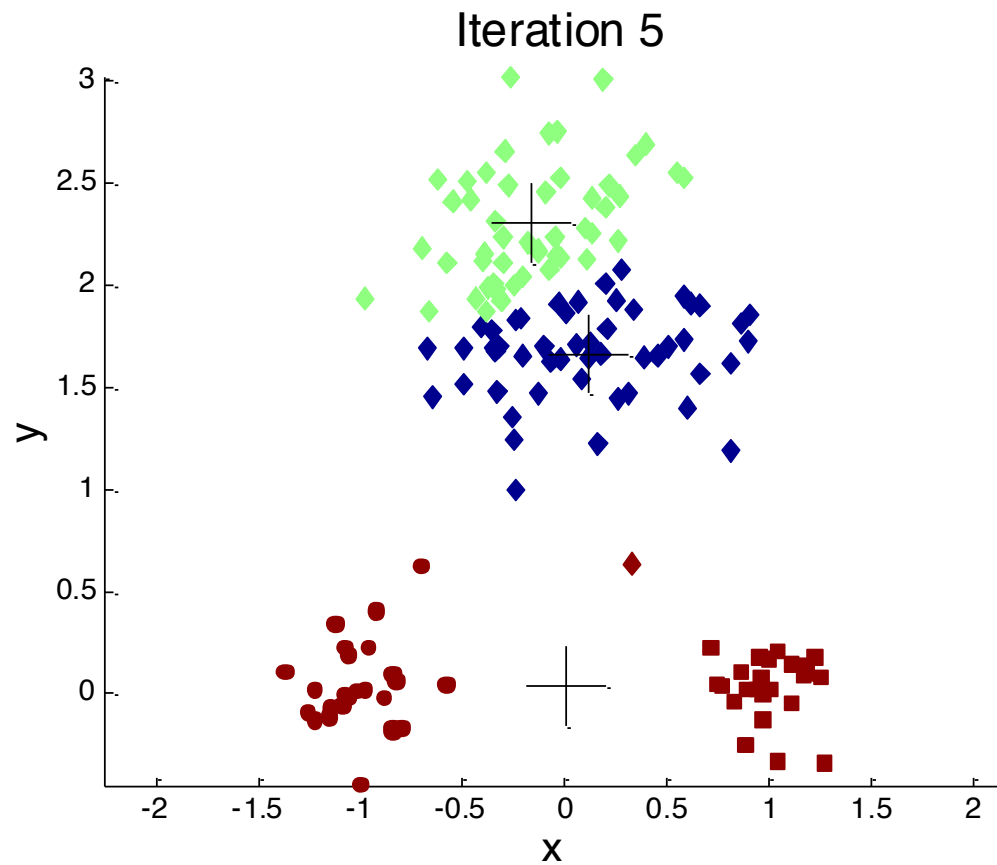
Importance of Choosing Initial Centroids



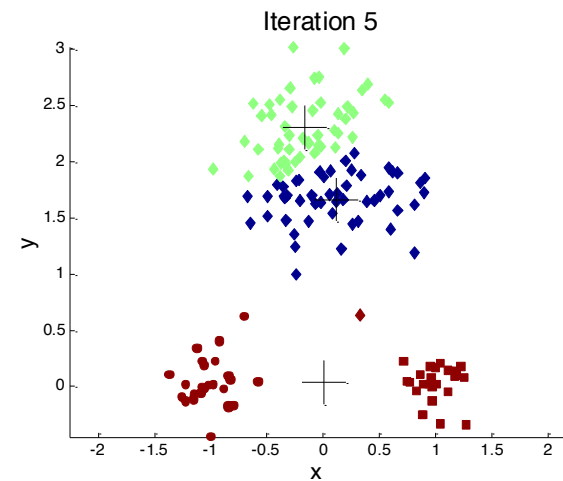
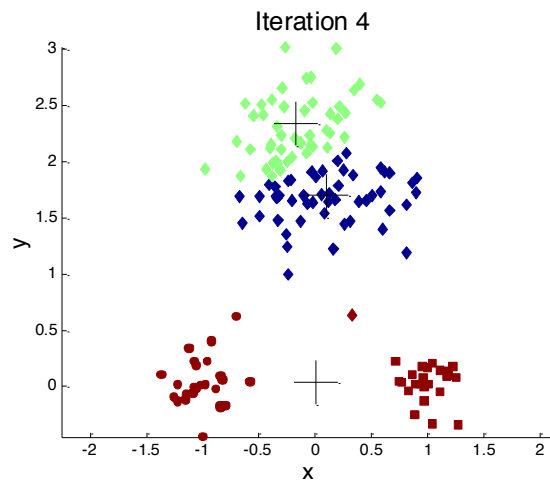
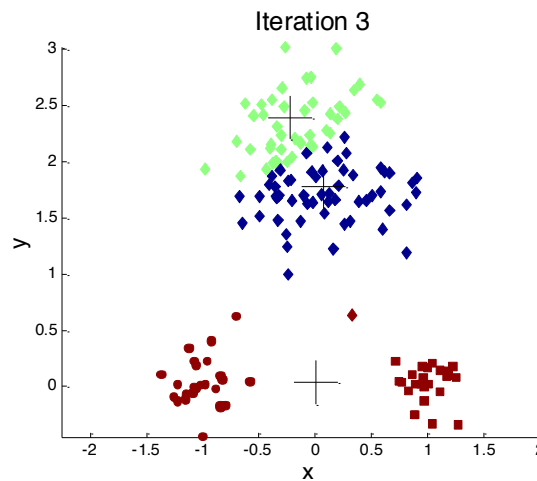
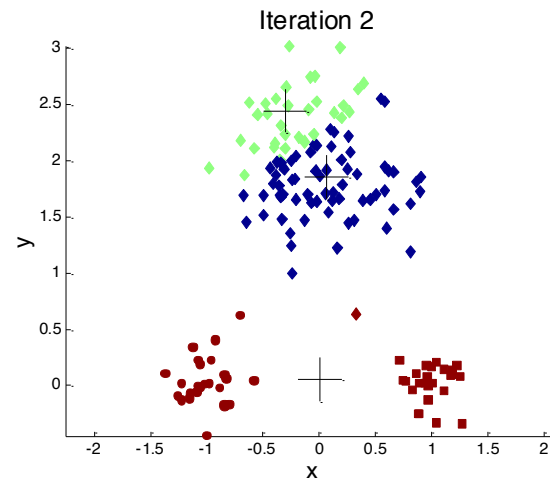
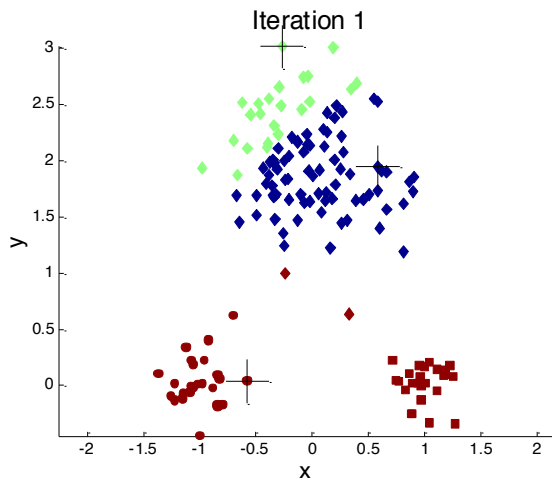
Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids ...



