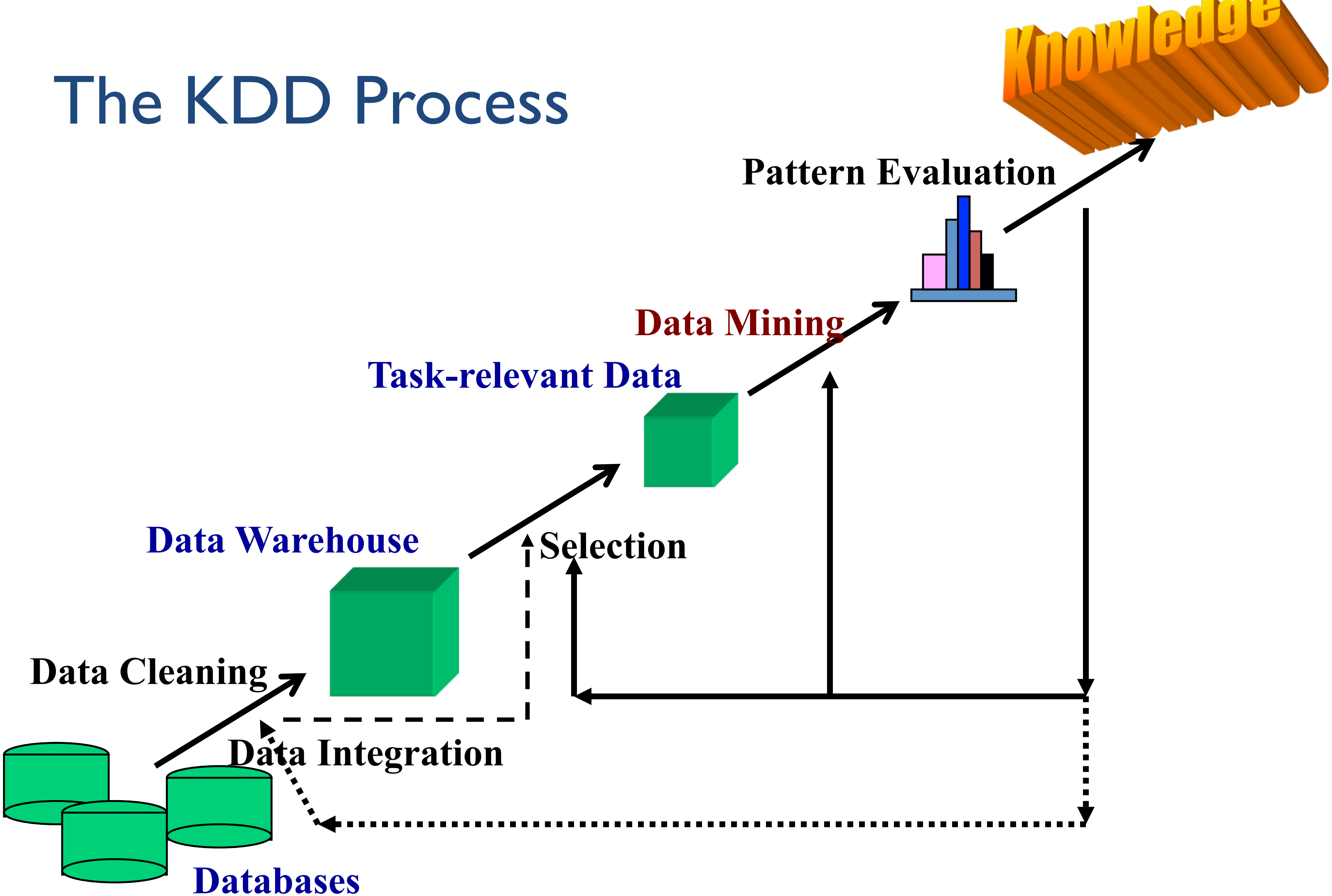
Data Mining Techniques

Anna Monreale
Computer Science Department

The KDD Process



Supervised vs. Unsupervised Learning

- **Supervised learning (classification)**
  - Supervision: The training data (observations, measurements, etc.) are accompanied by **labels** indicating the class of the observations
  - **New data is classified based on the training set**
- **Unsupervised learning (clustering)**
  - The class labels of training data is unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

CLUSTERING

Clustering Definition

- **Cluster**: A collection of data objects
- 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.

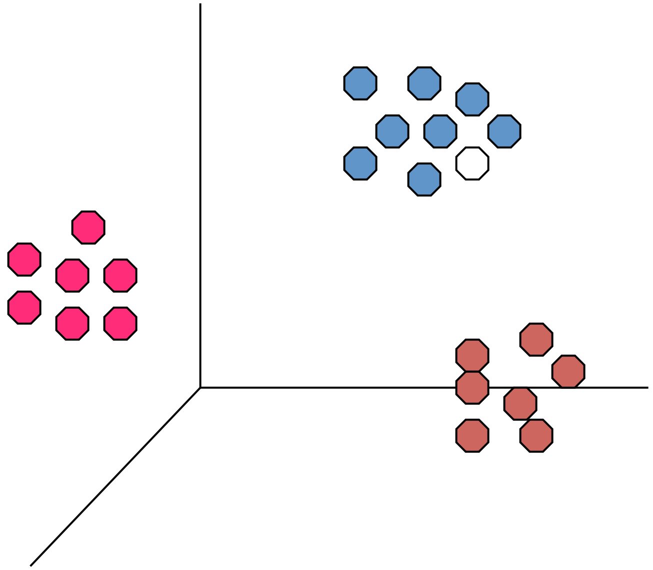
- **Similarity Measures?**
  - Euclidean Distance if attributes are continuous.
  - Other Problem-specific Measures.

Illustrating Clustering

Euclidean Distance Based Clustering in 3-D space

Intracluster distances are minimized

Intercluster distances are maximized



Different clustering approaches

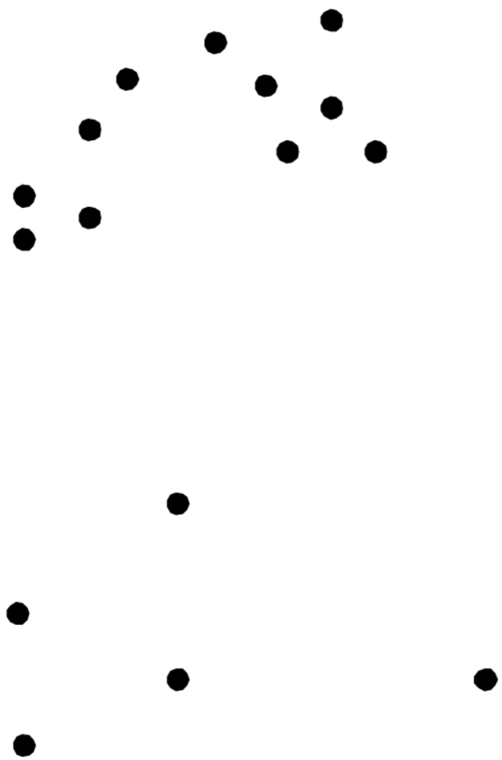
PARTITIONING ALGORITHMS

Directly divides data points into some prespecified number of clusters without a hierarchical structure

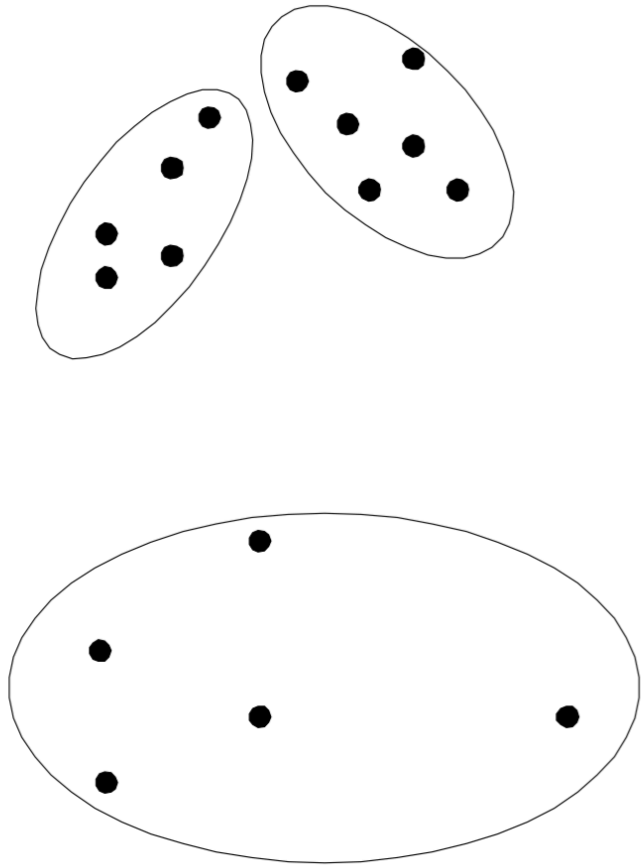
HIERARCHICAL ALGORITHMS

Groups data with a sequence of nested partitions, either from singleton clusters to a cluster containing all elements, or viceversa

