OntarioTech Science

Kourosh Davoudi
kourosh@uoit.ca

Review

CSCI 4150U: Data Mining

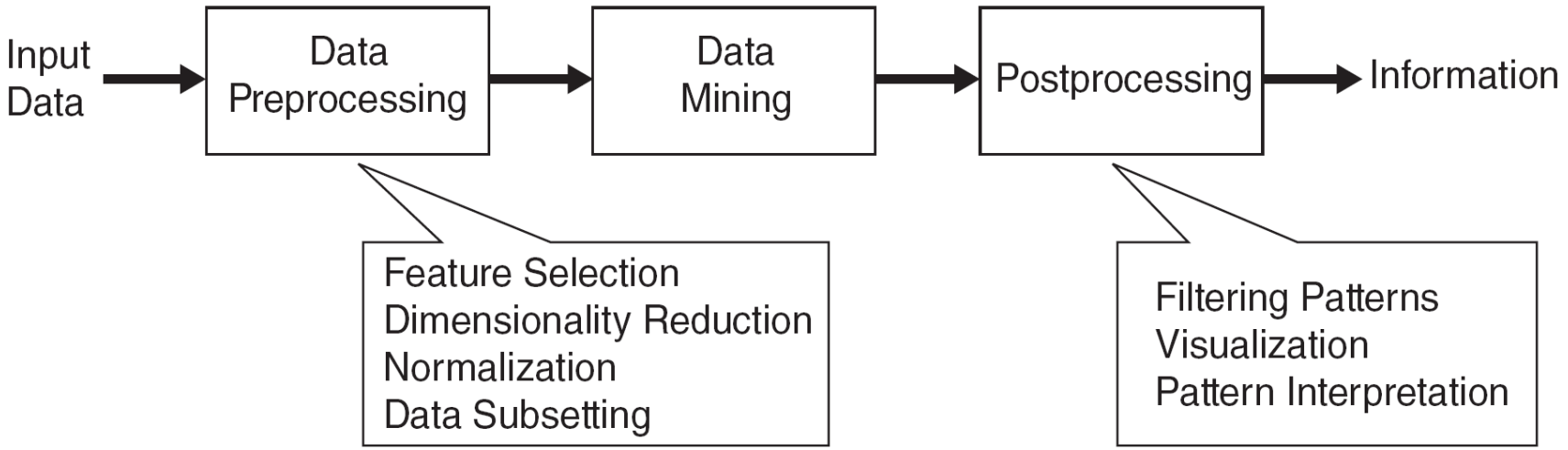
Course Learning Outcomes

- Data
- Data Exploratory Analysis
- Classification
- Clustering
- Anomaly Detection
- Association Rule Mining

Data

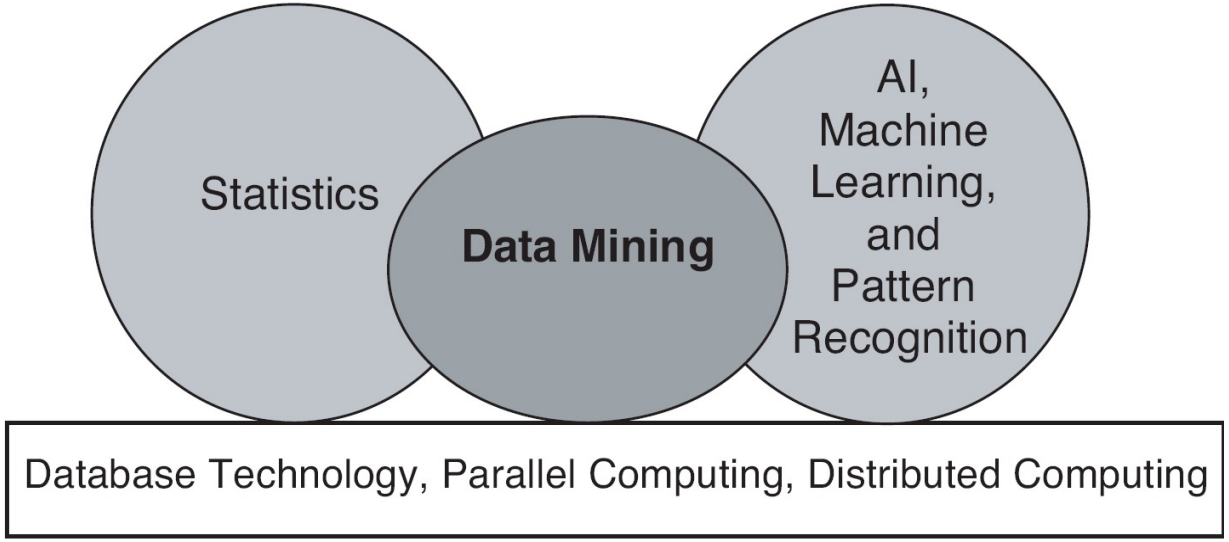
What is Data Mining?

- Many Definitions
  - Non-trivial extraction of implicit, previously unknown and potentially useful information from data

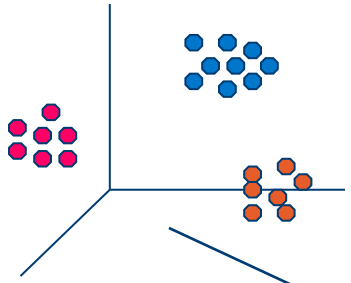


Origins of Data Mining

- Draws ideas from machine learning/AI, pattern recognition, statistics, and database systems



Data Mining Tasks …

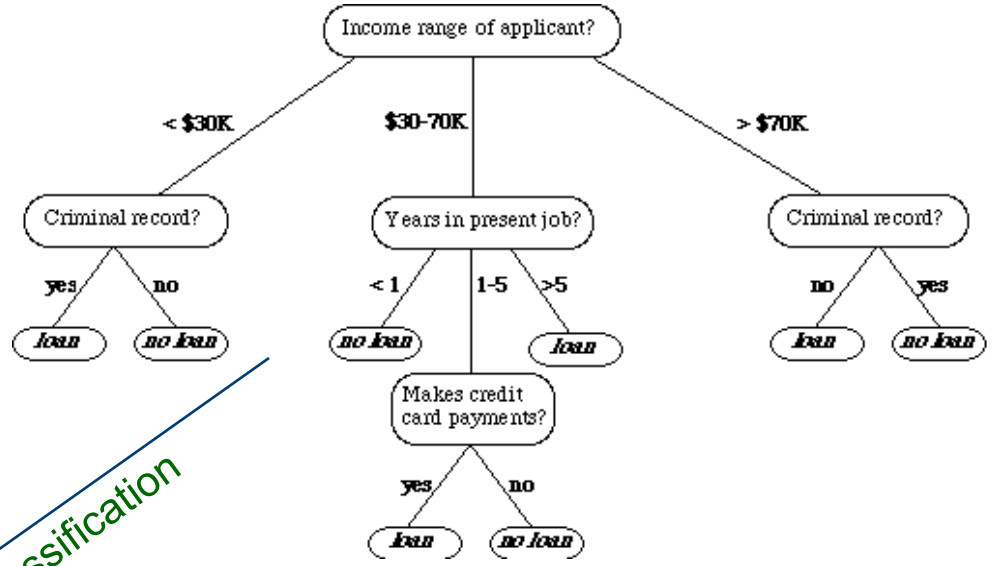


Clustering

Data

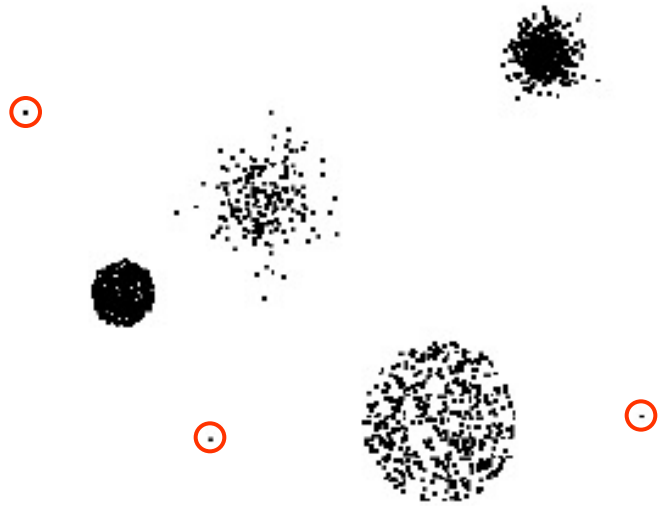
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes
11	No	Married	60K	No
12	Yes	Divorced	220K	No
13	No	Single	85K	Yes
14	No	Married	75K	No
15	No	Single	90K	Yes

classification



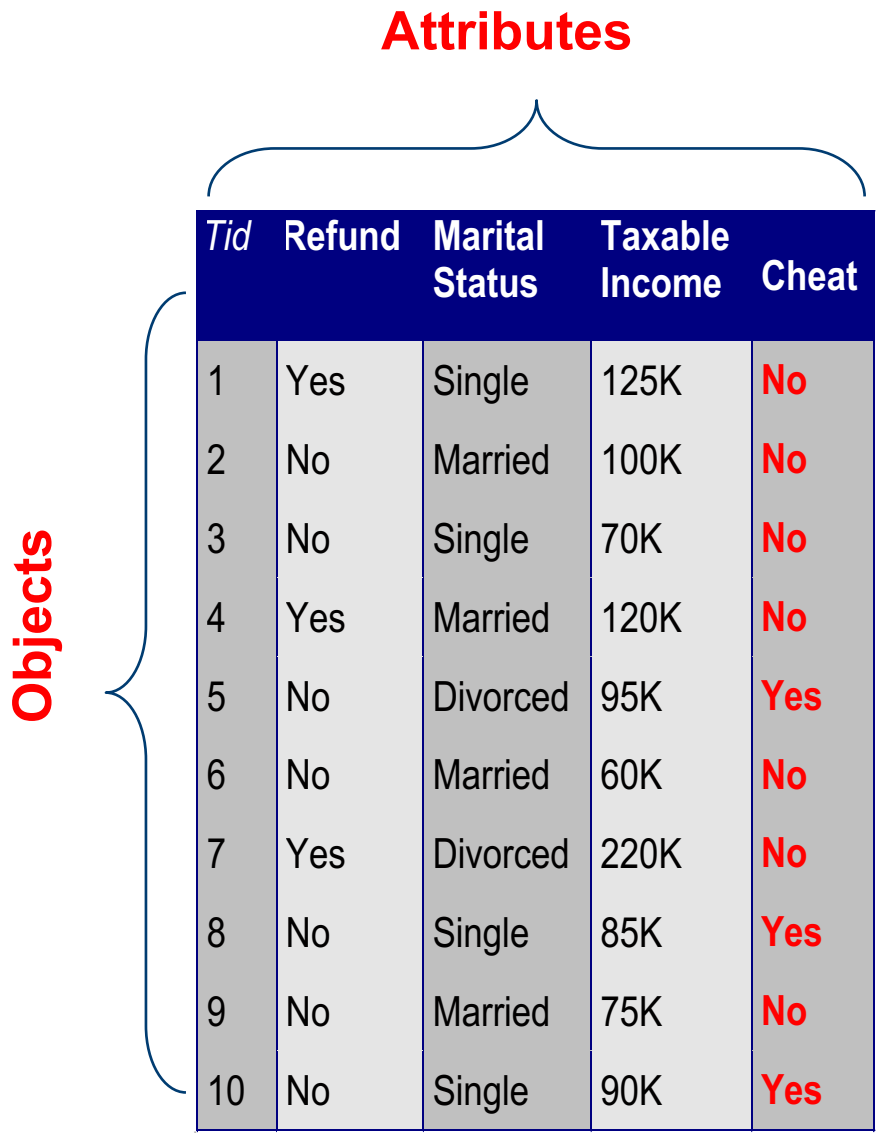
Association Rules

Anomaly Detection



What is Data?