Importance of Choosing Initial Centroids ...



Comments on the *K-Means* Method

■ Strength

- ★ *Relatively efficient: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t \ll n$.*
- ★ *Often terminates at a local optimum. The global optimum may be found using techniques such as: deterministic annealing and genetic algorithms*

■ Weakness

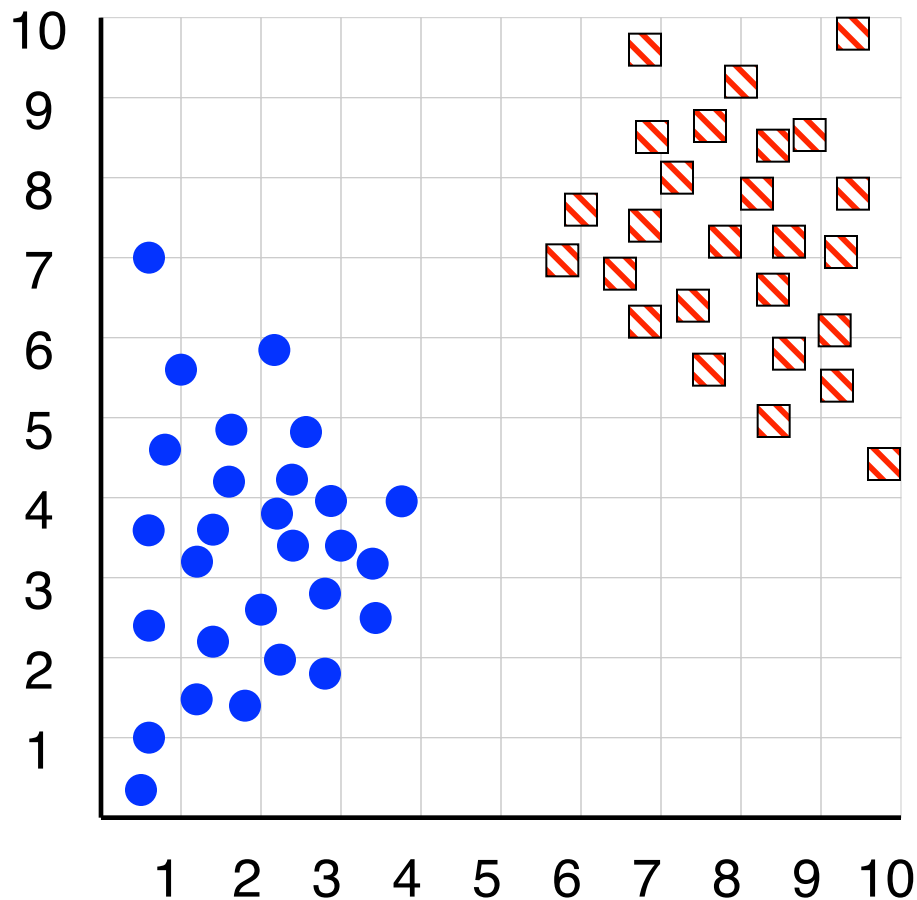
- ★ *Applicable only when *mean* is defined, then what about categorical data?*
- ★ *Need to specify k , the number of clusters, in advance*
- ★ *Unable to handle noisy data and outliers*
- ★ *Not suitable to discover clusters with non-convex shapes*

The *K-Medoids* Clustering Method

- Find *representative* objects, called medoids, in clusters
- *PAM* (Partitioning Around Medoids, 1987)
 - ★ starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
 - ★ *PAM* works effectively for small data sets, but does not scale well for large data sets

How can we tell the *right* number of clusters?