PARTITIONING Clustering



Original Points



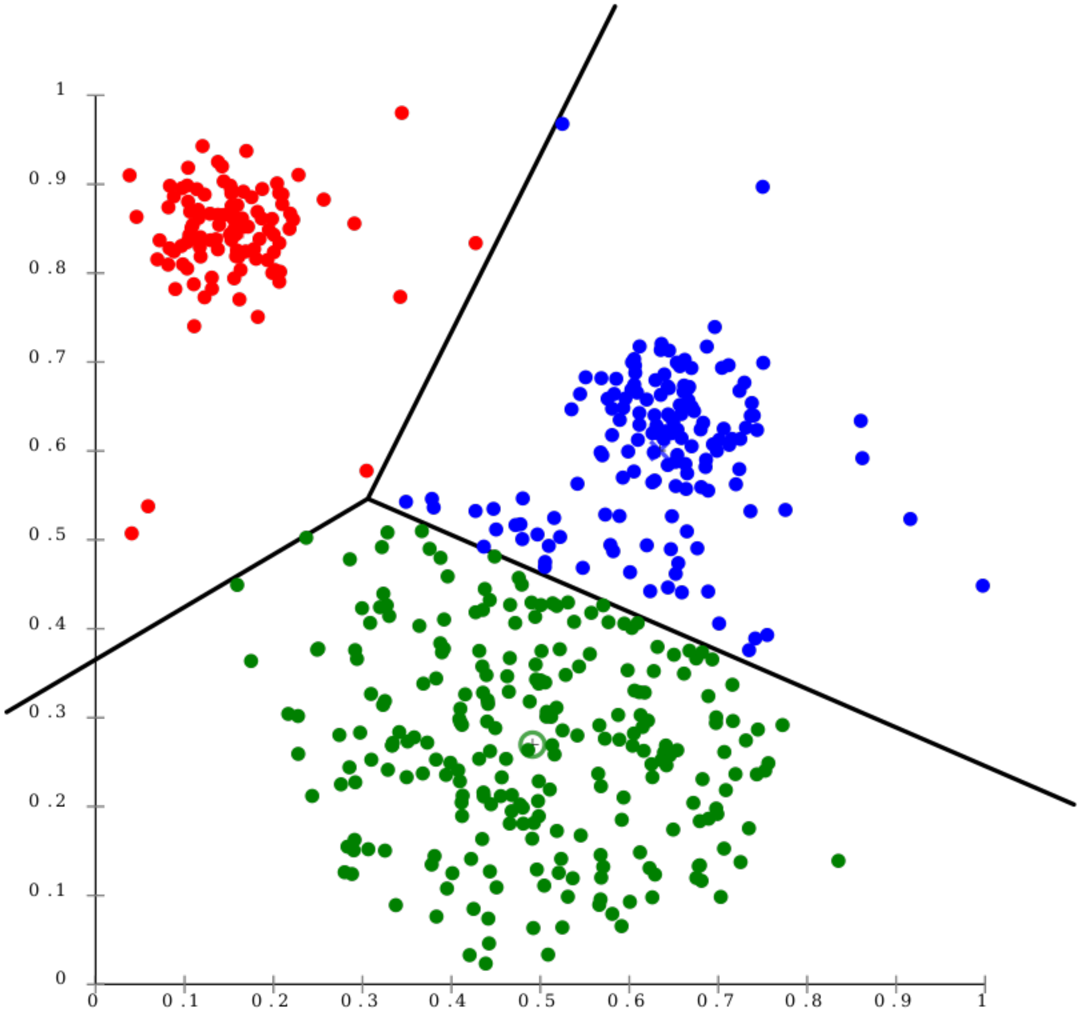
A Partitional Clustering

Center-based clustering

A cluster is a set of objects such that an object in a cluster is **closer (more similar) to the "center"** of a cluster, than to the center of any other cluster

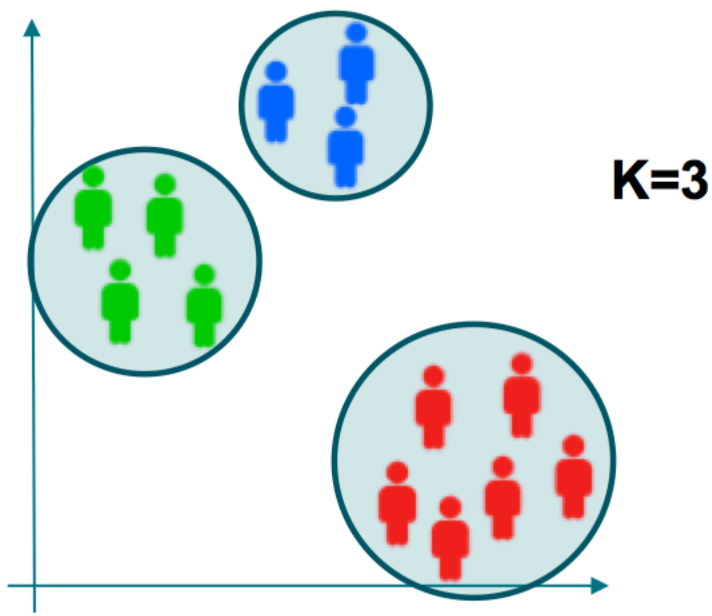
The center of a cluster is often a **centroid**, the average of all the points in the cluster, or a **medoid**, the most "representative" point of a cluster

K-means or k-medoid



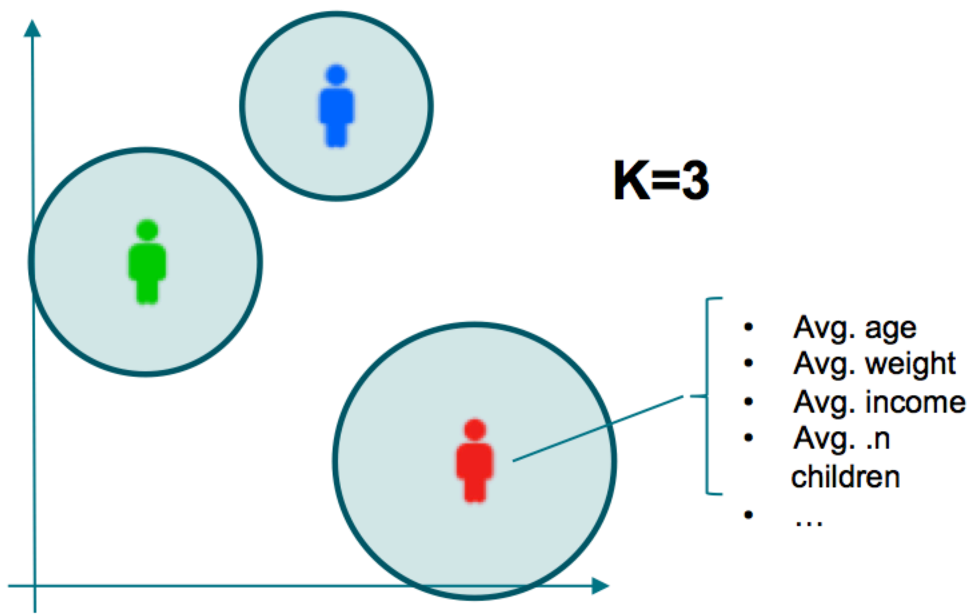
Clustering: K-means (family)

- Output 1: a partitioning of the initial set of objects

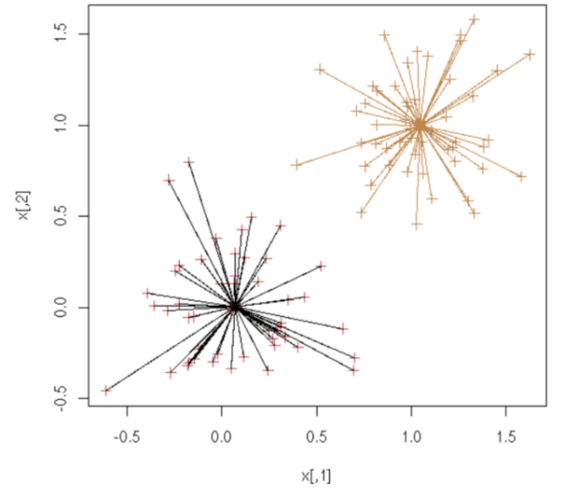


Clustering: K-means (family)

- Output 2: K representative objects (centroids)
- Centroid = average profile of the objects in the cluster