- Collection of data objects and their attributes



Types of Attributes

- **Nominal**: Meaningless, barely enough to distinguish one object from another
  - Examples: ID numbers, eye color
- **Ordinal**: "Order" has meaning
  - Examples: height {tall, medium, short}
- **Interval**: "Difference" has meaning
  - Examples: calendar dates, temperatures in Celsius or Fahrenheit.
- **Ratio**: "Ratio" has meaning
  - Examples: counts

Data Types

Record Data

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Data Matrix

Projection of x Load	Projection of y load	Distance	Load	Thickness
10.23	5.27	15.22	2.7	1.2
12.65	6.25	16.22	2.2	1.1

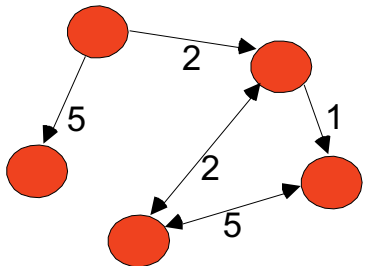
Document Data

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

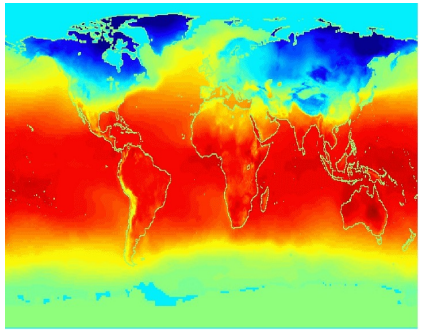
Transaction Data

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Graph Data

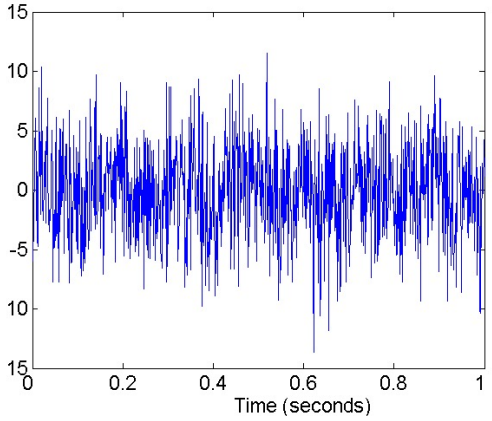


Ordered Data

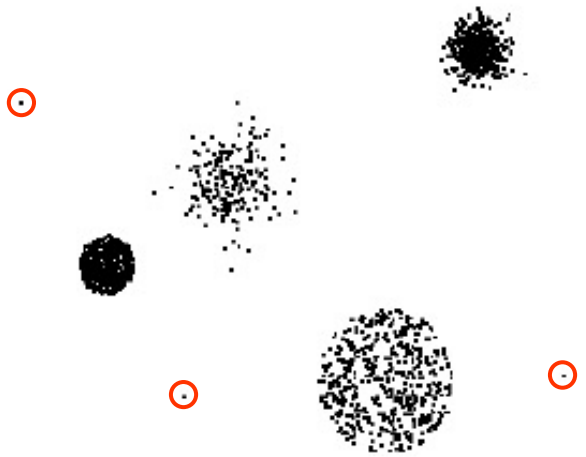


Data Quality

Noise



outliers



Duplicate data

Missing Values

Similarity and Dissimilarity Measures

Euclidean Distance

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

Minkowski Distance

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r}$$

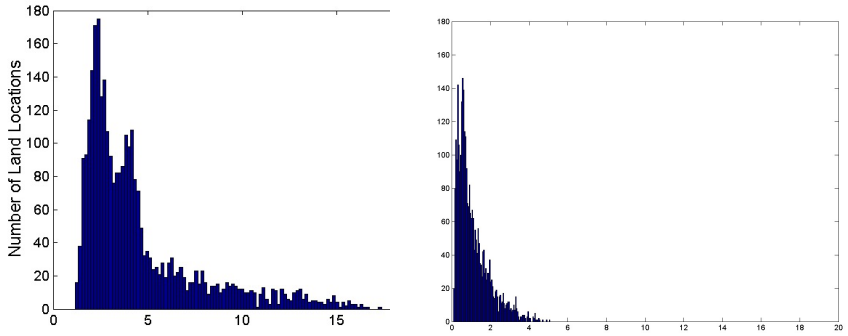
Mahalanobis Distance

$$d(x, y) = (x - y)^T \Sigma^{-1} (x - y)$$

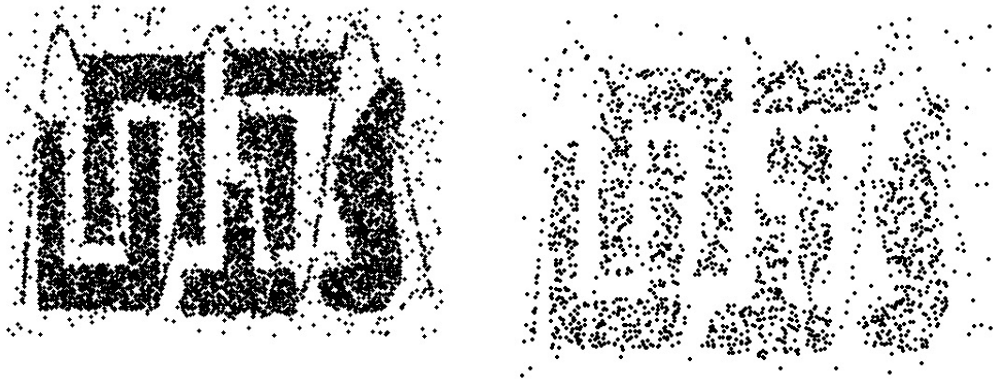
Others:

- Cosine Similarity
- Simple Matching (SMC)
- Jaccard (J) Coefficients
- Extended Jaccard Coefficient

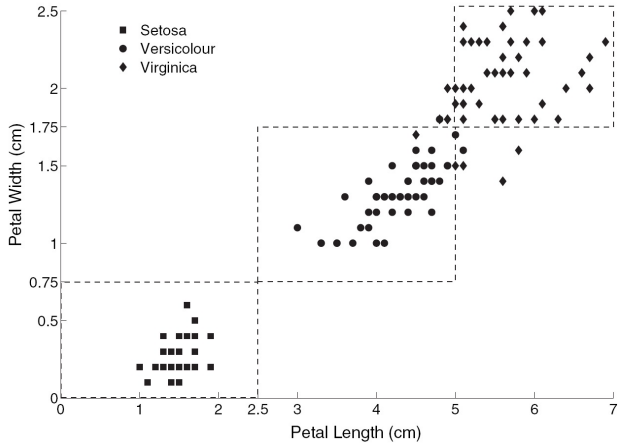
Data Preprocessing



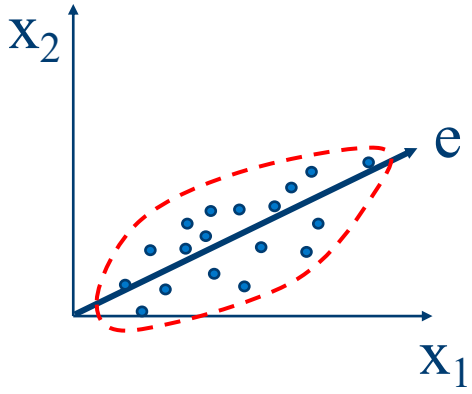
Aggregation



Sampling



Discretization

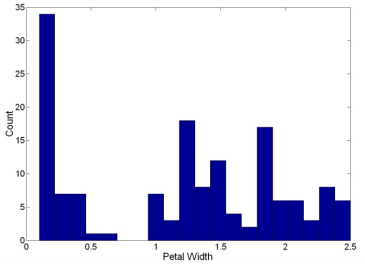


Dimensionality Reduction

Feature subset selection

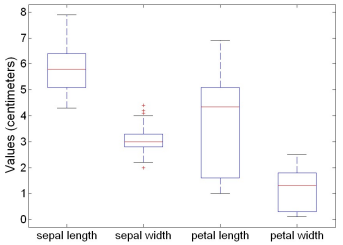
Data Exploratory Analysis

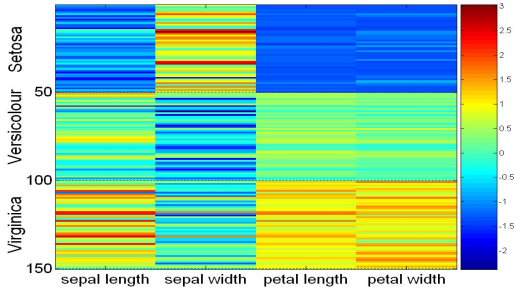
Techniques Used In Data Exploration

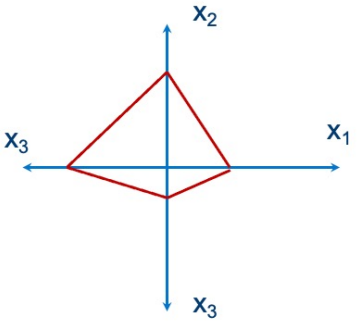


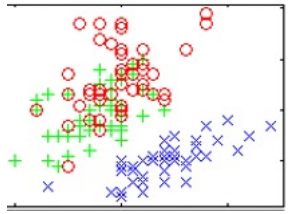
In our discussion of data exploration, we focus on

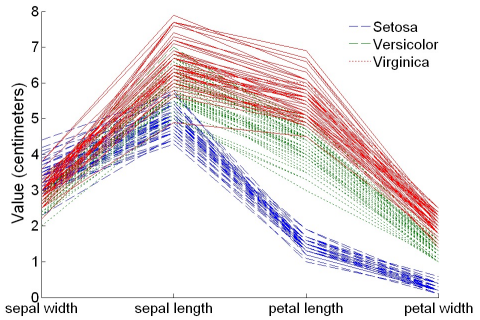
- Summary statistics
  - Mean/Standard Deviation/Frequency/Mode/Percentiles
- Visualization
  - Histogram
  - Box Plot
  - Scatter Plot
  - Contour Plot
  - Data Matrix/Correlation Matrix
  - Parallel Coordinates
  - Start Plots
- Online Analytical Processing (OLAP)

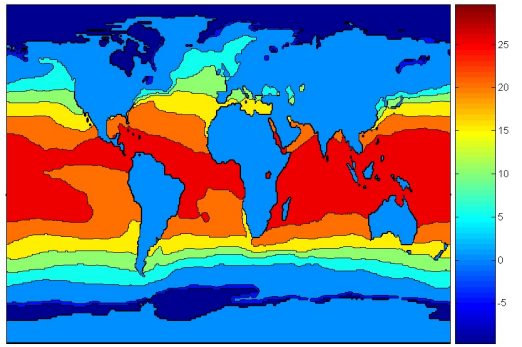








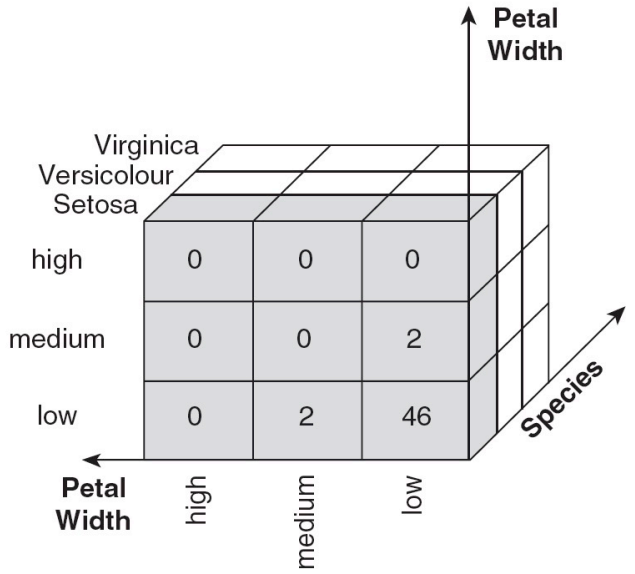




On-Line Analytical Processing (OLAP)

- Relational databases put data into tables, while OLAP uses a multidimensional array representation.
  - Such representations of data previously existed in statistics and other fields
- There are a number of data analysis and data exploration operations that are easier with such a data representation.