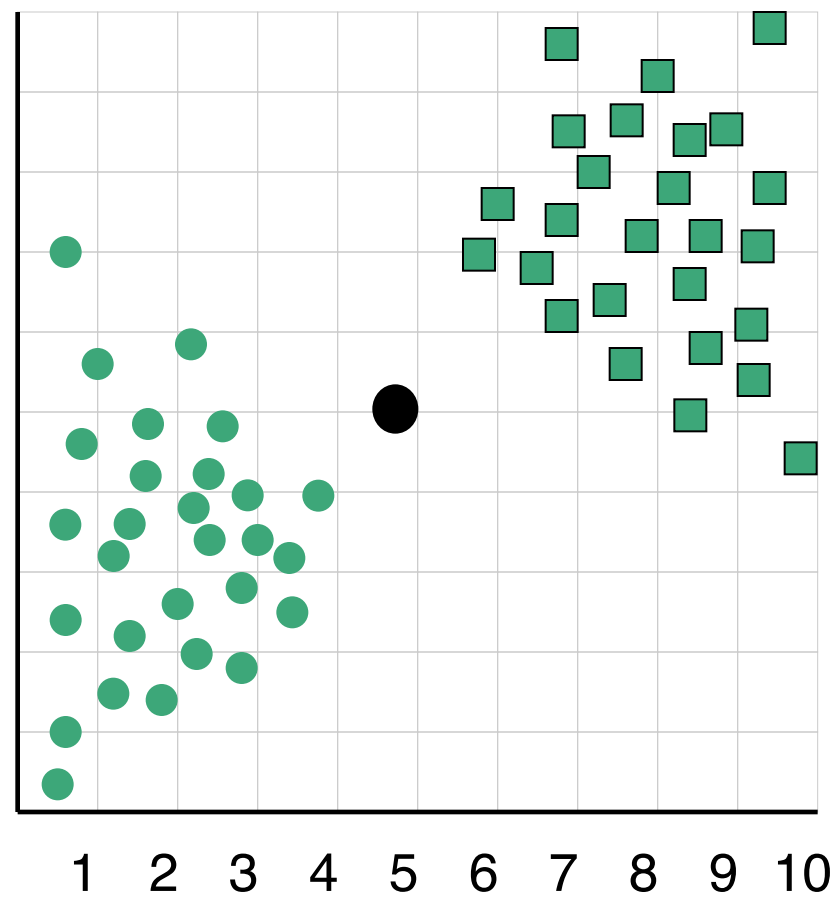
In general, this is a unsolved problem. However there are many approximate methods. In the next few slides we will see an example.



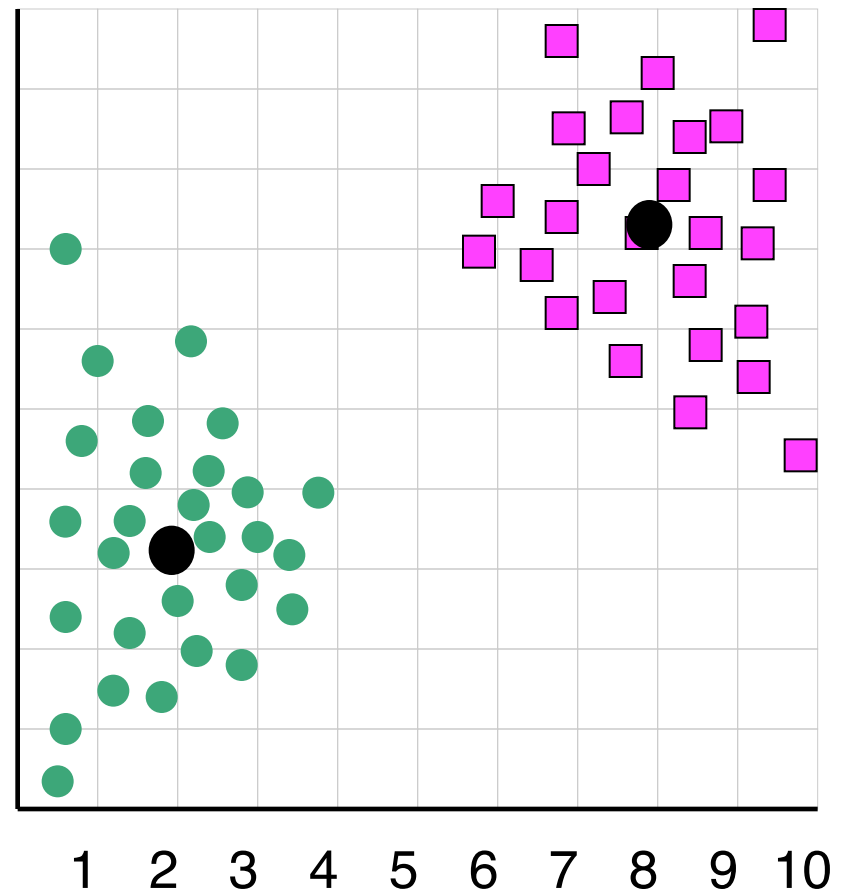
For our example, we will use the familiar **katydid**/**grasshopper** dataset.

However, in this case we are imagining that we do NOT know the class labels. We are only clustering on the X and Y axis values.

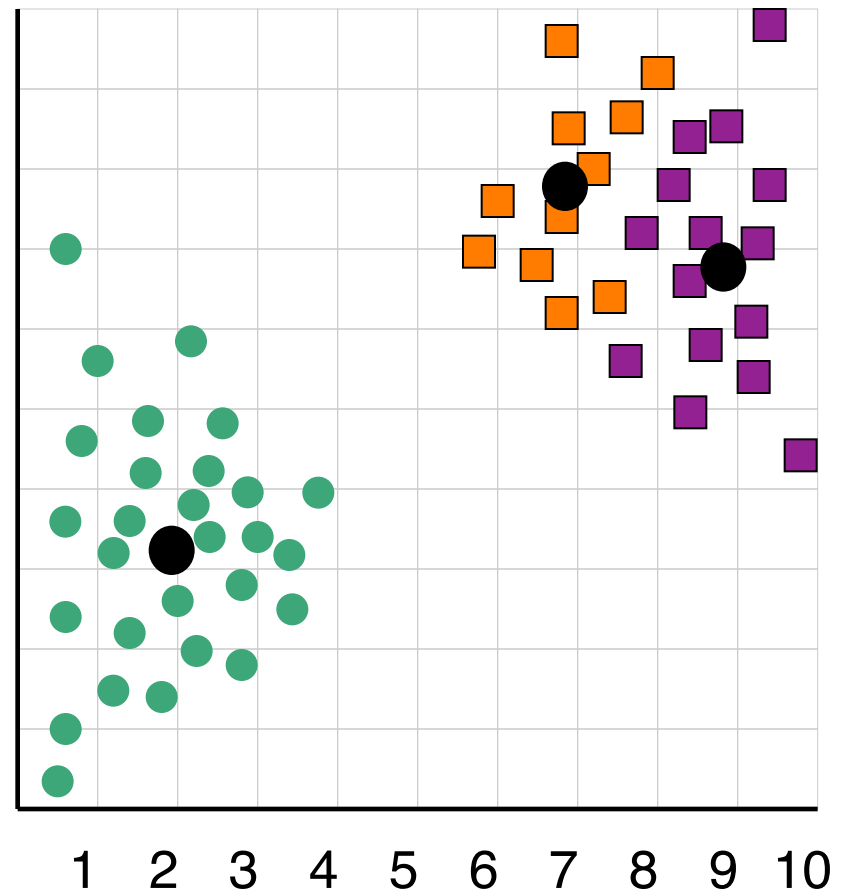
When $k = 1$, the objective function is 873.0