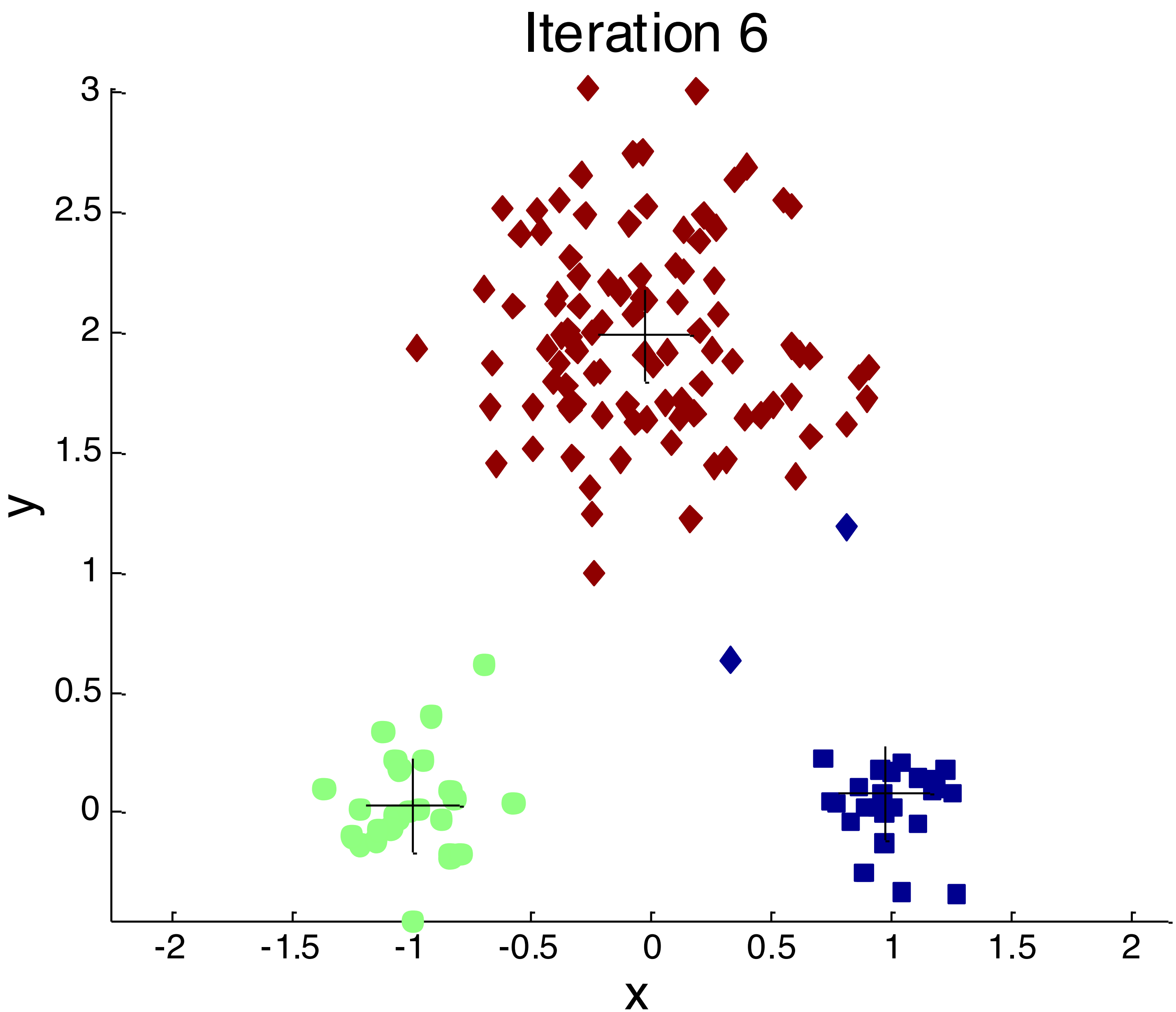
K-means Clustering



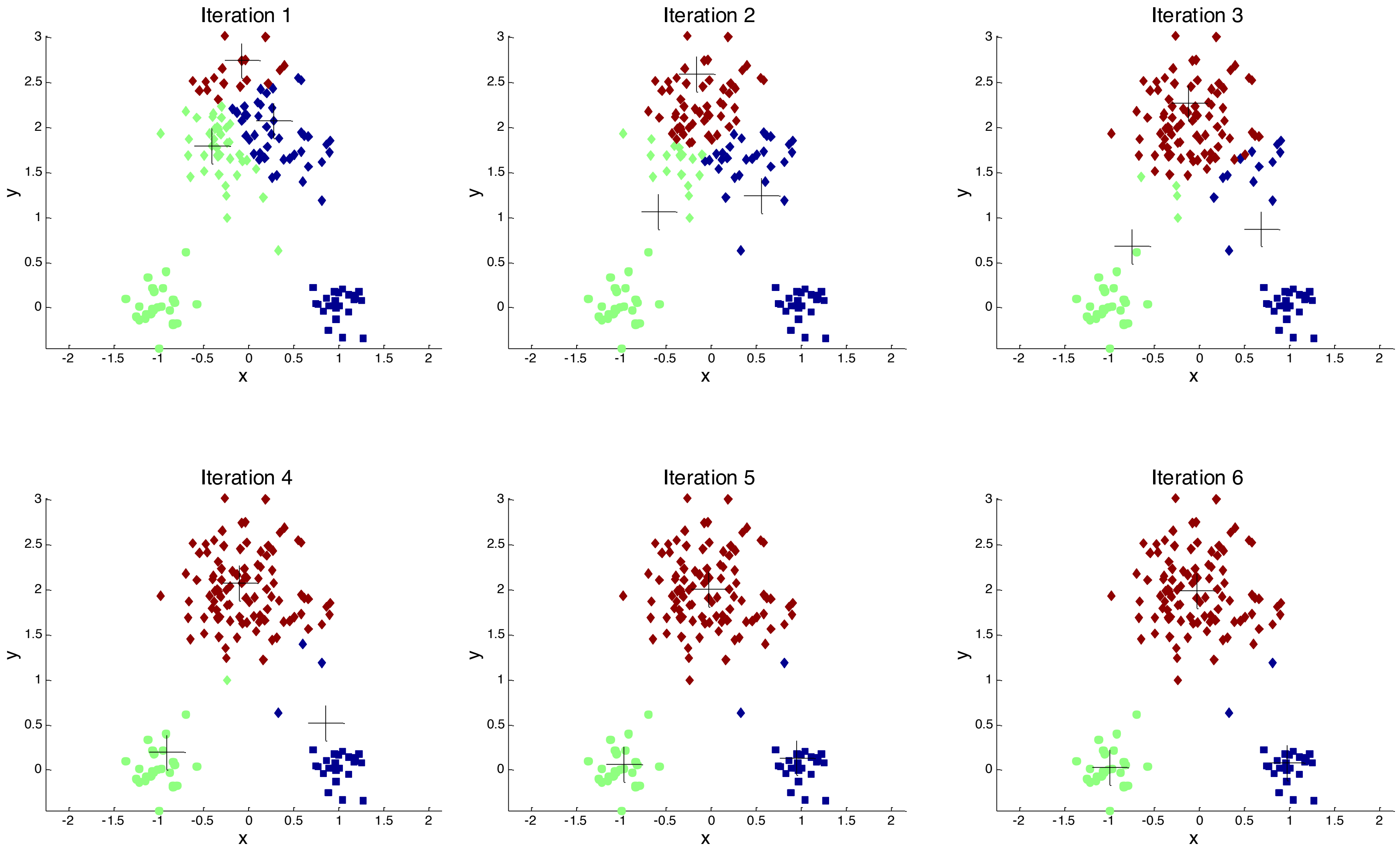
- Partitional clustering approach
- Number of clusters, K, must be specified
- Each cluster is associated with a **centroid**
- Each point is assigned to the cluster with the **closest centroid**
- The basic algorithm is very simple