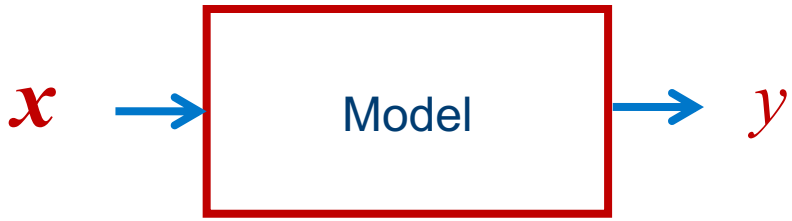
Petal Length	Petal Width	Species Type	Coun
low	low	Setosa	46
low	medium	Setosa	2
medium	low	Setosa	2
medium	medium	Versicolour	43
medium	high	Versicolour	3
medium	high	Virginica	3
high	medium	Versicolour	2
high	medium	Virginica	3
high	high	Versicolour	2
high	high	Virginica	44



Classification

Classification: Definition

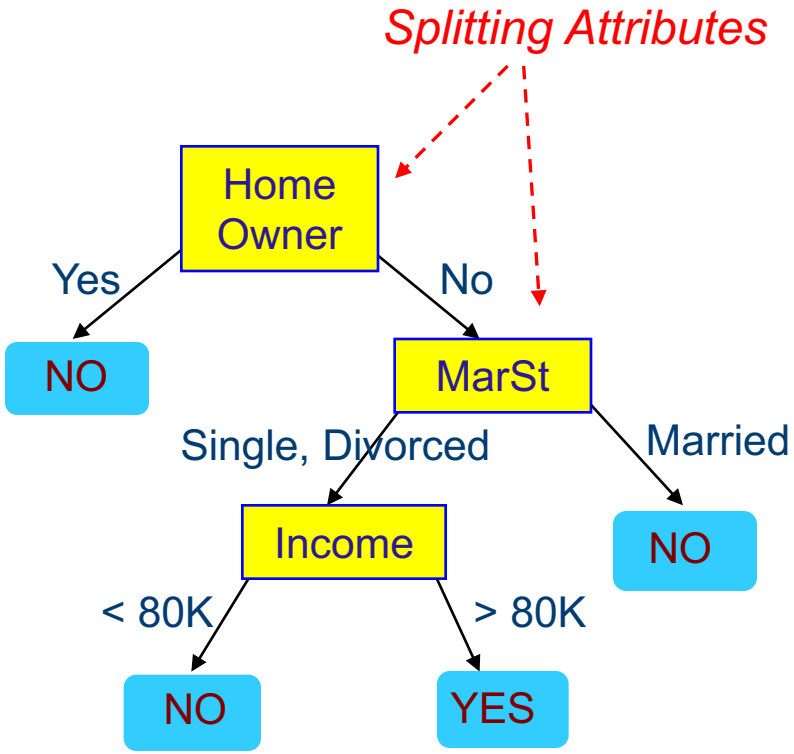
- Learn a model that maps each attribute set $\boldsymbol{x}$ into one of the predefined class labels y



Decision Tree

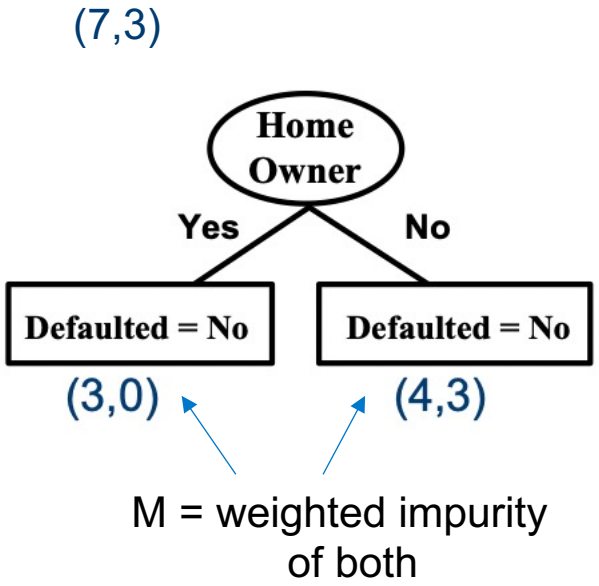
	categorical	categorical	continuous	class
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



Model: Decision Tree

Finding the Best Split



1. Compute impurity measure (P) before splitting
2. Compute impurity measure (M) after splitting
   - Compute impurity measure of each child node
   - M is the weighted impurity of child nodes
3. Choose the attribute test condition that produces the highest gain

Gain = P - M

or equivalently, lowest impurity measure after splitting (M)

Measures of Node Impurity

- Gini Index

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

Where $p_i(t)$ is the probability of class i at node t, and c is the total number of classes

- Entropy

$$Entropy = -\sum_{i=0}^{c-1} p_i(t)\ log_2 p_i(t)$$

- Misclassification error

$$Classification\ error = 1 - \max[p_i(t)]$$

Metrics for Performance Evaluation

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Other Measures

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

$$\text{Precision (p)} = \frac{a}{a+c}$$

$$\text{Recall (r)} = \frac{a}{a+b}$$

$$\text{F - measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

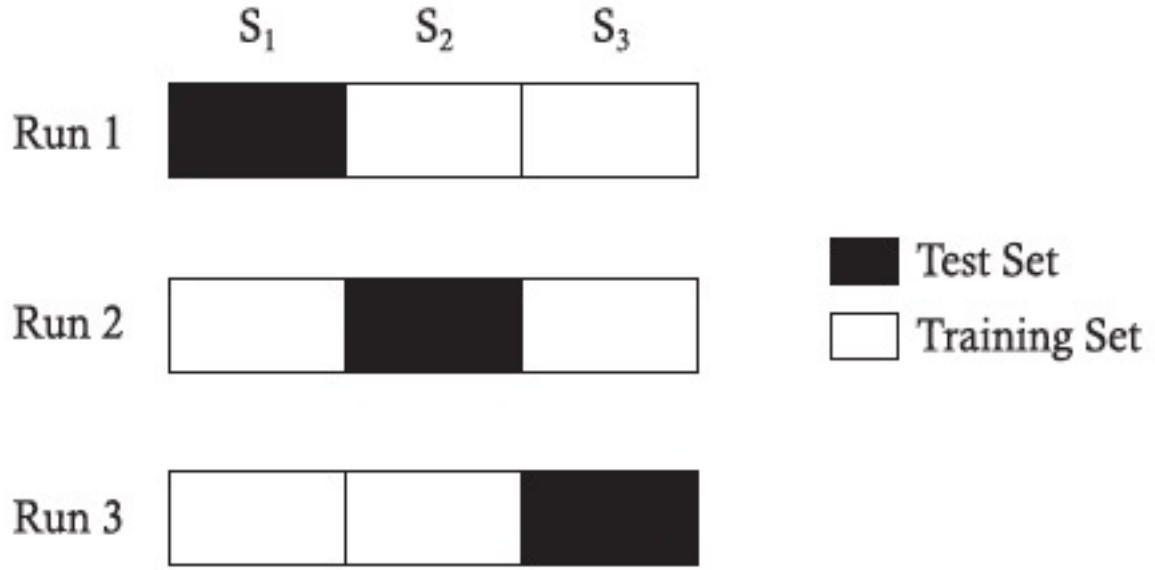
Cross-validation

- 3-fold cross-validation

Performance 1

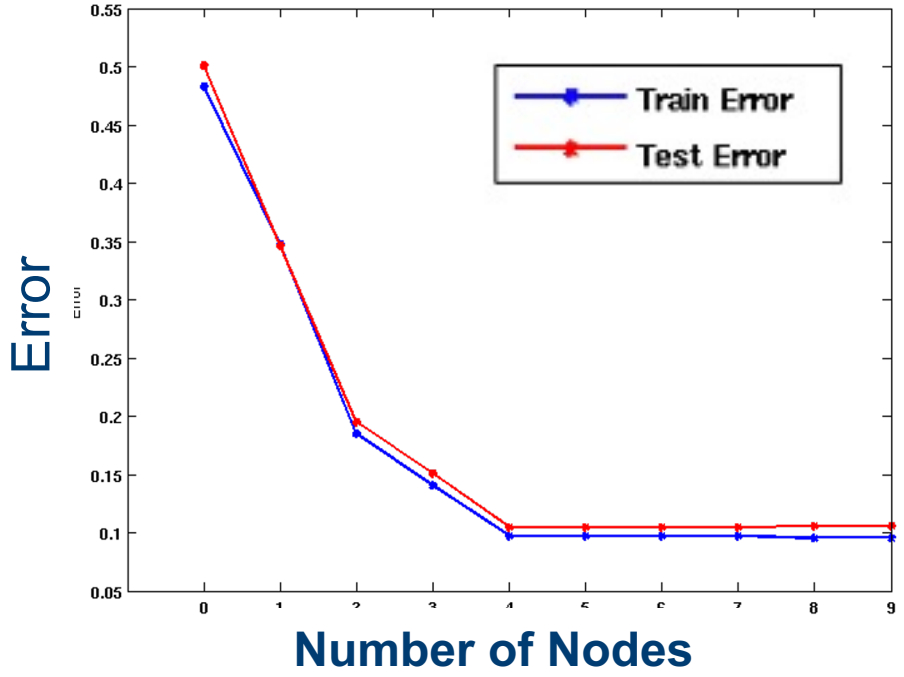
Performance 2

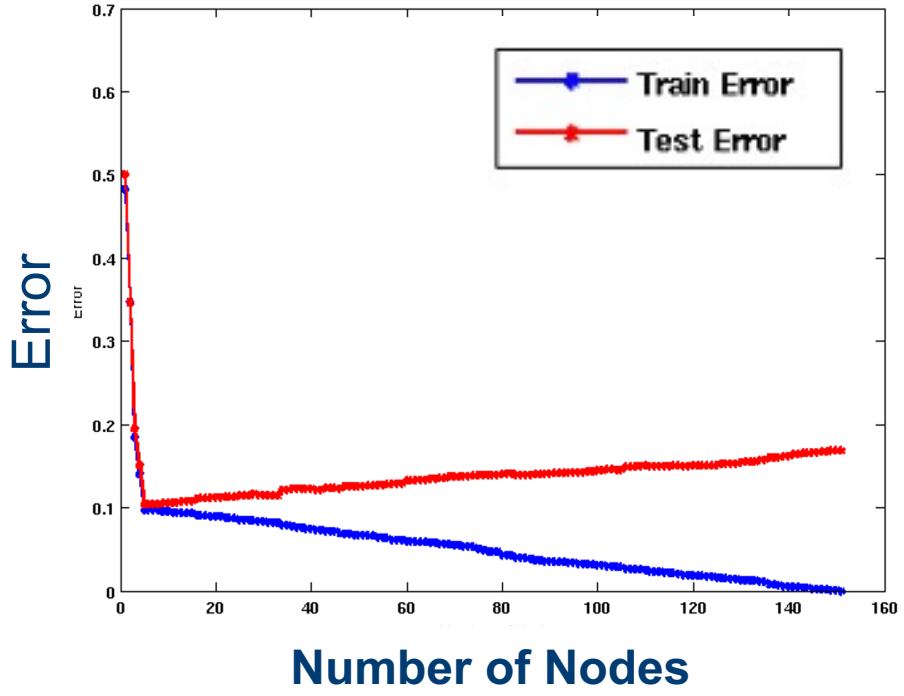
Performance 3



Final Performance = Average of All Performance

Model Overfitting

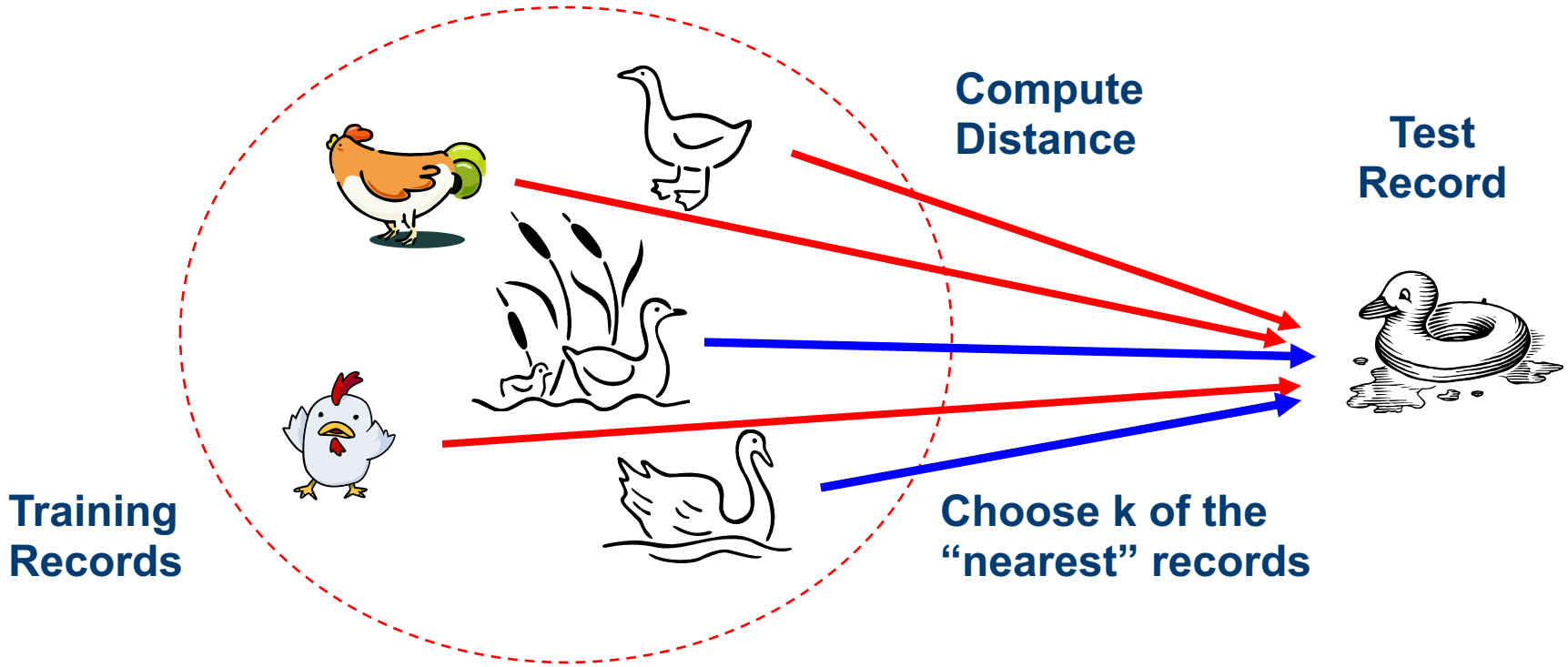




- As the model becomes more and more complex, test errors can start increasing even though training error may be decreasing
- Underfitting: when model is too simple, both training and test errors are large
- Overfitting: when model is too complex, training error is small but test error is large