When $k = 2$, the objective function is 173.1

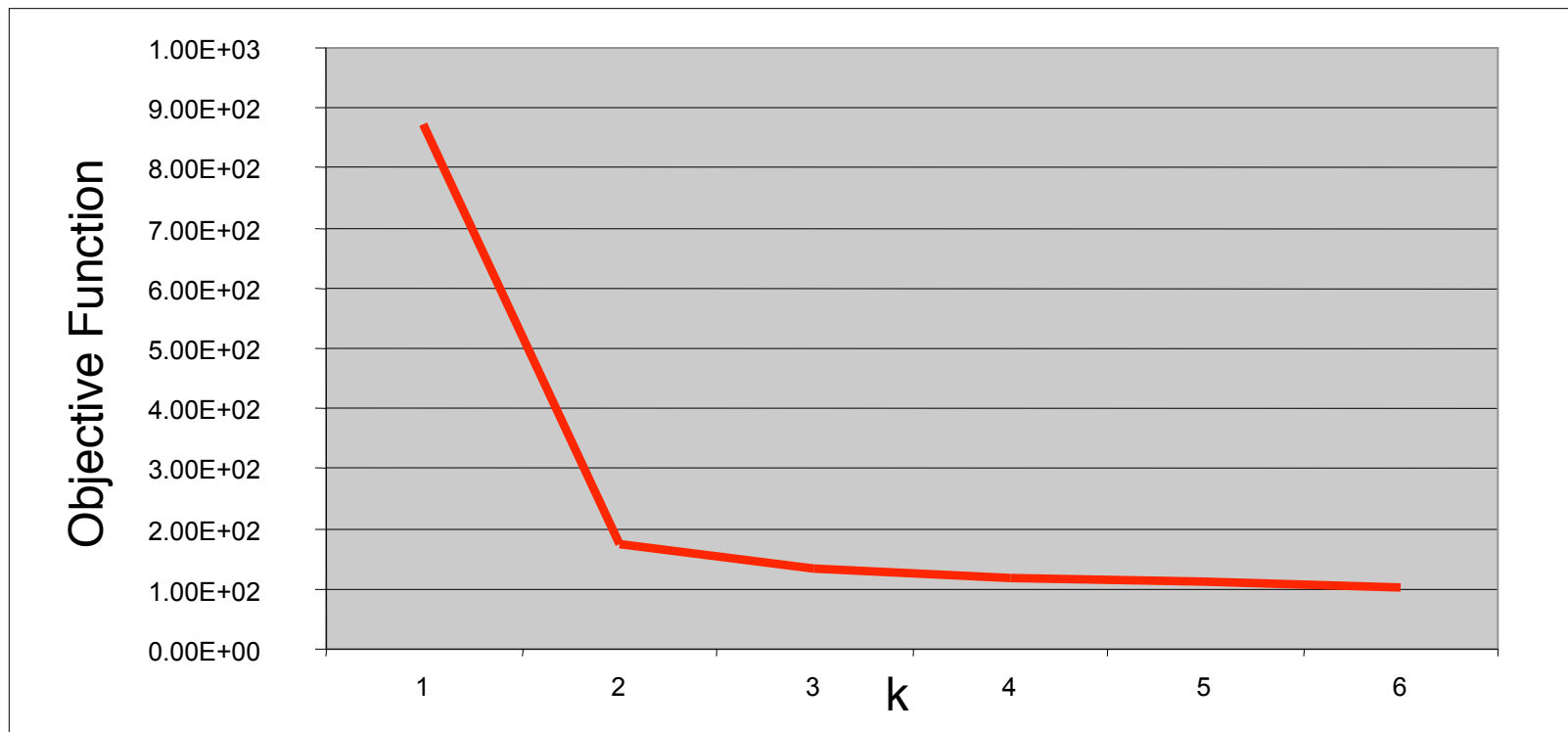


When $k = 3$, the objective function is 133.6



We can plot the objective function values for k equals 1 to 6...

The abrupt change at $k = 2$, is highly suggestive of two clusters in the data. This technique for determining the number of clusters is known as “knee finding” or “elbow finding”.



Note that the results are not always as clear cut as in this toy example

Association Rules

(market basket analysis)

- Retail shops are often interested in associations between different items that people buy.
 - Someone who buys bread is quite likely also to buy milk
 - A person who bought the book *Database System Concepts* is quite likely also to buy the book *Operating System Concepts*.
- Associations information can be used in several ways.
 - E.g. when a customer buys a particular book, an online shop may suggest associated books.
- **Association rules:**
 - bread* \Rightarrow *milk* *DB-Concepts, OS-Concepts* \Rightarrow *Networks*
 - Left hand side: **antecedent**, right hand side: **consequent**
 - An association rule must have an associated **population**; the population consists of a set of **instances**
 - E.g. each transaction (sale) at a shop is an instance, and the set of all transactions is the population

Association Rule Discovery: Application 1

■ Marketing and Sales Promotion:

★ Let the rule discovered be

$\{\text{Bagels, ...}\} \rightarrow \{\text{Potato Chips}\}$

★ **Potato Chips as consequent** \Rightarrow Can be used to determine what should be done to boost its sales.