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

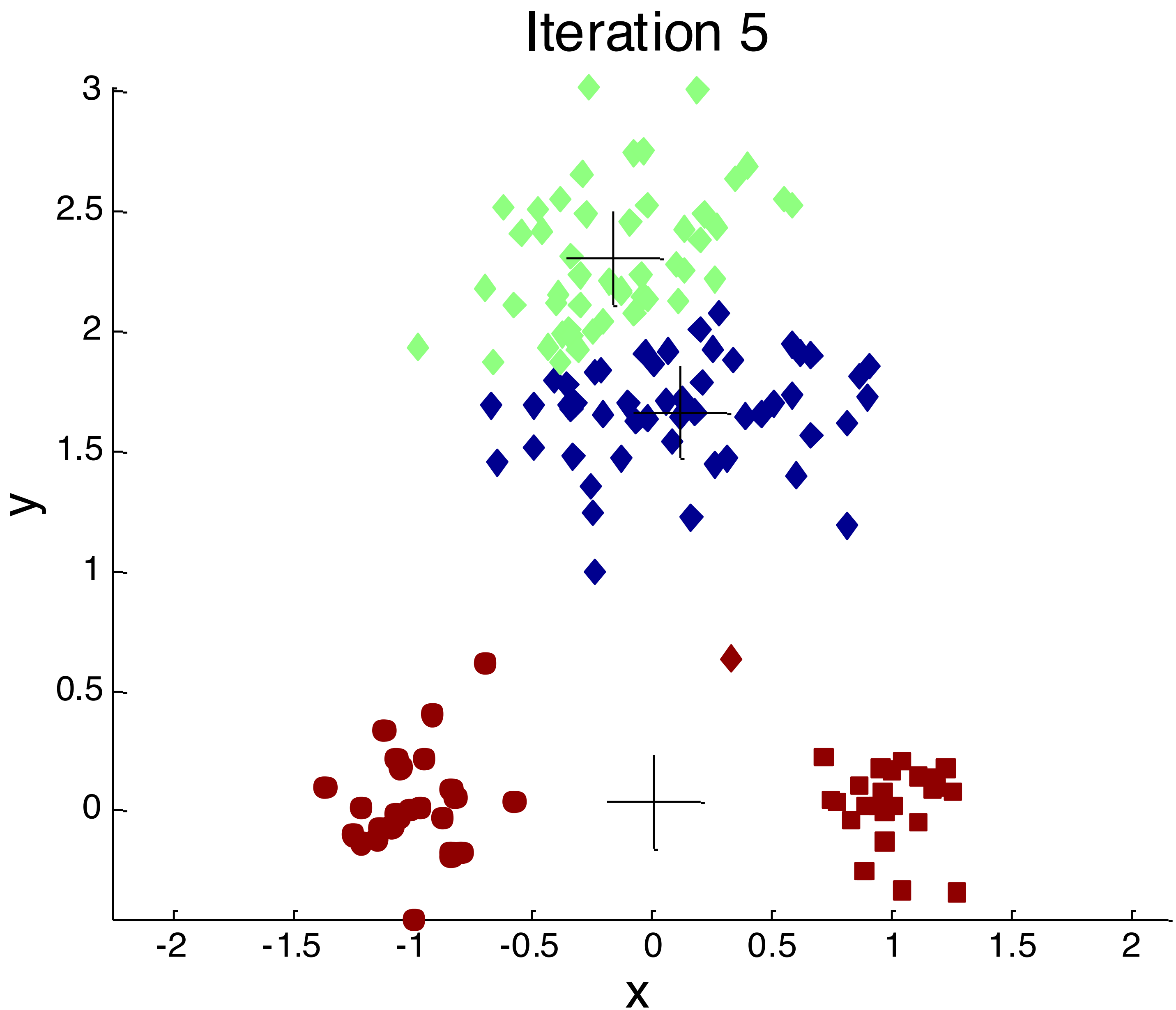
Importance of Choosing Initial Centroids



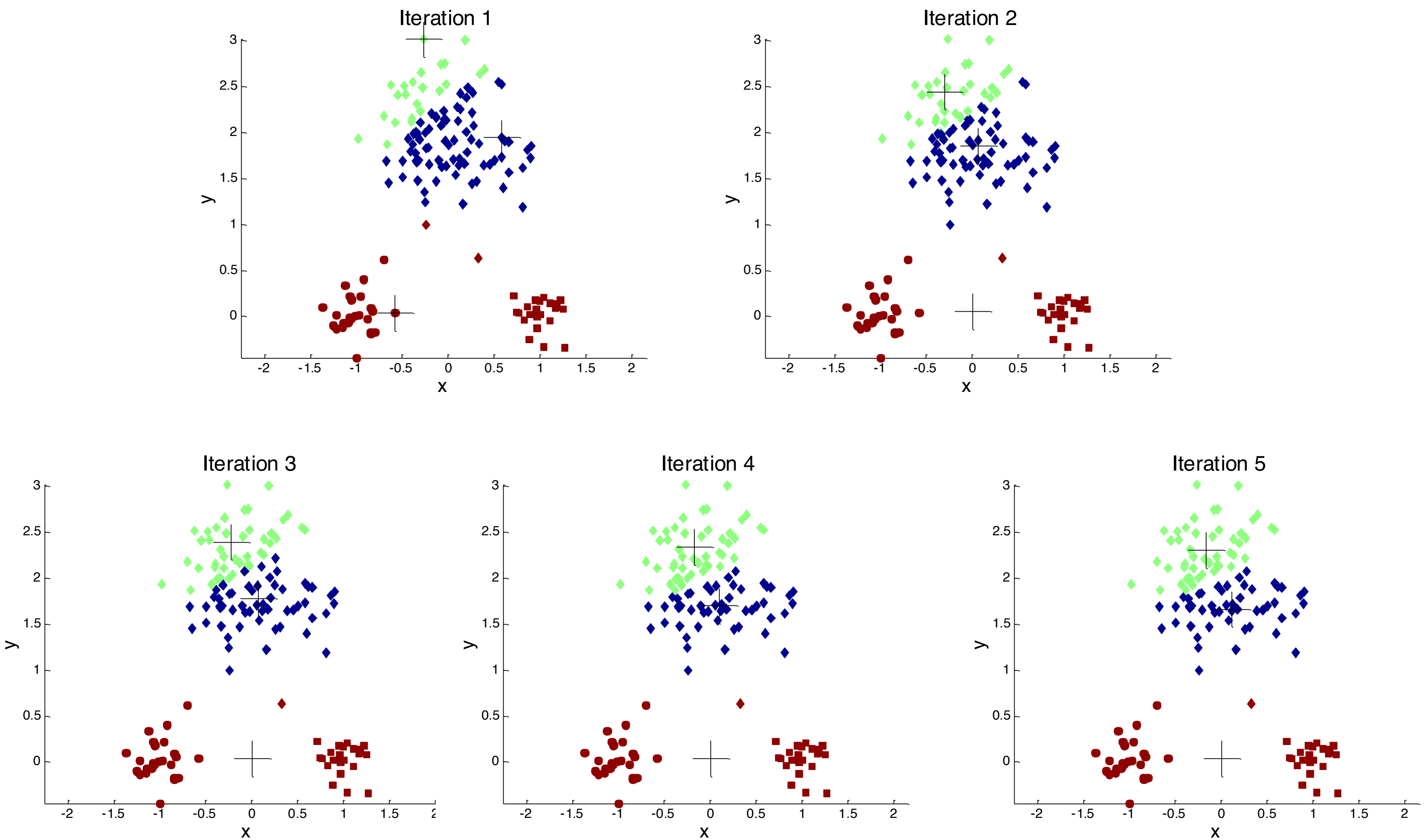
Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids …



Importance of Choosing Initial Centroids …



Pre-processing and Post-processing

- **Pre-processing**
  - Normalize the data
  - Eliminate outliers
- **Post-processing**
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are ‘close’ and that have relatively low SSE

Market Segmentation

Goal: subdivide a market into distinct **subsets of customers** where any subset may conceivably be selected as a market target to be reached with a **distinct marketing mix.**

Approach

1. **Collect different attributes** of customers based on their geographical, **Demographic,** lifestyle, **Behavioral** related information
2. **Find clusters** of similar customers
3. Measure **the clustering quality** by observing buying patterns of customers in same cluster vs. those from different clusters.

A Behavior Based Segmentation Example

Using unsupervised clustering segmentation for a grocery chain which would like better product assortment for its high profitable customers

Potential Inputs

Value	• Basket Size • Visit Frequency
Basket	• Spend by category • Type of category • Brand spend (i.e. private label)
Promotions	• % bought on targeted promotion • % bought from flyer
Time	• Time of day • Day of week
Location	• Store format • Area population density