Nearest Neighbor Classifiers (kNN)

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



Naïve Bayes Classifier

- Assume independence among attributes X_i when class is given:

$$\mathrm{P}(X_1, X_2, \ldots, X_d \,|C_j) = \mathrm{P}(X_1|\, C_j)\, \mathrm{P}(X_2|\, C_j) \ldots\, \mathrm{P}(X_d|\, C_j)$$

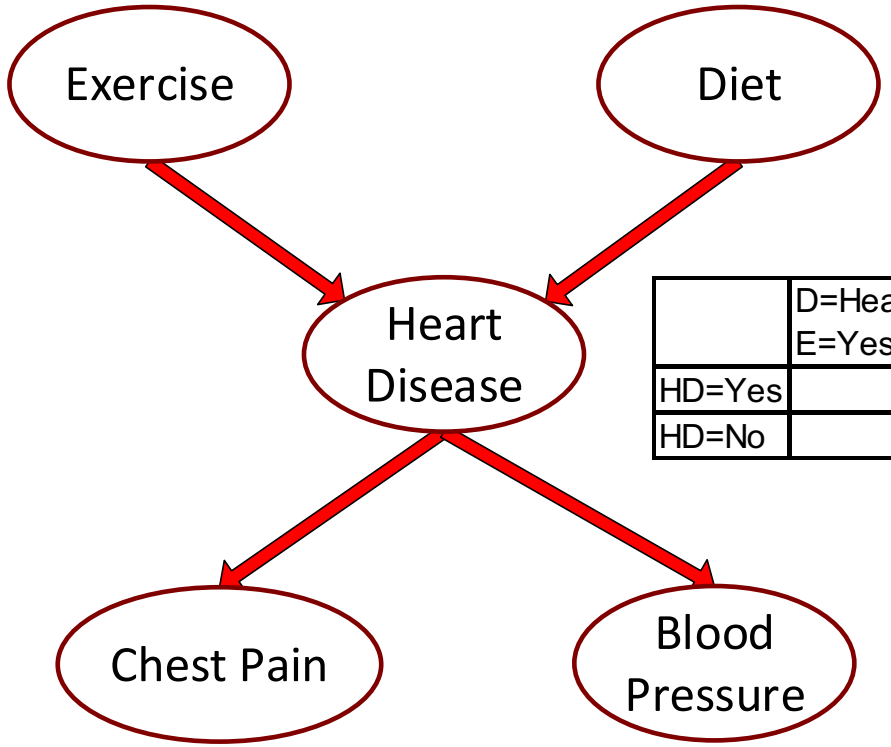
- Now we can estimate $\mathrm{P}(X_i|\, C_j)$ for all X_i and C_j combinations from the training data

- New point is classified to C_j if $\mathrm{P}(C_j)\, \Pi\, \mathrm{P}(X_i|\, C_j)$ is maximal.

Example of Bayesian Belief Network

Exercise=Yes	0.7
Exercise=No	0.3

Diet=Healthy	0.25
Diet=Unhealthy	0.75



	D=Healthy E=Yes	D=Healthy E=No	D=Unhealthy E=Yes	D=Unhealthy E=No
HD=Yes	0.25	0.45	0.55	0.75
HD=No	0.75	0.55	0.45	0.25

	HD=Yes	HD=No
CP=Yes	0.8	0.01
CP=No	0.2	0.99

	HD=Yes	HD=No
BP=High	0.85	0.2
BP=Low	0.15	0.8

Bagging

- Sampling with replacement

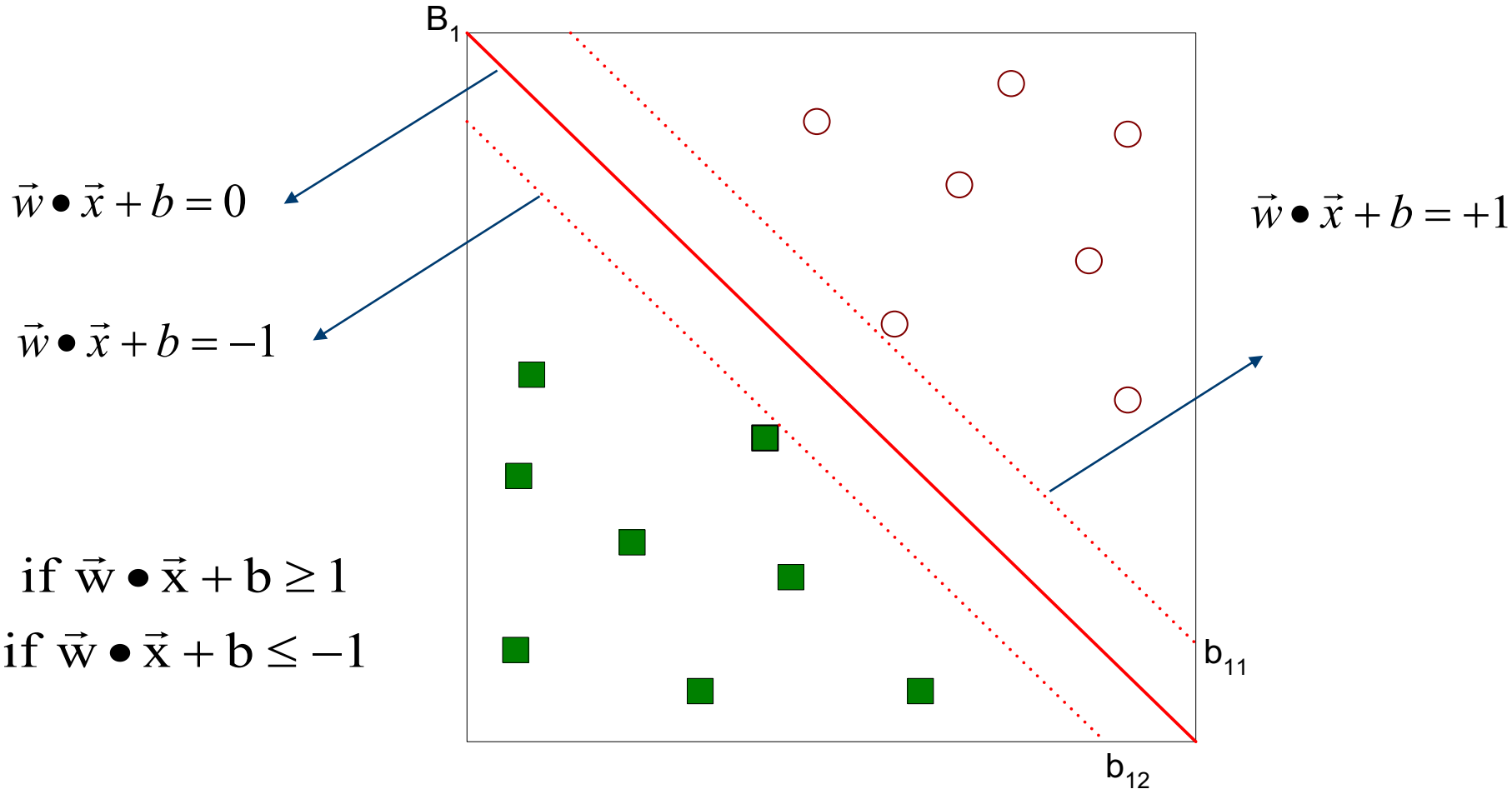
Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

- Build classifier on each bootstrap sample

Boosting

- An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records:
  - Initially, all N records are assigned equal weights
  - Unlike bagging, weights may change at the end of each boosting round

Support Vector Machines



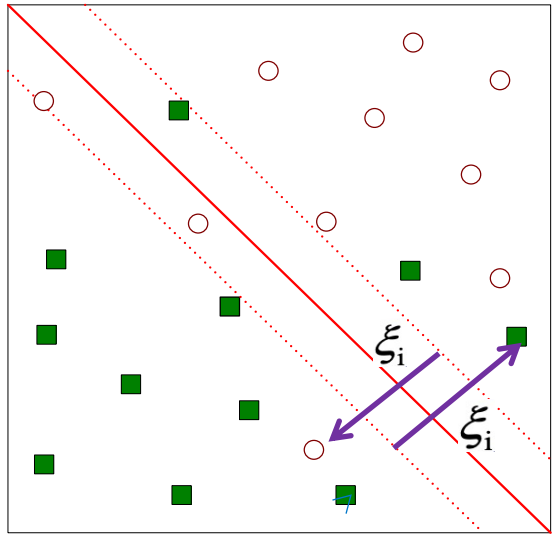
$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x} + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x} + b \leq -1 \end{cases}$$

$$\text{Margin} = \frac{2}{\| \vec{w} \|}$$

Support Vector Machines

$\vec{w} \bullet \vec{x} + b = +1$

$\vec{w} \bullet \vec{x} + b = -1$



- What if the problem is not linearly separable?
  - Introduce slack variables
    - Need to minimize:

$$L(w) = \frac{\| \vec{w} \|^2}{2} + C\left(\sum_{i=1}^{N} \xi_i^k \right)$$

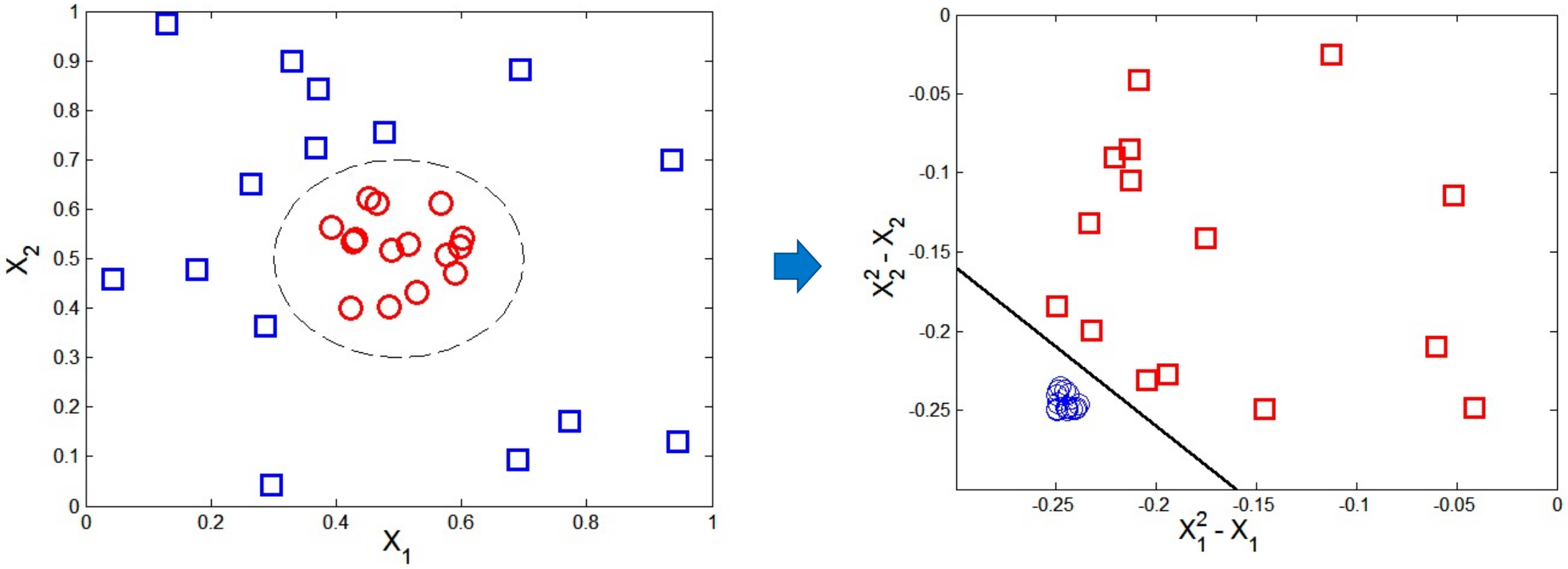
    - Subject to:

$$y_i = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x}_i + b \geq 1 - \xi_i \\ -1 & \text{if } \vec{w} \bullet \vec{x}_i + b \leq -1 + \xi_i \end{cases}$$