★ **Bagels in the antecedent** \Rightarrow Can be used to see which products would be affected if the store discontinues selling bagels.

★ **Bagels in antecedent and Potato chips in consequent** \Rightarrow Can be used to see what products should be sold with Bagels to promote sale of Potato chips!

Association Rule Discovery: Application 2

■ Supermarket shelf management.

- ★ Goal: To identify items that are bought together by sufficiently many customers.
- ★ Approach: Process the point-of-sale data collected with barcode scanners to find dependencies among items.
- ★ Wal-mart, Target, and departmental store managers are big into this.
- ★ All your ticket gets processed & analyzed in a warehouse.

Association Rule Mining

- Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Market-Basket transactions

Example of Association Rules

$\{\text{Diaper}\} \rightarrow \{\text{Beer}\},$
 $\{\text{Milk, Bread}\} \rightarrow \{\text{Eggs, Coke}\},$
 $\{\text{Beer, Bread}\} \rightarrow \{\text{Milk}\},$

Implication means co-occurrence,
not causality!

Definition: Frequent Itemset

■ Itemset

★ A collection of one or more items

✓ Example: {Milk, Bread, Diaper}

★ k-itemset

✓ An itemset that contains k items

■ Support count (σ)

★ Frequency of occurrence of an itemset

★ E.g. $\sigma(\{\text{Milk, Bread, Diaper}\}) = 2$

■ Support

★ Fraction of transactions that contain an itemset

★ E.g. $s(\{\text{Milk, Bread, Diaper}\}) = 2/5$

■ Frequent Itemset

★ An itemset whose support is greater than or equal to a *minsup* threshold

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Definition: Association Rule

- Association Rule

- An implication expression of the form $X \rightarrow Y$, where X and Y are itemsets
- Example:
 $\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}$

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

- Rule Evaluation Metrics

- Support (s)
 - ◆ Fraction of transactions that contain both X and Y
- Confidence (c)
 - ◆ Measures how often items in Y appear in transactions that contain X

Example:

$\{\text{Milk, Diaper}\} \Rightarrow \text{Beer}$

$$s = \frac{\sigma(\text{Milk, Diaper, Beer})}{|T|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk, Diaper, Beer})}{\sigma(\text{Milk, Diaper})} = \frac{2}{3} = 0.67$$

Association Rule Mining Task

- Given a set of transactions T , the goal of association rule mining is to find all rules having
 - ★ support $\geq \textit{minsup}$ threshold
 - ★ confidence $\geq \textit{minconf}$ threshold
- Brute-force approach:
 - ★ List all possible association rules
 - ★ Compute the support and confidence for each rule
 - ★ Prune rules that fail the *minsup* and *minconf* thresholds

⇒ **Computationally prohibitive!**

Mining Association Rules

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Rules:

$\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}$ ($s=0.4$, $c=0.67$)

$\{\text{Milk, Beer}\} \rightarrow \{\text{Diaper}\}$ ($s=0.4$, $c=1.0$)

$\{\text{Diaper, Beer}\} \rightarrow \{\text{Milk}\}$ ($s=0.4$, $c=0.67$)

$\{\text{Beer}\} \rightarrow \{\text{Milk, Diaper}\}$ ($s=0.4$, $c=0.67$)

$\{\text{Diaper}\} \rightarrow \{\text{Milk, Beer}\}$ ($s=0.4$, $c=0.5$)

$\{\text{Milk}\} \rightarrow \{\text{Diaper, Beer}\}$ ($s=0.4$, $c=0.5$)

Observations:

- All the above rules are binary partitions of the same itemset:
 $\{\text{Milk, Diaper, Beer}\}$
- Rules originating from the same itemset have identical support but can have different confidence
- Thus, we may decouple the support and confidence requirements

Mining Association Rules

■ Two-step approach:

1. Frequent Itemset Generation

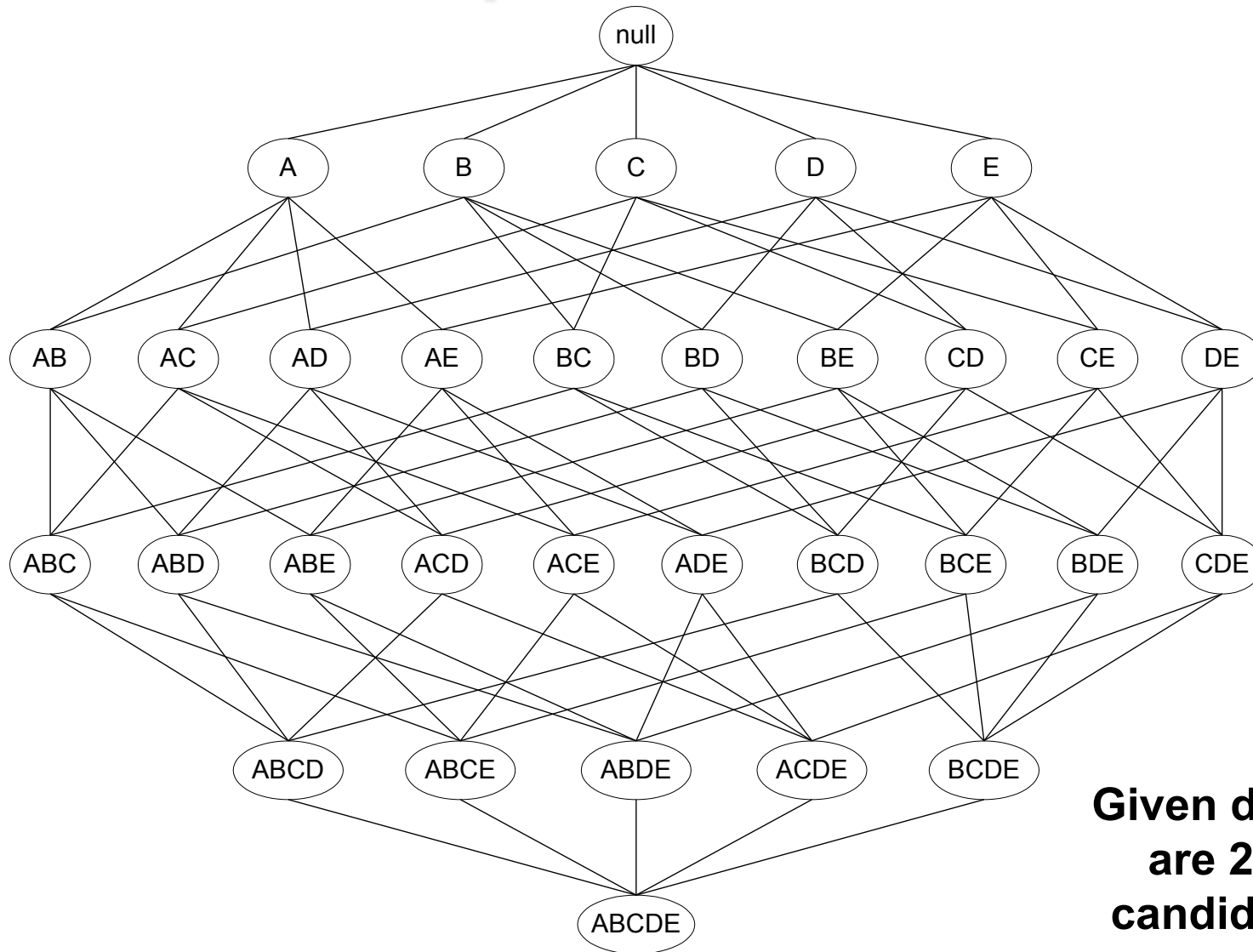
- Generate all itemsets whose support \geq minsup