Clustering approach

Deal Seeking Mom

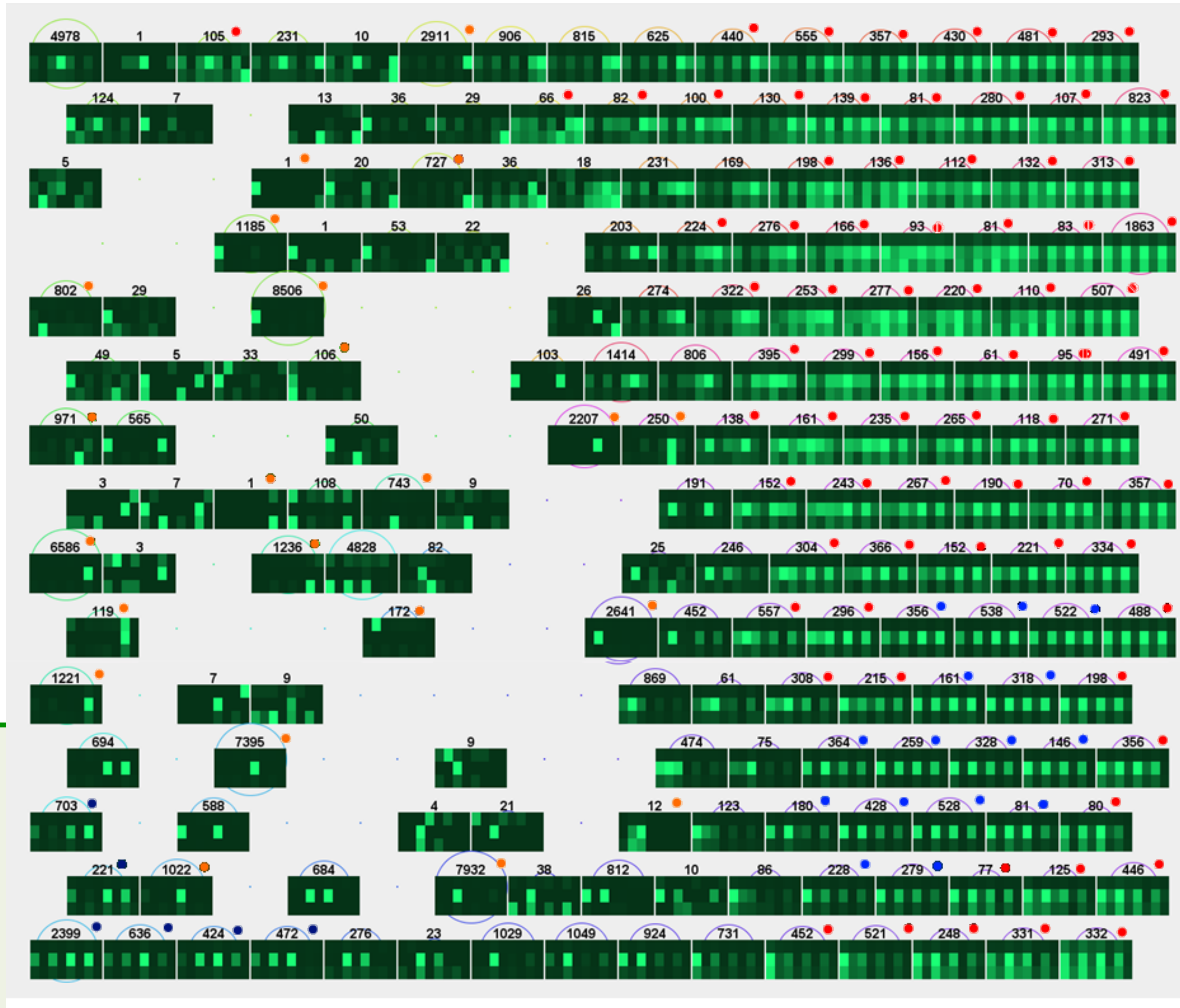
Key Differentiators

- Full store shop
- High avg. basket size / # trips

- High % purchased on promotion
- Rewards seeker

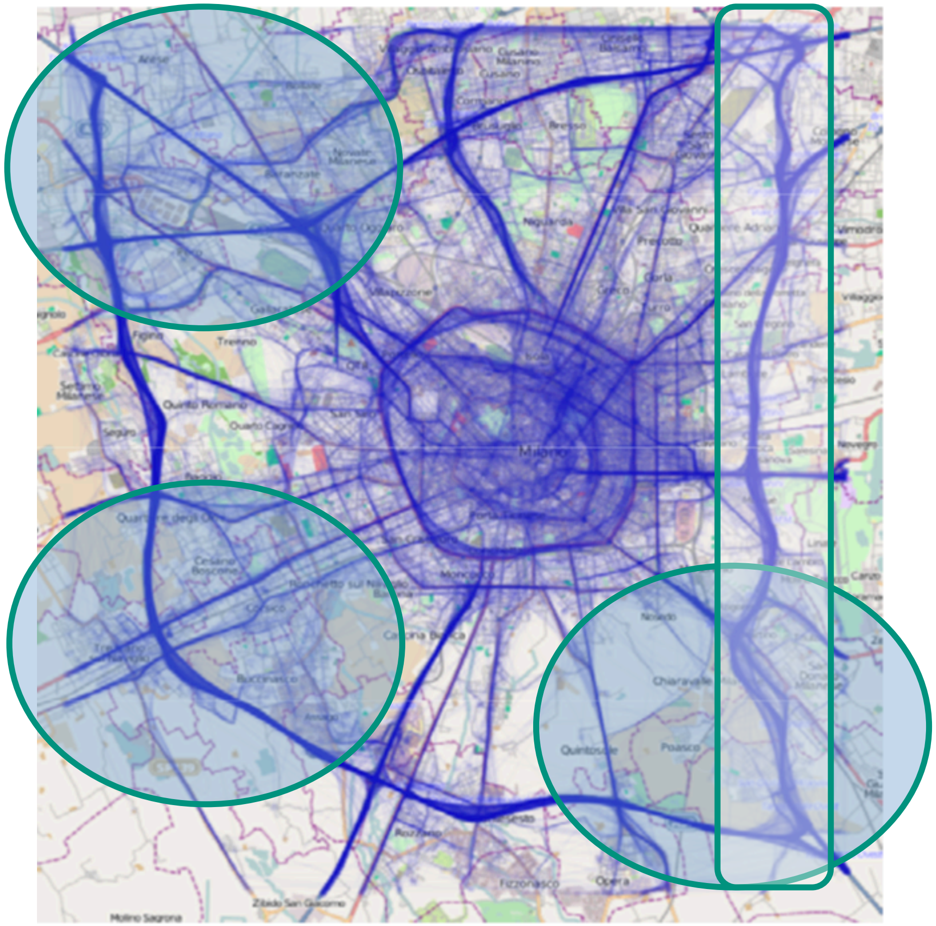
- High spend categories
  - Fresh produce
  - Organic food
  - Multipack juice, snack