    - If hyperparameter k is usually is 1 or 2

Nonlinear Support Vector Machines

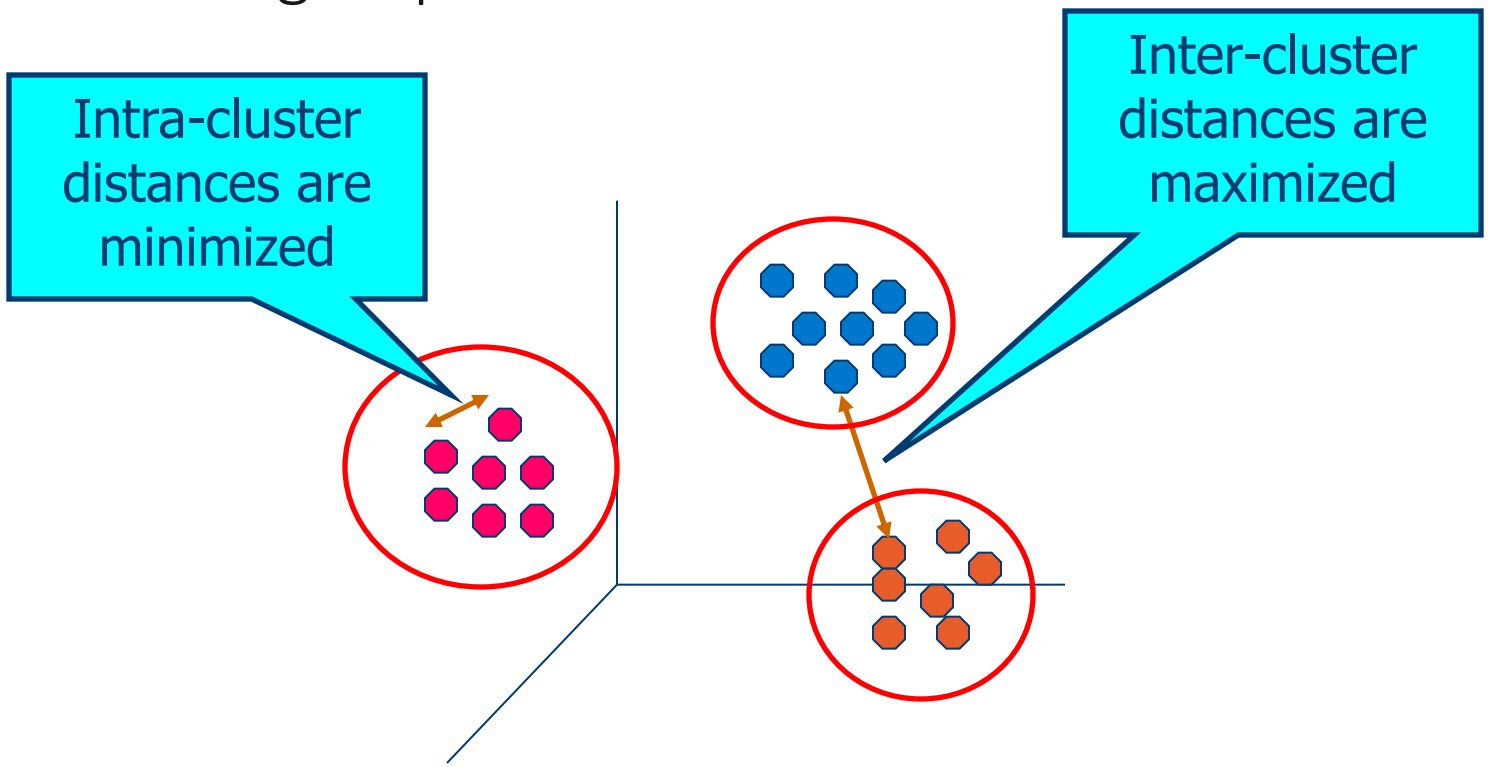
- What if decision boundary is not linear?



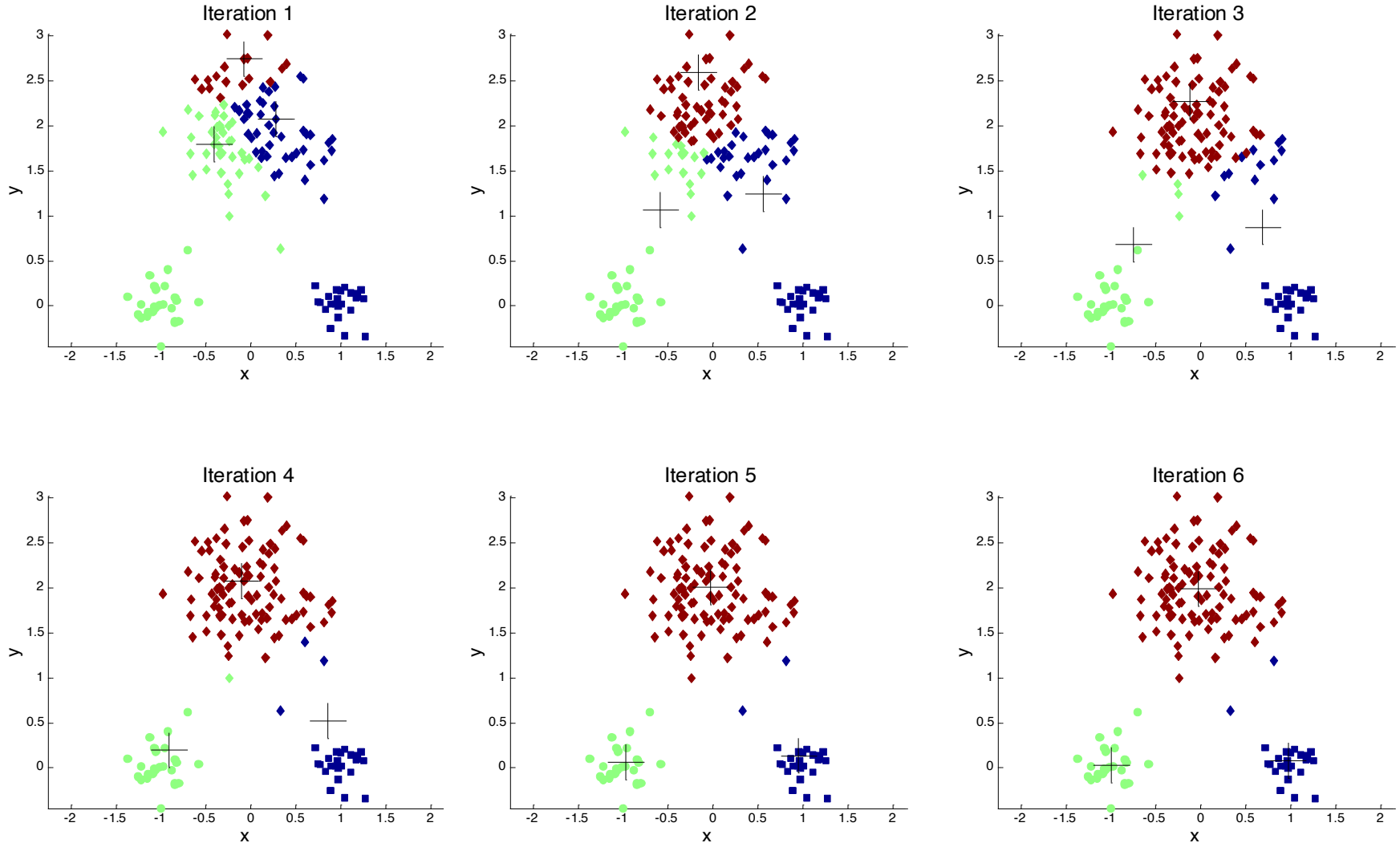
Clustering

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

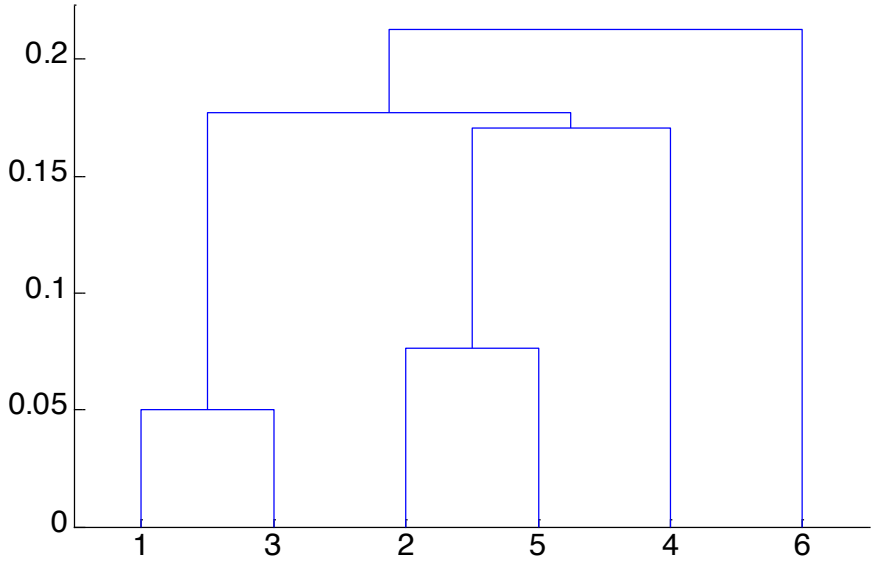


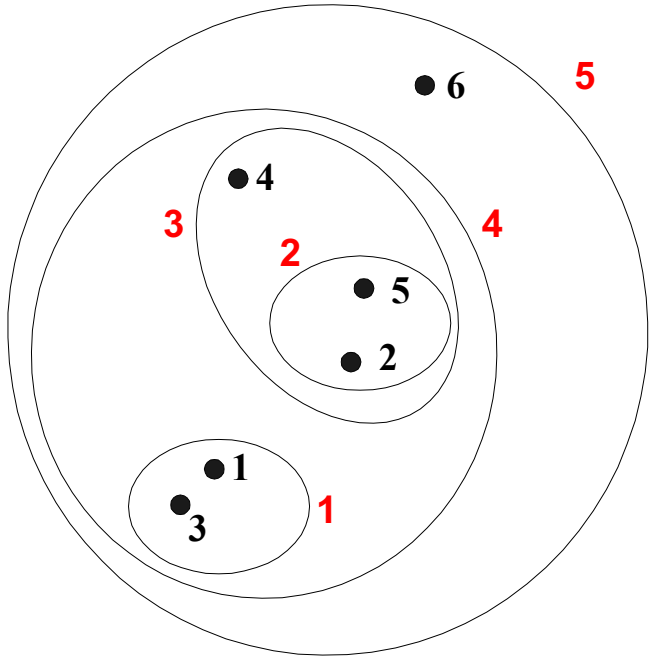
K-means Clustering



Hierarchical Clustering

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





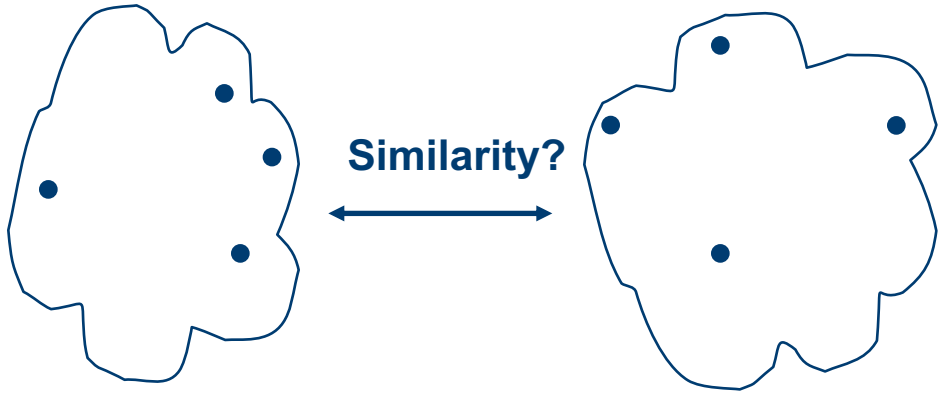
Agglomerative Clustering Algorithm

- Most popular hierarchical clustering technique

- Basic algorithm is straightforward:
  1. Compute the proximity matrix
  2. Let each data point be a cluster
  3. **Repeat**
  4. Merge the two closest clusters
  5. Update the proximity matrix
  6. **Until** only a single cluster remains

- Key operation is the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms

How to Define Inter-Cluster Distance



- MIN
- MAX
- Group Average
- Distance Between Centroids

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

Proximity Matrix

DBSCAN

- DBSCAN is a density-based algorithm.
  - Density = number of points within a specified radius (Eps)
  - A point is a core point if it has at least a specified number of points (MinPts) within Eps (distance =< Eps)
    - These are points that are at the interior of a cluster
    - Counts the point itself
  - A border point is not a core point, but is in the neighborhood of a core point
  - A noise point is any point that is not a core point or a border point

DBSCAN Algorithm

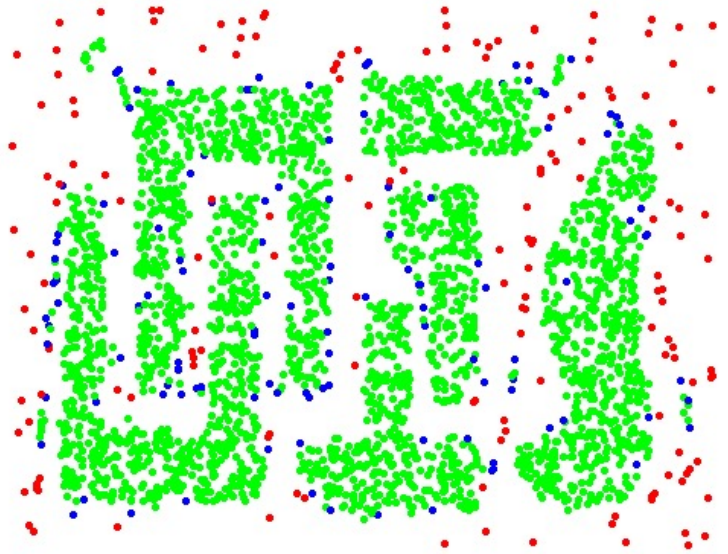
- Eliminate noise points
- Perform clustering on the remaining points

Algorithm 8.4 DBSCAN algorithm.