2. Rule Generation

- Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

■ Frequent itemset generation is still computationally expensive

Frequent Itemset Generation

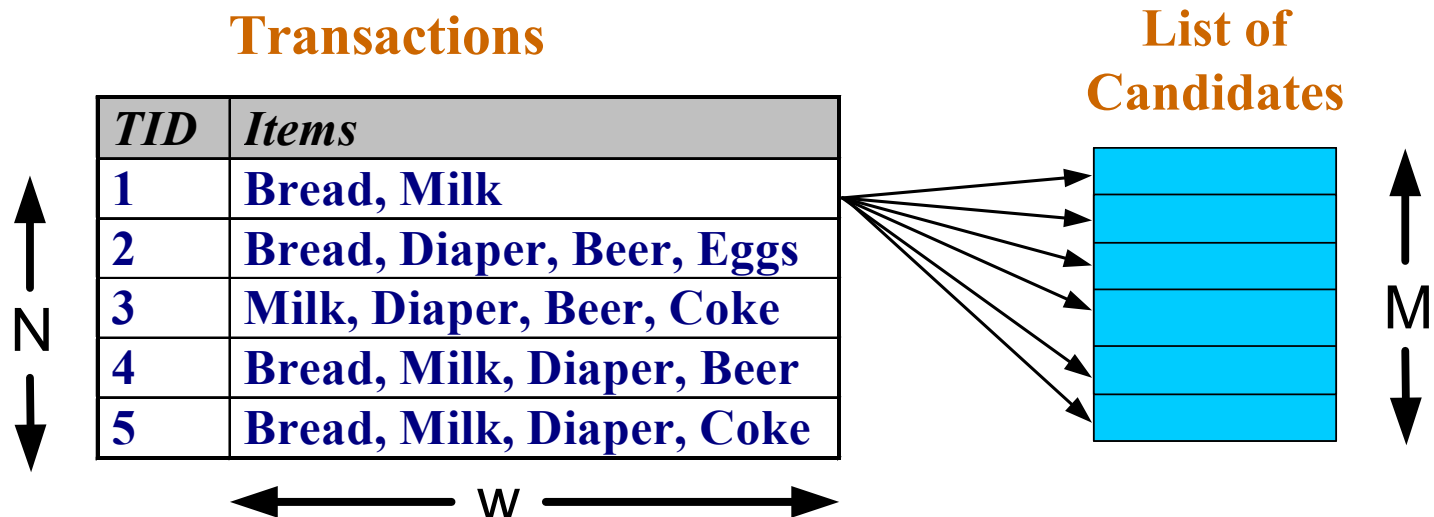


**Given d items, there
are 2^d possible
candidate itemsets**

Frequent Itemset Generation

■ Brute-force approach:

- ★ Each itemset in the lattice is a **candidate** frequent itemset
- ★ Count the support of each candidate by scanning the database



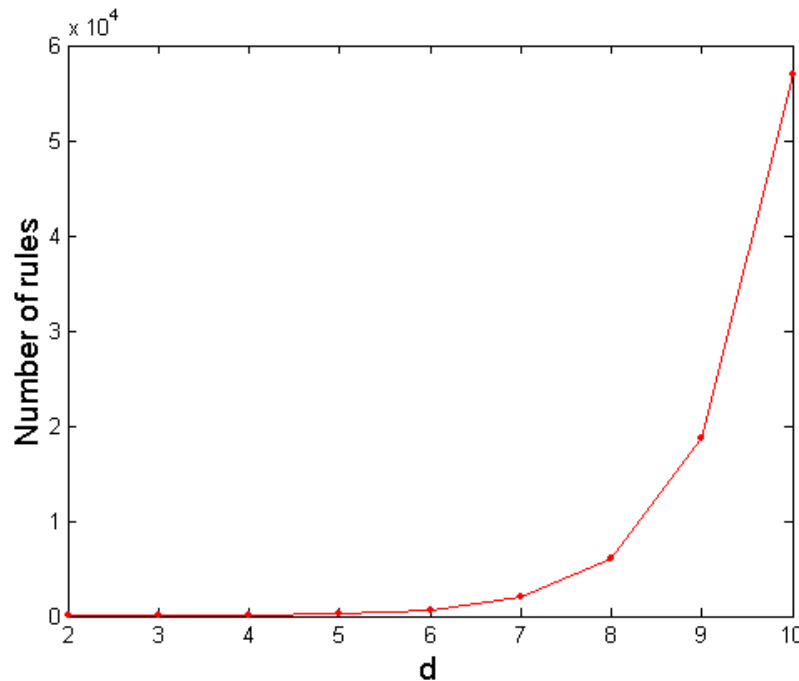
- ★ Match each transaction against every candidate
- ★ Complexity $\sim O(NMw) \Rightarrow$ **Expensive since $M = 2^d$!!!**