CDR Profiling

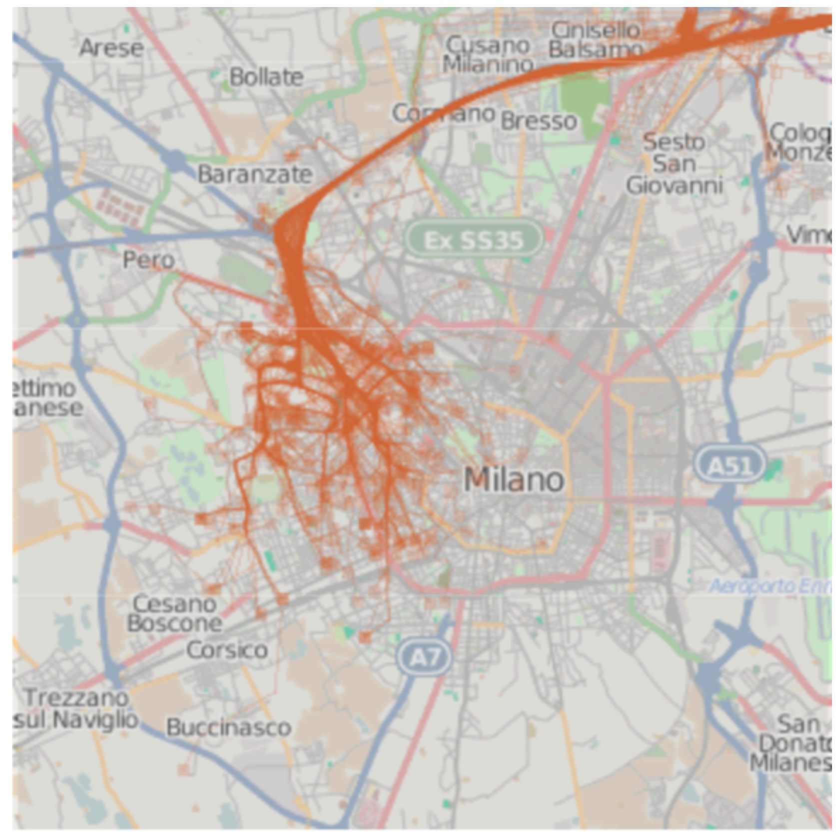


call
ited by
rator

A particular Clustering Application

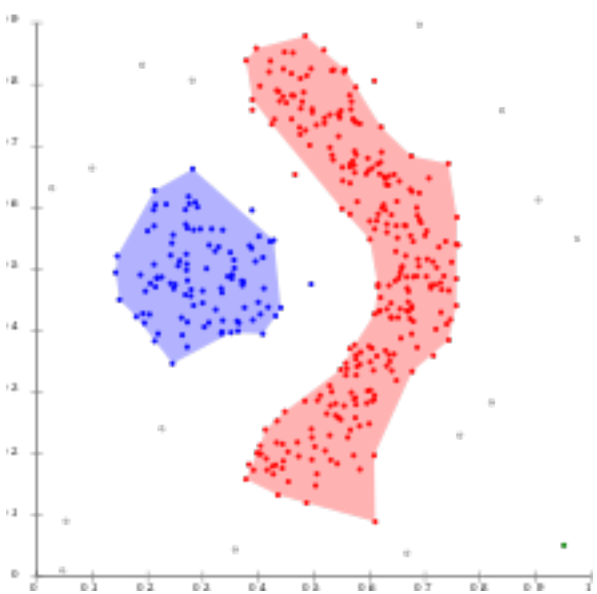


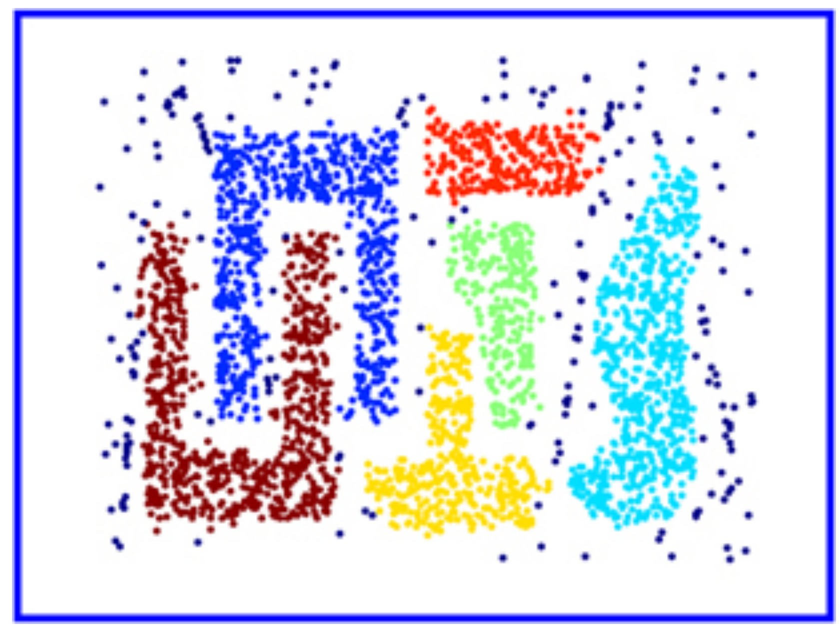
NO GLOBULAR CLUSTERS



Density-based clustering

Clusters are **dense regions** in the data space separated by regions with **lower density**





DBSCAN

- A density-based algorithm.
  - Density = number of points within a specified radius (Eps)

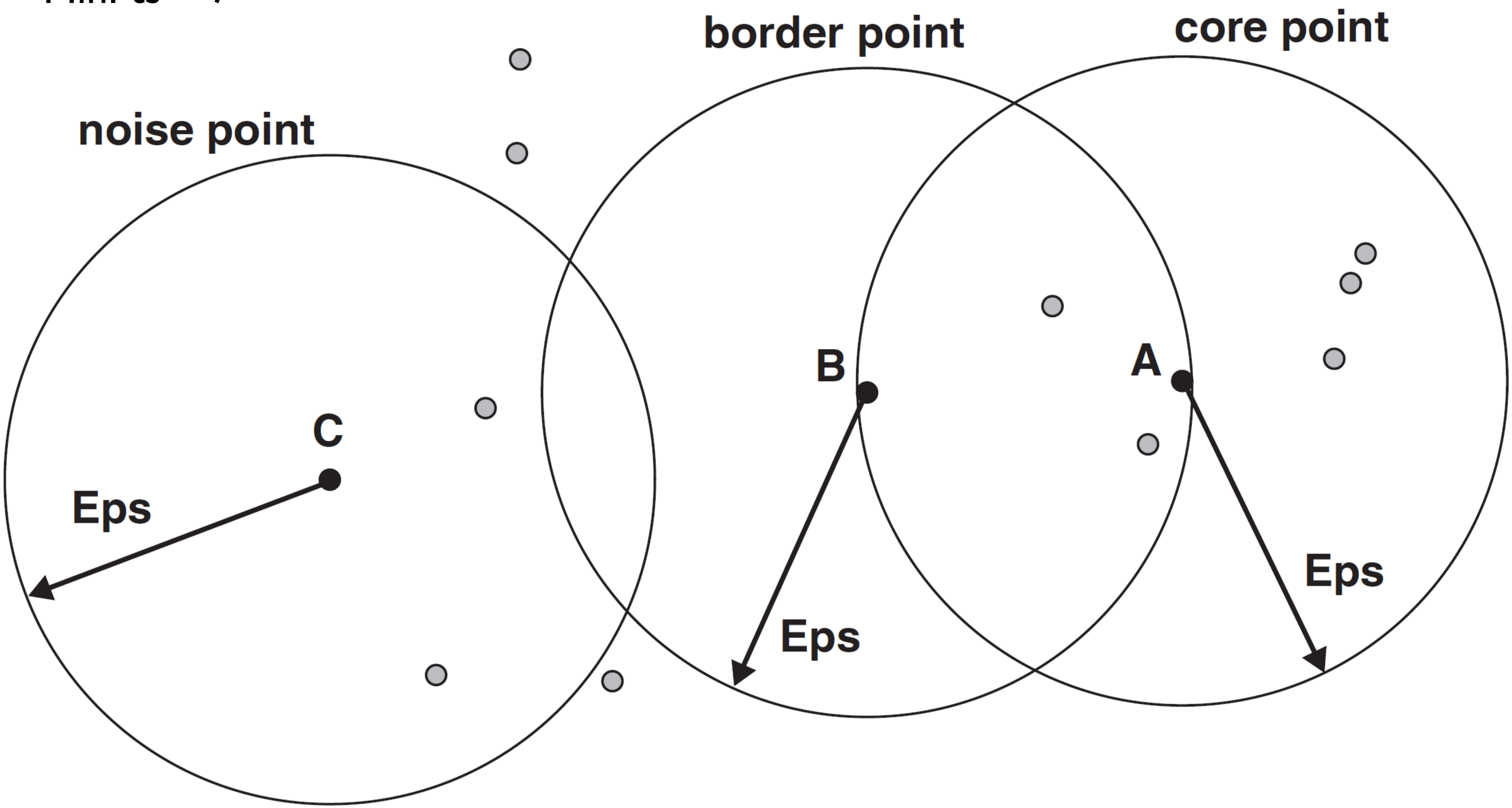
- A point is a core point if it has at least a specified number of points (MinPts) within Eps
  - These are points that are at the interior of a cluster
  - Counts the point itself