1: Label all points as core, border, or noise points.
2: Eliminate noise points.
3: Put an edge between all core points that are within *Eps* of each other.
4: Make each group of connected core points into a separate cluster.
5: Assign each border point to one of the clusters of its associated core points.

DBSCAN: Core, Border and Noise Points

Original Points

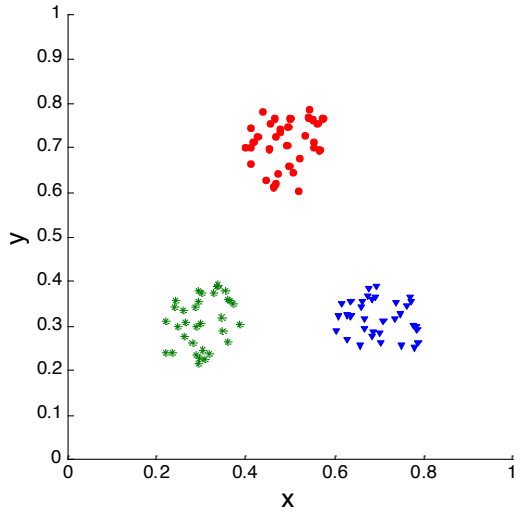


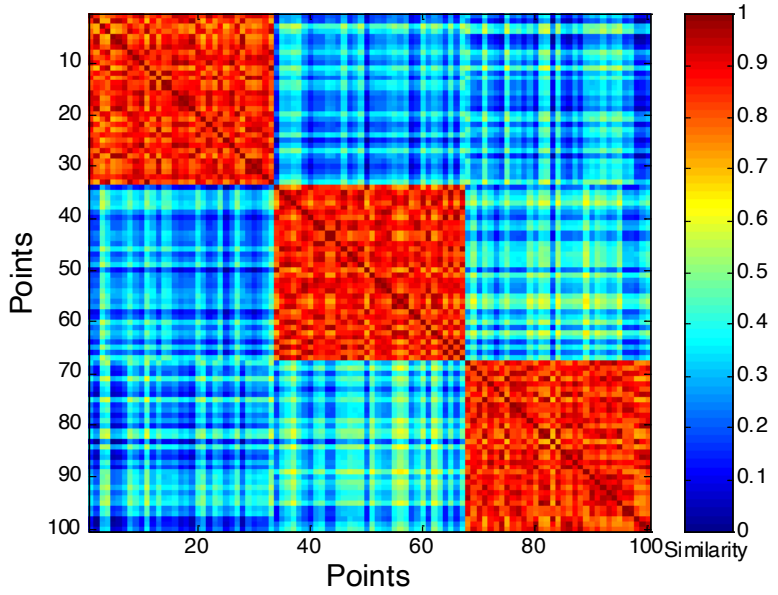
Point types: core, border and noise

Eps = 10, MinPts = 4

Cluster Validity

Correlation of ideal similarity and proximity matrices

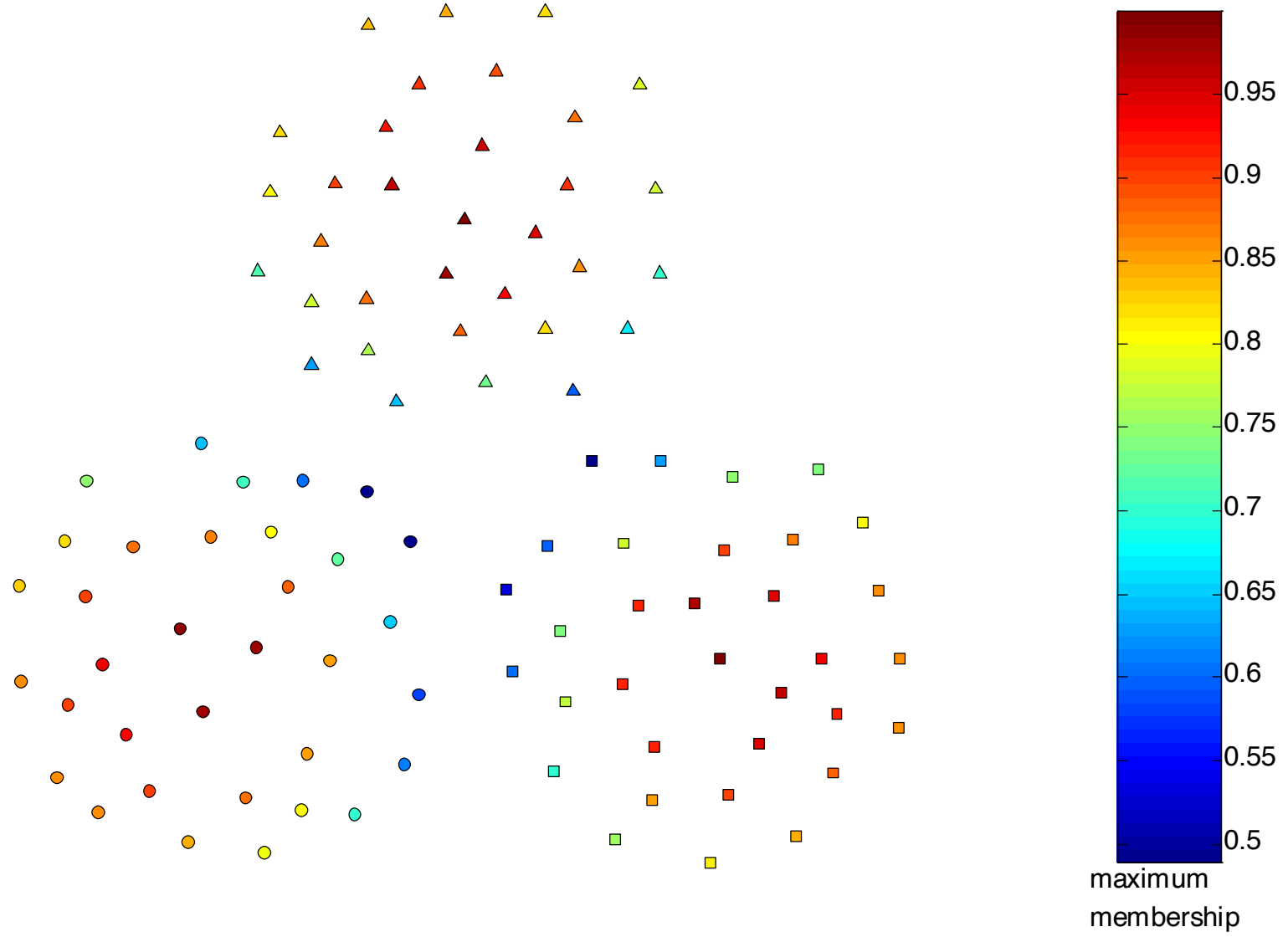




Internal Measures: SSE

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

Fuzzy K-means Applied to Sample Data



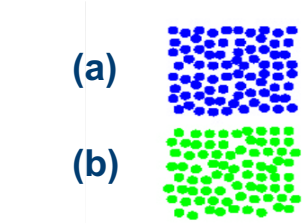
CURE Algorithm

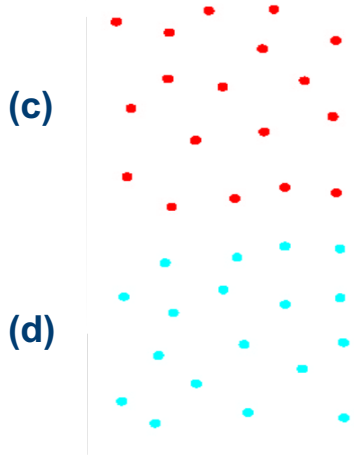
- "Shrink" representative points toward the center of the cluster by a factor, α

- Shrinking representative points toward the center helps avoid problems with noise and outliers

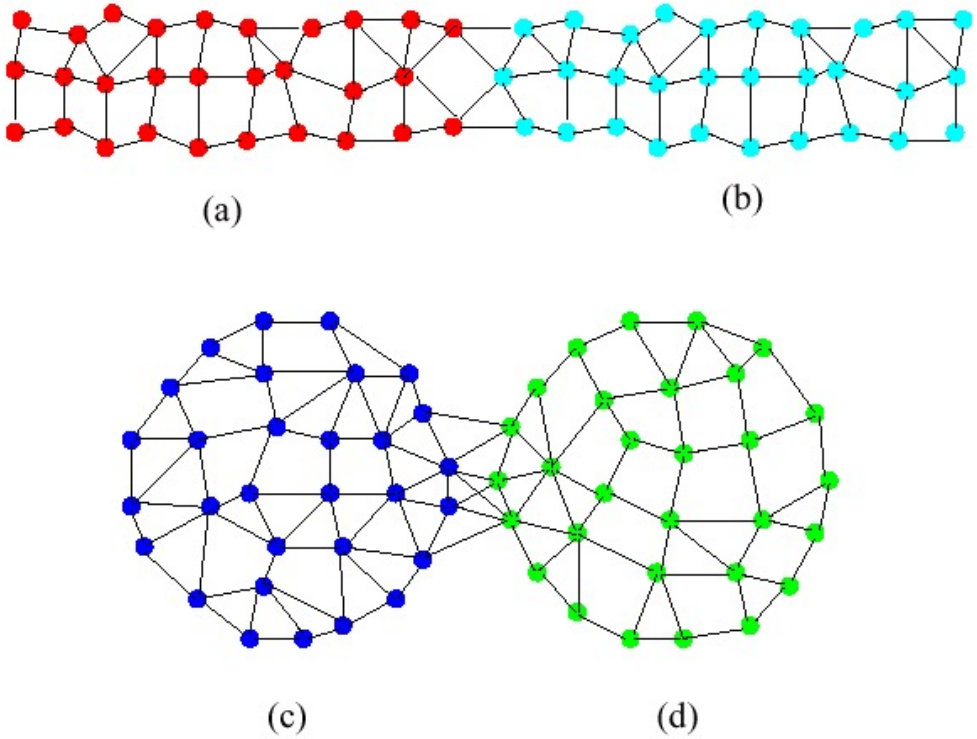
- Cluster similarity is the similarity of the closest pair of representative points from different clusters

Chameleon: Clustering Using Dynamic Modeling





Closeness schemes will merge (a) and (b)



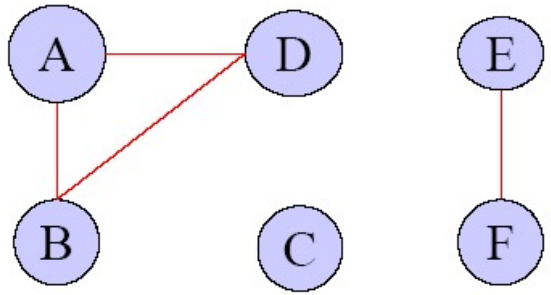
Average connectivity schemes will merge (c) and (d)

Example: Jarvis-Patrick Clustering

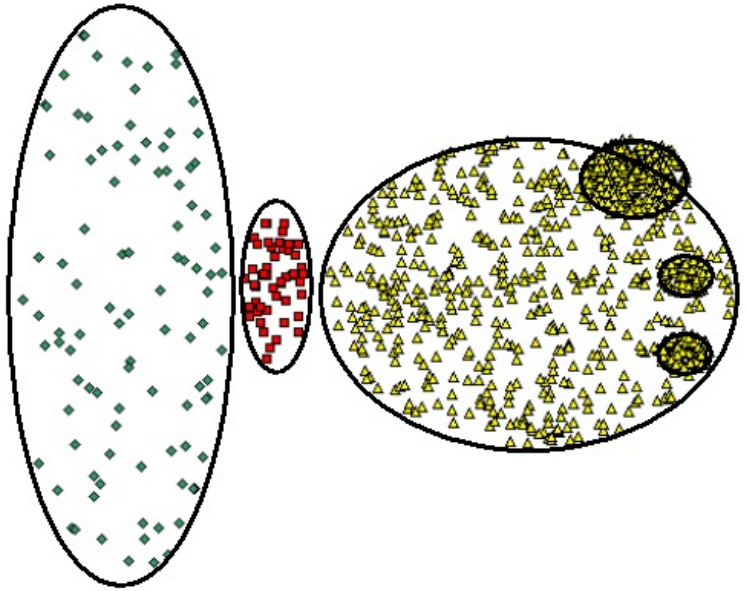
- Jarvis-Patrick Clustering for K=3 and T=2