Computational Complexity

■ Given d unique items:

★ Total number of itemsets = 2^d

★ Total number of possible association rules:



$$R = \sum_{k=1}^{d-1} \left[\binom{d}{k} \times \sum_{j=1}^{d-k} \binom{d-k}{j} \right]$$
$$= 3^d - 2^{d+1} + 1$$

If $d=6$, $R = 602$ rules

Frequent Itemset Generation Strategies

- Reduce the **number of candidates** (M)
 - ★ Complete search: $M=2^d$
 - ★ Use pruning techniques to reduce M
- Reduce the **number of transactions** (N)
 - ★ Reduce size of N as the size of itemset increases
- Reduce the **number of comparisons** (NM)
 - ★ Use efficient data structures to store the candidates or transactions
 - ★ No need to match every candidate against every transaction

Reducing Number of Candidates

■ Apriori principle:

★ If an itemset is frequent, then all of its subsets must also be frequent

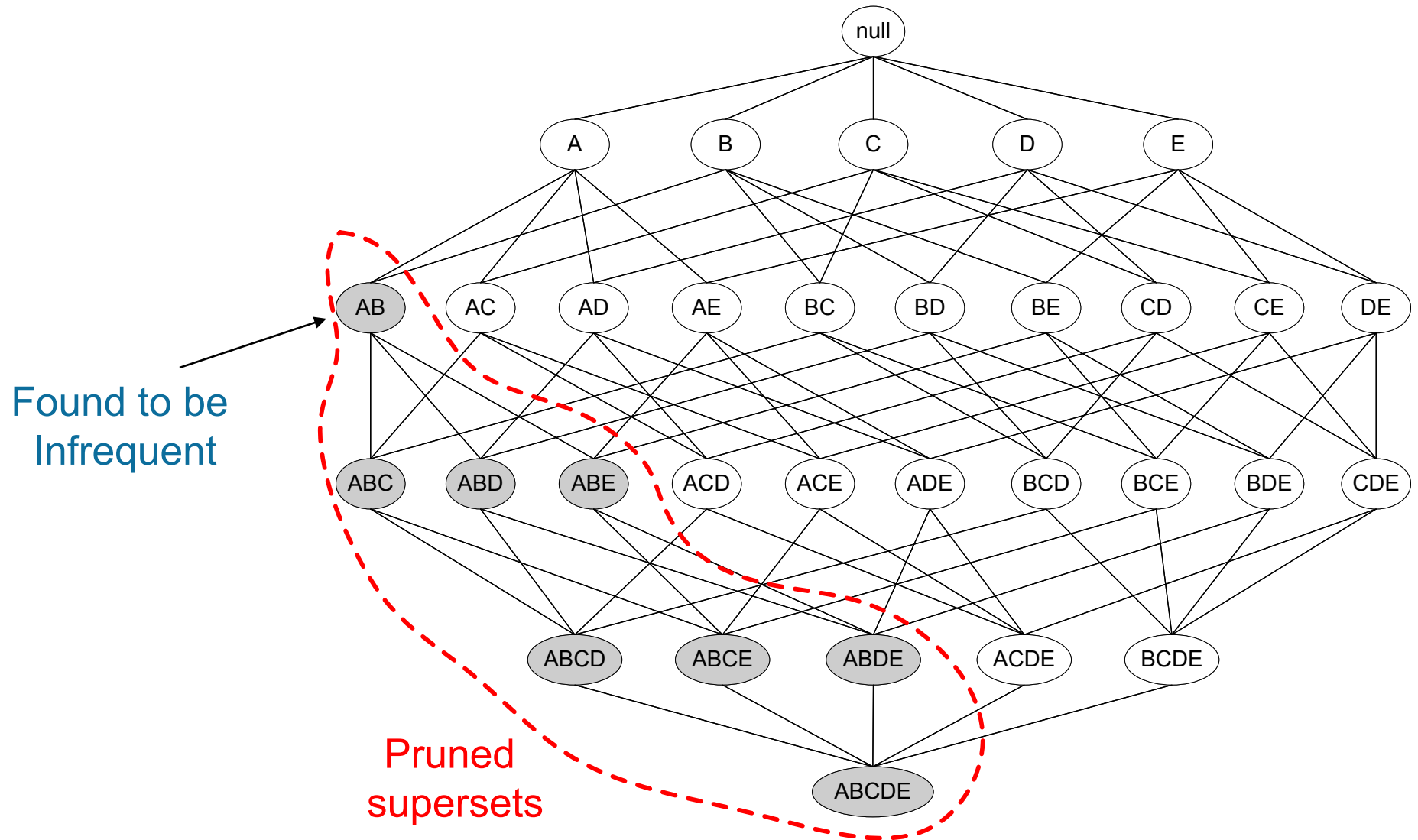
■ Apriori principle holds due to the following property of the support measure:

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

★ Support of an itemset never exceeds the support of its subsets

★ This is known as the **anti-monotone** property of support

Illustrating Apriori Principle



Illustrating Apriori Principle

Item	Count
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1

Items (1-itemsets)



Itemset	Count
{Bread,Milk}	3
{Bread,Beer}	2
{Bread,Diaper}	3
{Milk,Beer}	2
{Milk,Diaper}	3
{Beer,Diaper}	3

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)



Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	3



Minimum Support = 3

If every subset is considered,

$${}^6C_1 + {}^6C_2 + {}^6C_3 = 41$$

With support-based pruning,

$$6 + 6 + 1 = 13$$

Apriori Algorithm

■ Method:

- ★ Let $k=1$
- ★ Generate frequent itemsets of length 1
- ★ Repeat until no new frequent itemsets are identified
 - ✓ Generate length $(k+1)$ candidate itemsets from length k frequent itemsets
 - ✓ Prune candidate itemsets containing subsets of length k that are infrequent
 - ✓ Count the support of each candidate by scanning the DB
 - ✓ Eliminate candidates that are infrequent, leaving only those that are frequent

Rule Generation

- Given a frequent itemset L , find all non-empty subsets $f \subset L$ such that $f \rightarrow L - f$ satisfies the minimum confidence requirement

★ If $\{A,B,C,D\}$ is a frequent itemset, candidate rules:

✓ $ABC \rightarrow D,$	$ABD \rightarrow C,$	$ACD \rightarrow B,$	$BCD \rightarrow A,$
$A \rightarrow BCD,$	$B \rightarrow ACD,$	$C \rightarrow ABD,$	$D \rightarrow ABC$
$AB \rightarrow CD,$	$AC \rightarrow BD,$	$AD \rightarrow BC,$	$BC \rightarrow AD,$
$BD \rightarrow AC,$	$CD \rightarrow AB,$		

- If $|L| = k$, then there are $2^k - 2$ candidate association rules (ignoring $L \rightarrow \emptyset$ and $\emptyset \rightarrow L$)

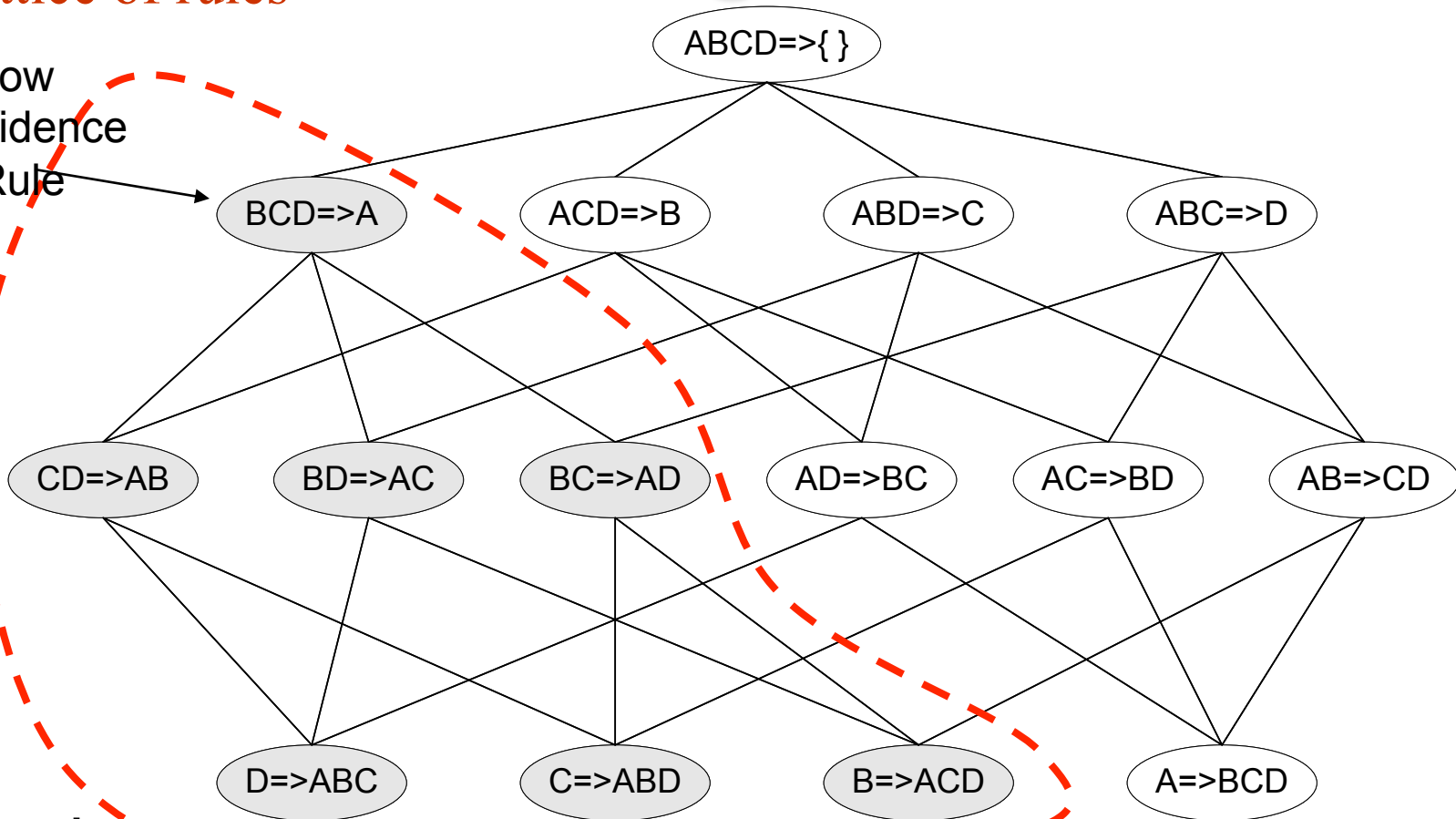
Rule Generation

- How to efficiently generate rules from frequent itemsets?
 - ★ In general, confidence does not have an anti-monotone property
 $c(ABC \rightarrow D)$ can be larger or smaller than $c(AB \rightarrow D)$
 - ★ But confidence of rules generated from the same itemset has an anti-monotone property
 - ✓ e.g., $L = \{A, B, C, D\}$:
 - $c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$
 - ✓ Confidence is anti-monotone w.r.t. number of items on the RHS of the rule

Rule Generation for Apriori Algorithm

Lattice of rules

Low
Confidence
Rule



Pruned
Rules

Conclusions

- We have learned about the 3 major data mining/machine learning tasks
- Almost all data mining research is in these 3 areas, or is a minor extension of one or more of them.