- 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

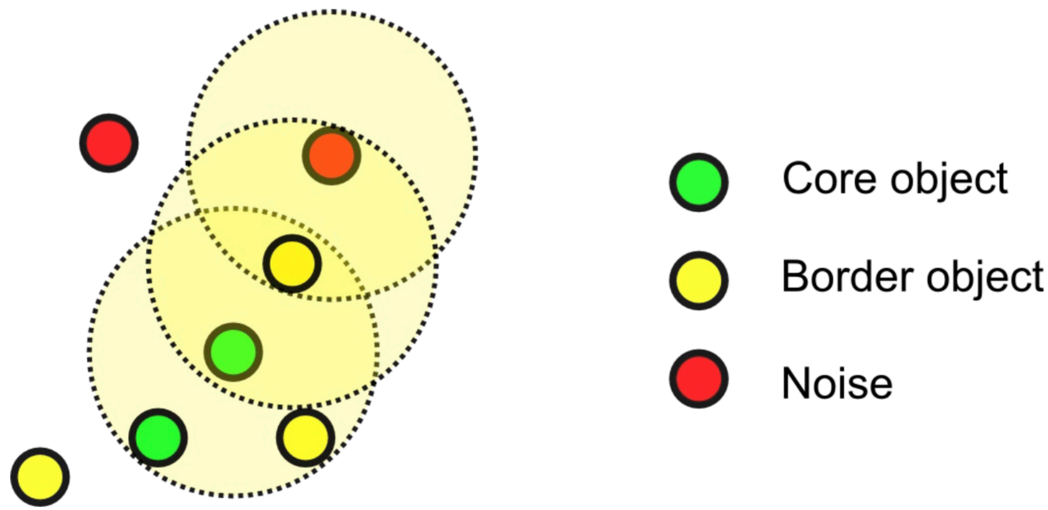
DBSCAN: Core, Border, and Noise Points

MinPts = 7



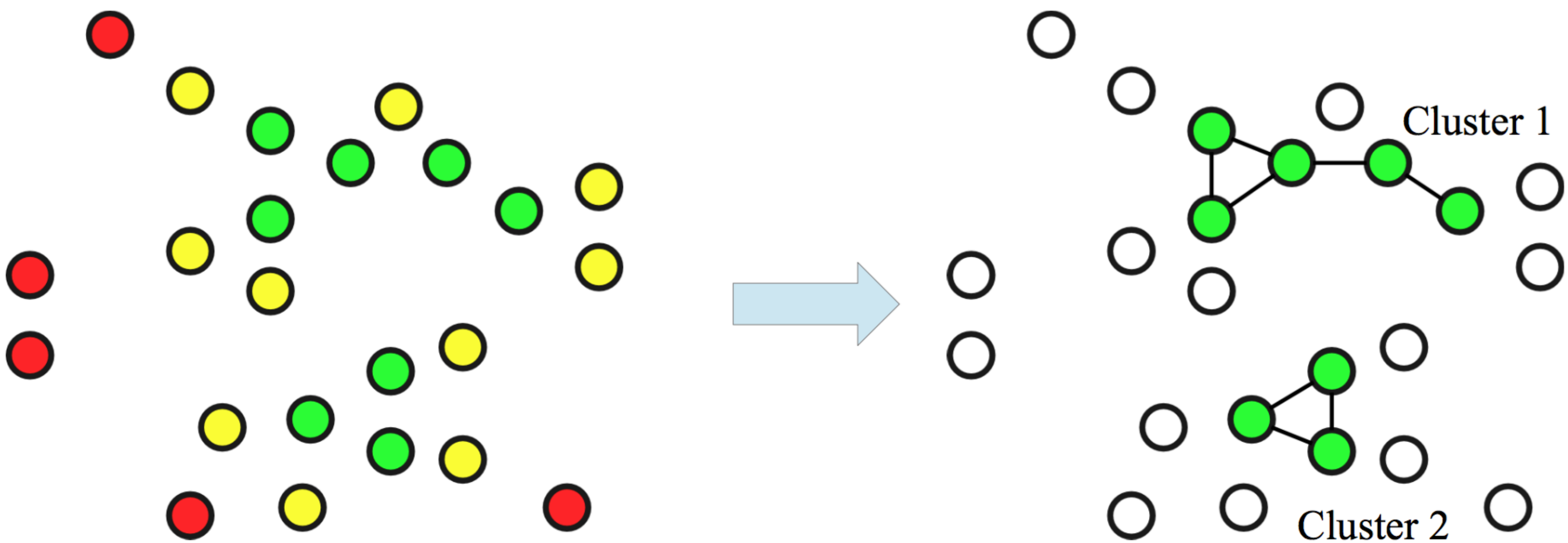
DBSCAN: Step 1

- Label points as **core** (dense), **border** and **noise**
  - Based on thresholds R (radius of neighborhood) and min_pts (min number of neighbors)



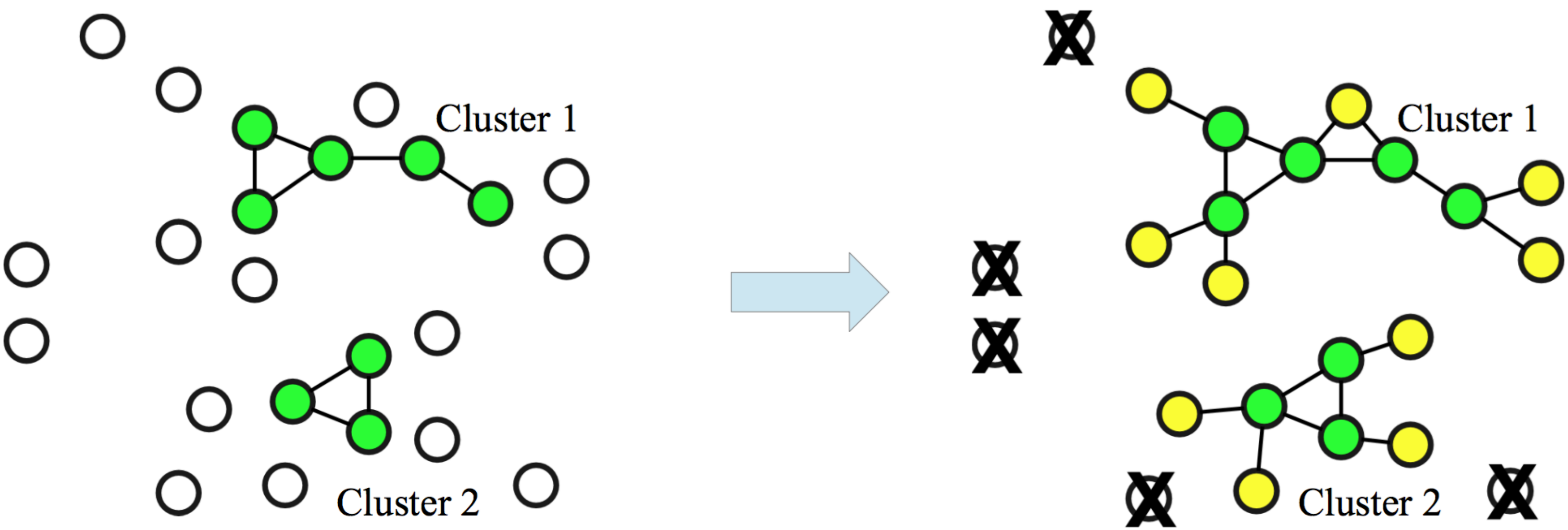
DBSCAN: Step 2

- Connect core objects that are neighbors, and put them in the same cluster



DBSCAN: Step 3

- Associate border objects to (one of) their core(s), and remove noise



CLASSIFICATION

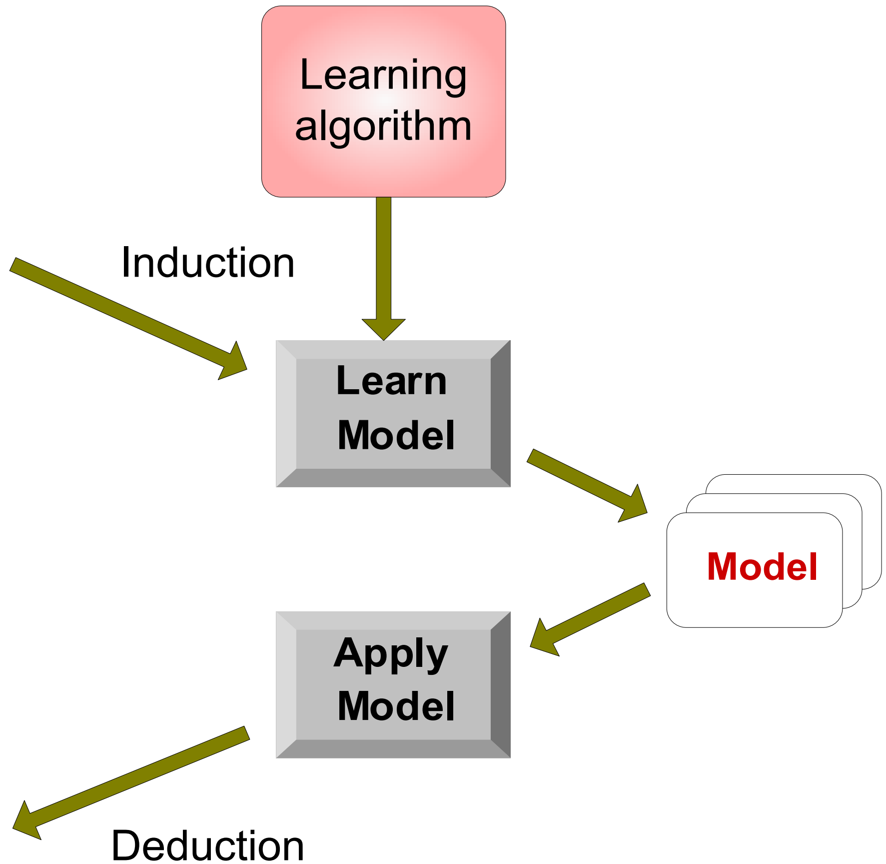
Classification: Definition

- Given a collection of records (*training set*)
  - Each record contains a set of *attributes*, one of the attributes is the *class*.
- Find a *model* for class attribute as a function of the values of other attributes.
- **Goal**: <u>previously unseen</u> records should be assigned a class as accurately as possible.
  - A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

General Approach for Building Classification Model

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Examples of Classification Task

Task	Attribute set, x	Class label, y
Categorizing email messages	Features extracted from email message header and content	spam or non-spam
Identifying tumor cells	Features extracted from MRI scans	malignant or benign cells
Cataloging galaxies	Features extracted from telescope images	Elliptical, spiral, or irregular-shaped galaxies

Classification Techniques

- Base Classifiers
  - Decision Tree based Methods
  - Rule-based Methods
  - Nearest-neighbor
  - Neural Networks
  - Deep Learning
  - Naïve Bayes and Bayesian Belief Networks
  - Support Vector Machines
- Ensemble Classifiers
  - Boosting, Bagging, Random Forests