	A	B	C	D	E	F
A	0	2	1	2	0	0
B		0	1	2	0	0
C			0	0	0	1
D				0	0	0
E					0	2
F						0

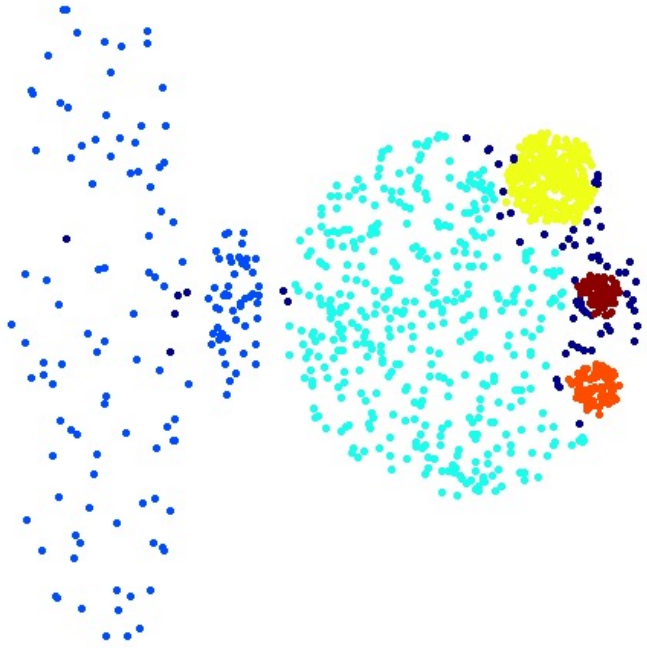
- Thus, three clusters are obtained: {A, B, D}, {C}, {E, F}



When Jarvis-Patrick Works Reasonably Well



Original Points



Jarvis Patrick Clustering

SNN Density-Based Clustering

1. Compute the similarity matrix
   This corresponds to a similarity graph with data points for nodes and edges whose weights are the similarities between data points

2. Sparsify the similarity matrix by keeping only the *k* most similar neighbors
   This corresponds to only keeping the *k* strongest links of the similarity graph

3. Construct the shared nearest neighbor graph from the sparsified similarity matrix.
   At this point, we could apply a similarity threshold and find the connected components to obtain the clusters (Jarvis-Patrick algorithm)

4. Find the SNN density of each Point.
   Using a user specified parameters, ***Eps***, find the number points that have an SNN similarity of *Eps* or greater to each point. This is the SNN density of the point

SNN Density-Based Clustering…

5. **Find the core points**
   Using a user specified parameter, *MinPts*, find the core points, i.e., all points that have an SNN density greater than *MinPts*

6. **Form clusters from the core points**
   If two core points are within a "radius", *Eps*, of each other they are placed in the same cluster

7. **Discard all noise points**
   All non-core points that are not within a "radius" of *Eps* of a core point are discarded

8. **Assign all non-noise, non-core points to clusters**
   This can be done by assigning such points to the nearest core point

   (Note that steps 4-8 are DBSCAN)

EM Algorithm for GMM Summary

Given the data set: $x_1, x_2, \ldots, x_n$, and number of cluster K

Initialize π_k $\mu_k \Sigma_k$ randomly ($k = 1, 2, \ldots K$)

Loop (until convergence)

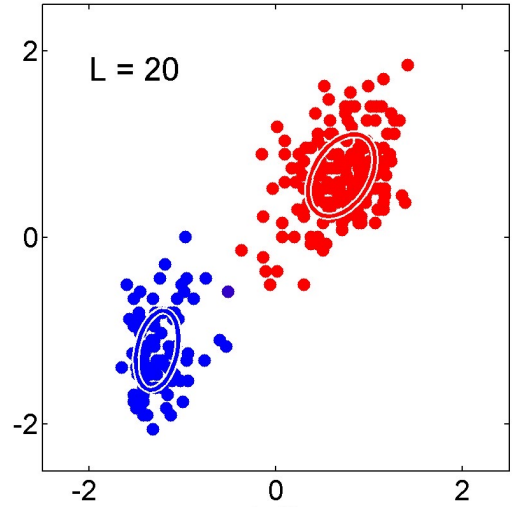
E-step: computer Expectation:

$$r_{ik} = \frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}$$

M-step: petameter update:

$$\pi_k = \frac{\sum_i r_{ik}}{n} \quad \mu_k = \frac{\sum_i r_{ik} x_i}{\sum_i r_{ik}} \quad \Sigma_k = \frac{\sum_i r_{ik}(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i r_{ik}}$$

End

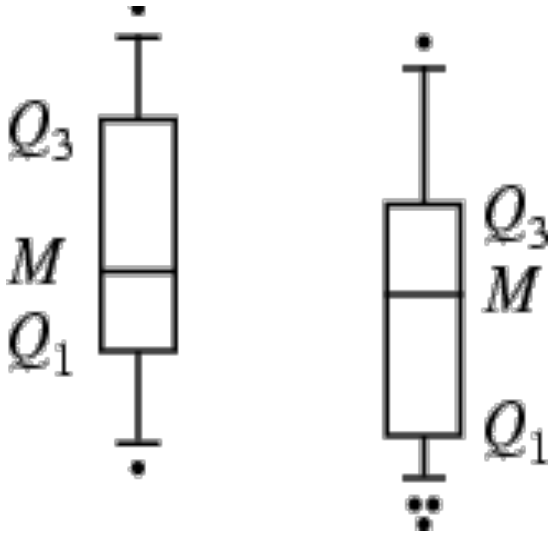


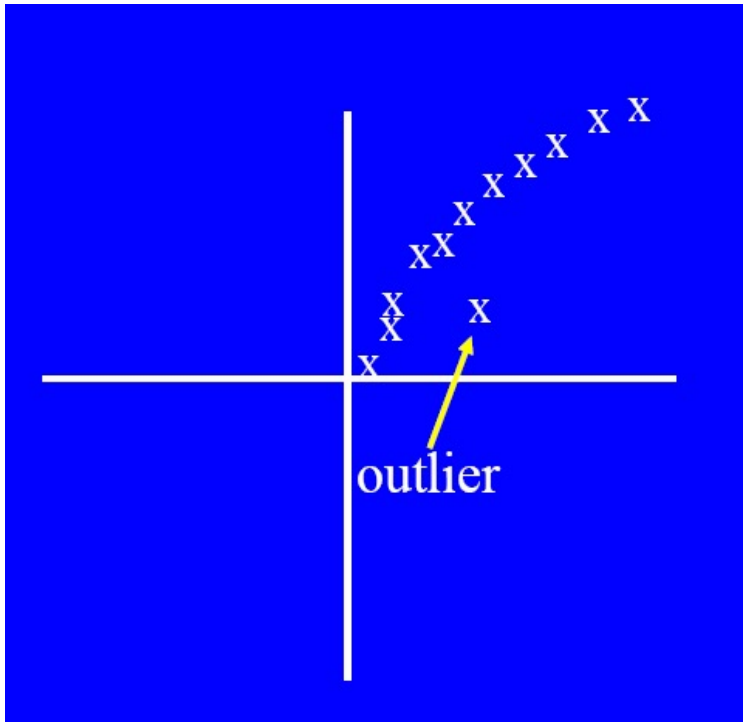
https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html

Anomaly Detection

Visual Approaches

- Boxplots or scatter plots
- Limitations
  - Not automatic
  - Subjective





Proximity/Distance-based methods

Approach 1:

- The outlier score of an object is the distance to its k'th nearest neighbor
  - You need to specify k

Approach 2:

- The average distance to k neighbors
- More robust to the choice of k

Density-Based Approaches

- Density definitions:
  - One definition: Inverse of distance to k'th neighbor:

$$density(x, k) = \frac{1}{dist(x, k)}$$

  - Another definition: Inverse of the average distance to k neighbors:

$$density(x, k) = \frac{1}{avg.dist(x, k)}$$

  - DBSCAN definition

Still, if there are regions of different density, this approach can have problems

Relative Density

- Consider the density of a point relative to that of its k nearest neighbors and consider that as the anomaly score.

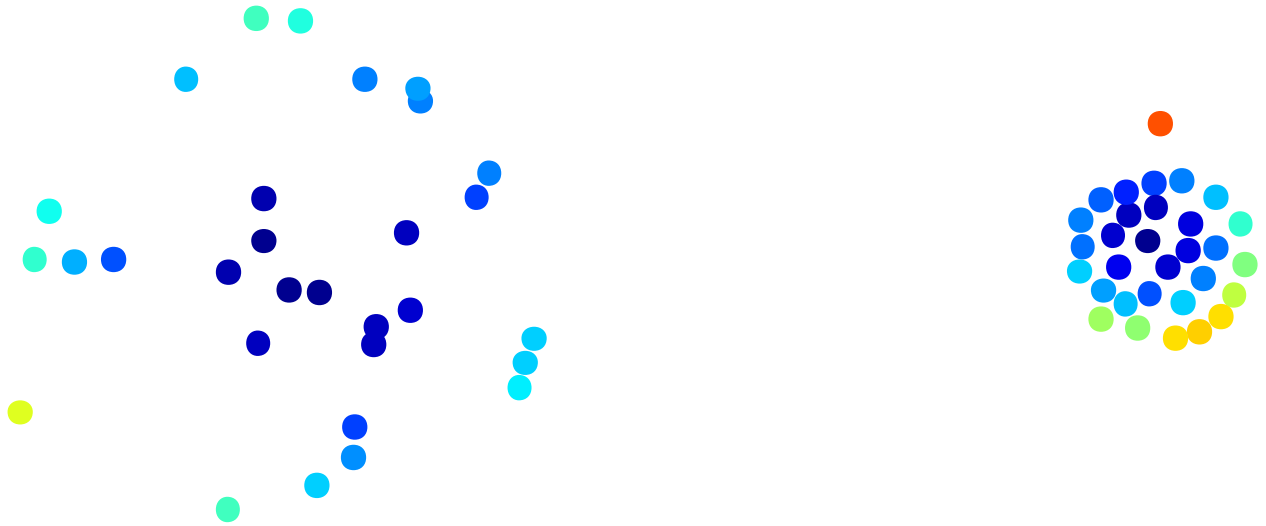
$$Relative\ Density\ (x,k) = \frac{\sum_{y \in N(x,k)} density(y,k)/k}{density(x,k)}$$

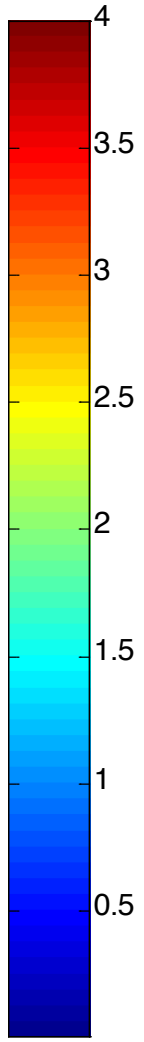
This approach can handle regions with different density

Relative Distance of Points from Closest Centroid

Anomaly score of an object is *Relative Distance*

$$Relative\ Distance(x, cluster_i) = \frac{dist(x, c_i)}{Median\{dist(y, c_i): y\ \in cluster_i\}}$$





Outlier Score

This approach can handle regions with different density

Association Rule Mining

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

Market-Basket transactions

TID	*Items*
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 Association Rules

{Diaper} → {Beer},
{Milk, Bread} → {Coke},
{Bread} → {Milk},

Definition: Association Rule

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Association Rule

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

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:

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

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

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

Mining Association Rules

- Two-step approach:
  - Frequent Itemset Generation
    - Generate all itemsets whose support ≥ minsup
  - 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!

Reducing Number of Candidates

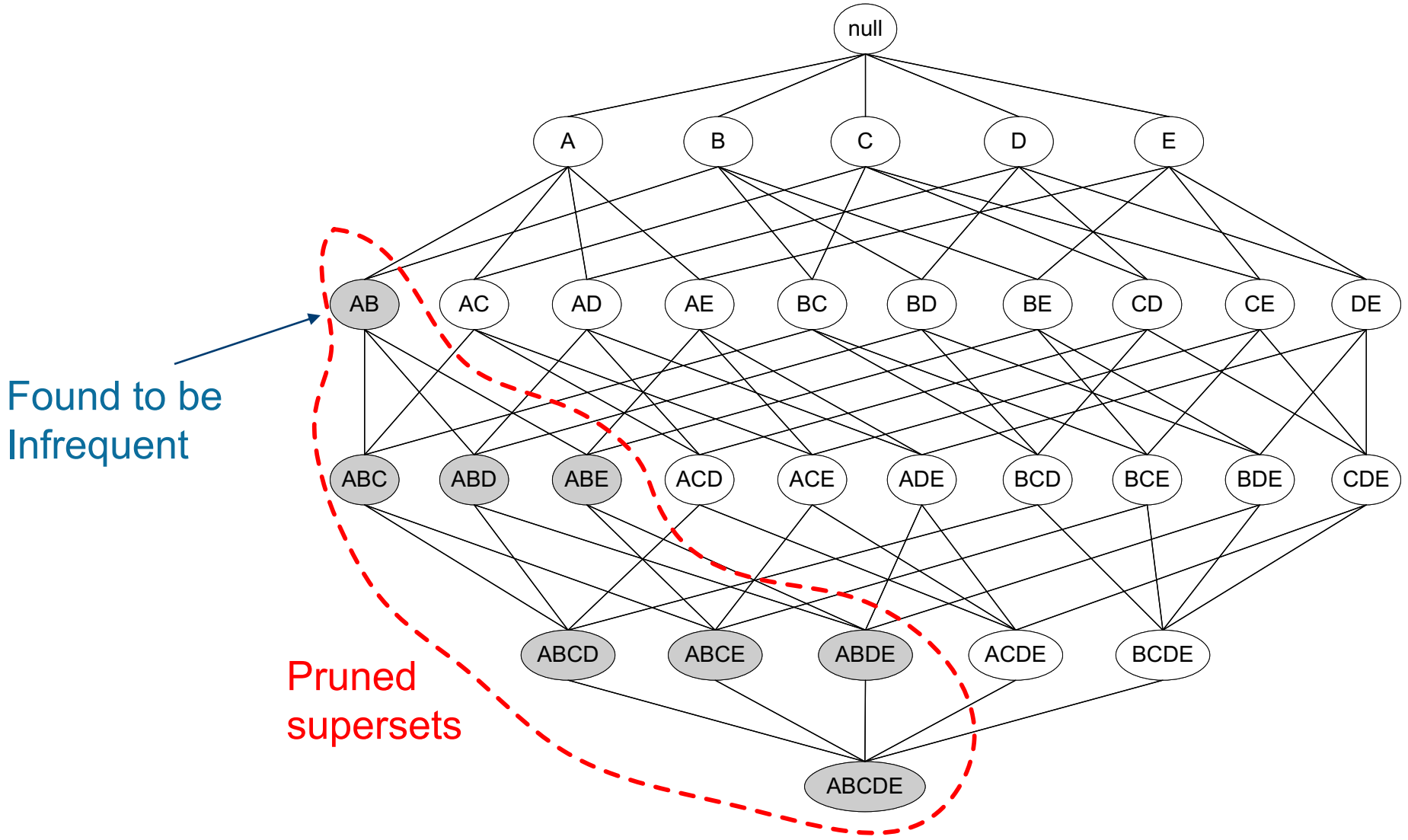
- Apriori Principle:
  - If an itemset is frequent, then all of its subsets must also be frequent

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

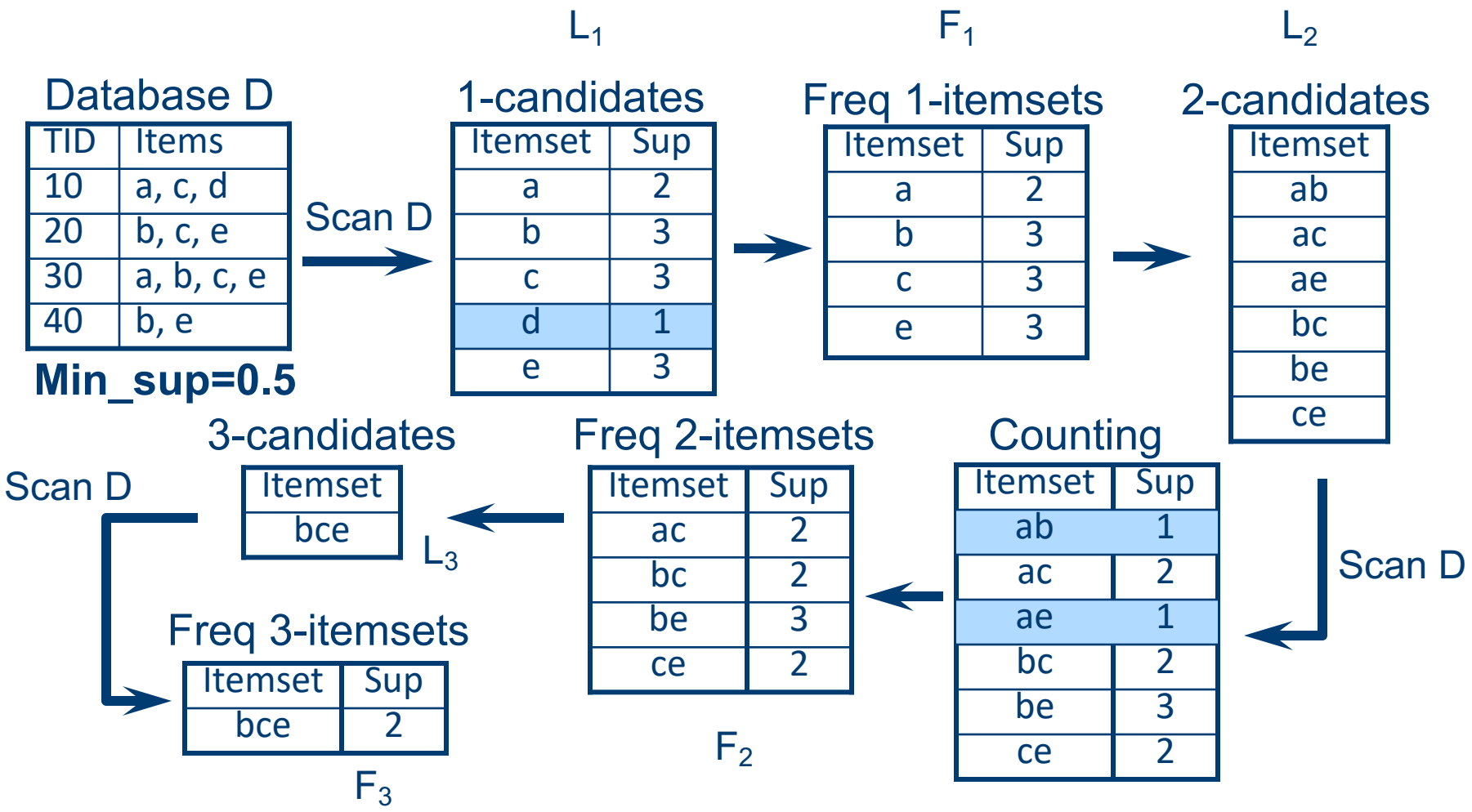
$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

  - Support of an itemset never exceeds the support of its subsets
  - This is known as the anti-monotone property of support

Illustrating Apriori Principle



Example: Apriori-based Mining

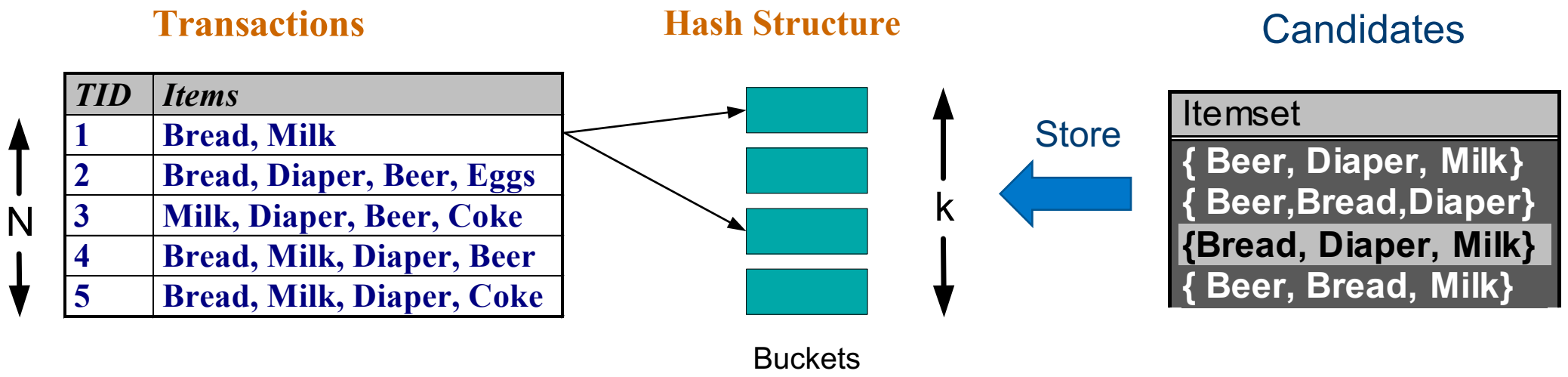


Know you know how to create L_3 from F_2 😄

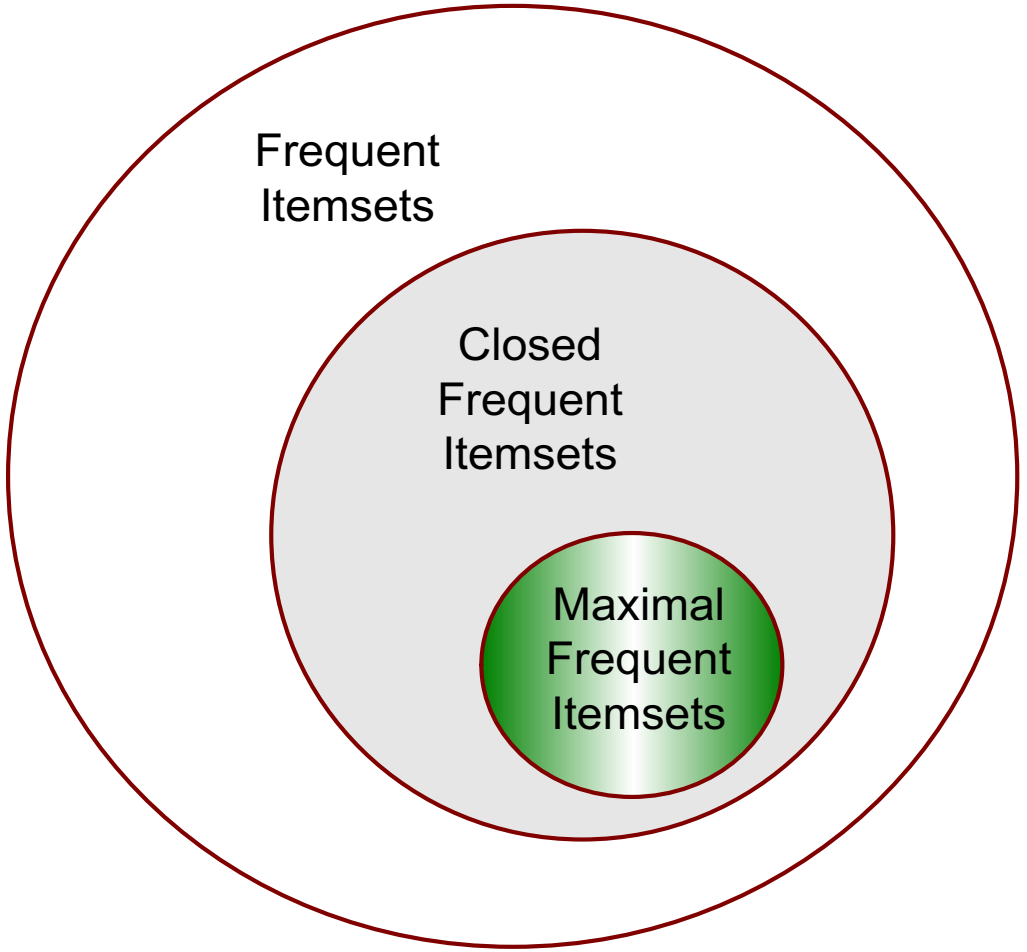
Support Counting of Candidate Itemsets

To reduce number of comparisons, store the candidate itemsets in a hash structure

- Instead of matching each transaction against every candidate, match it against candidates contained in the hashed buckets



Maximal vs Closed Itemsets



FP-growth Algorithm

- To discover frequent itemsets by a compressed representation of the database using an FP-tree (Frequent Pattern Tree)
- Once an FP-tree has been constructed, it uses a recursive divide-and-conquer approach to mine the frequent itemsets
- It does not employ the generate-and-test paradigm of the Apriori algorithm.

FP-growth Ideas

- Grow long patterns from short ones using local frequent items
  - “abc” is a frequent pattern
  - Get all transactions having “abc”:

    DB|abc (projected database on abc)

  - “d” is a local frequent item in DB|abc → abcd is a frequent pattern
  - Get all transactions having “abcd” (projected database on “abcd”) and find longer itemsets

Thanks!