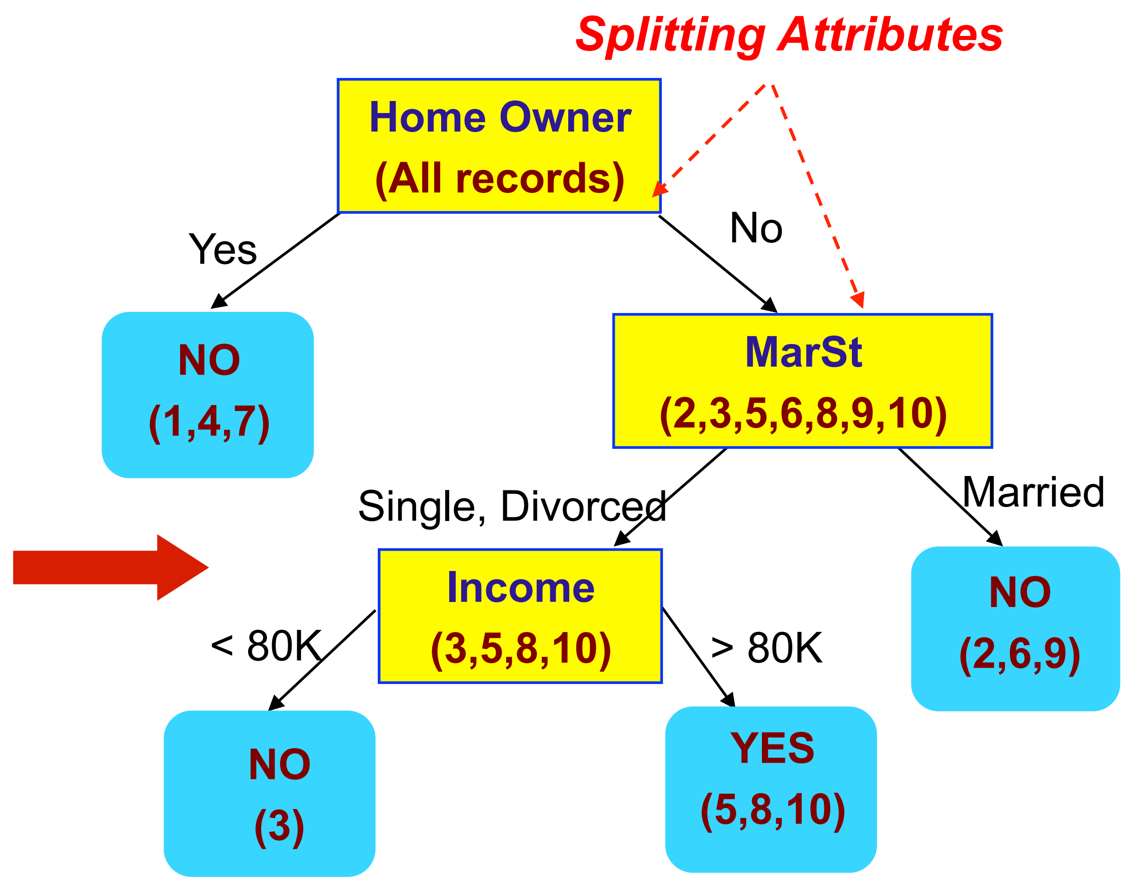
Example of a Decision Tree

Consider the problem of predicting whether a loan borrower will repay the loan or default on the loan payments.

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

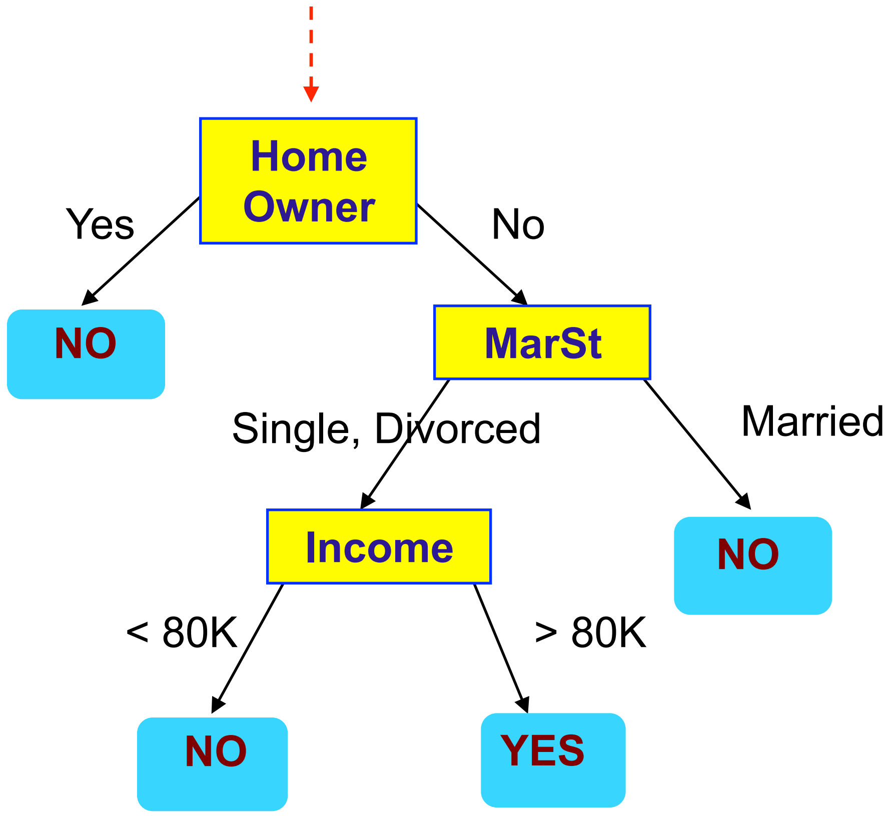
Training Data



Model: Decision Tree

Apply Model to Test Data

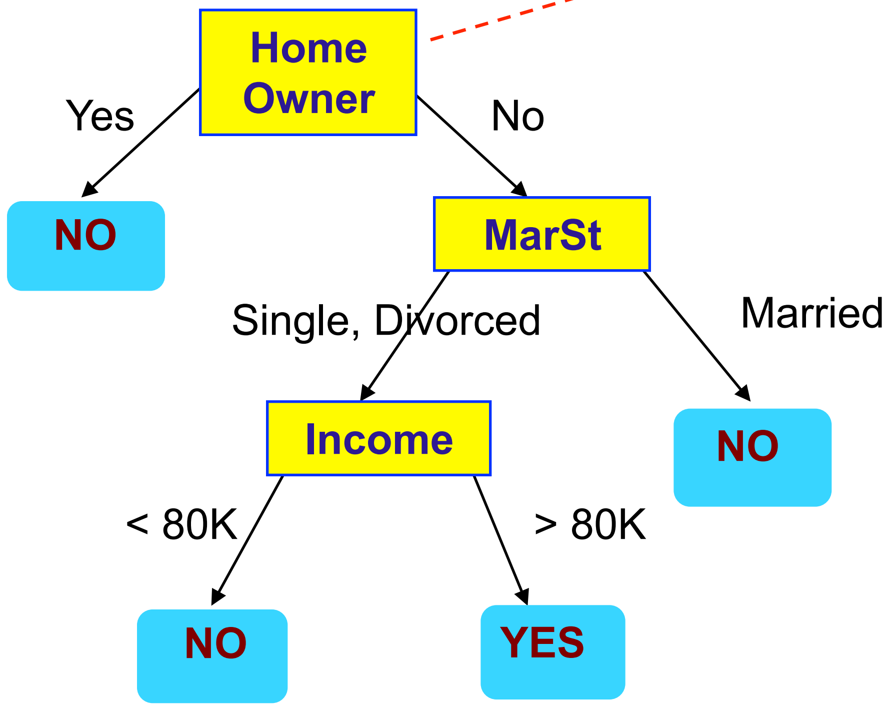
Start from the root of tree.



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

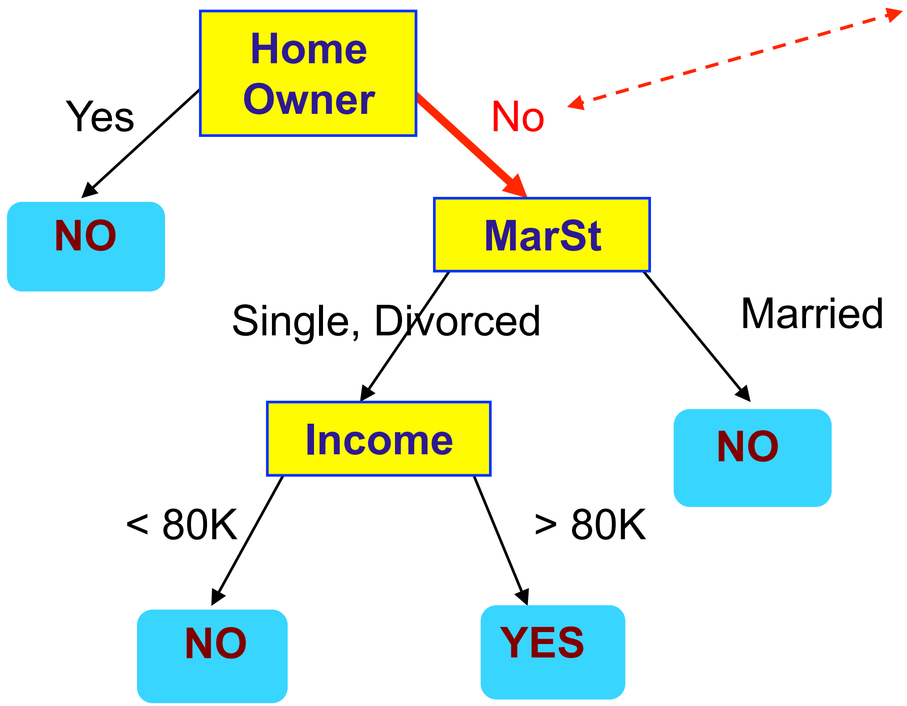
Apply Model to Test Data



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

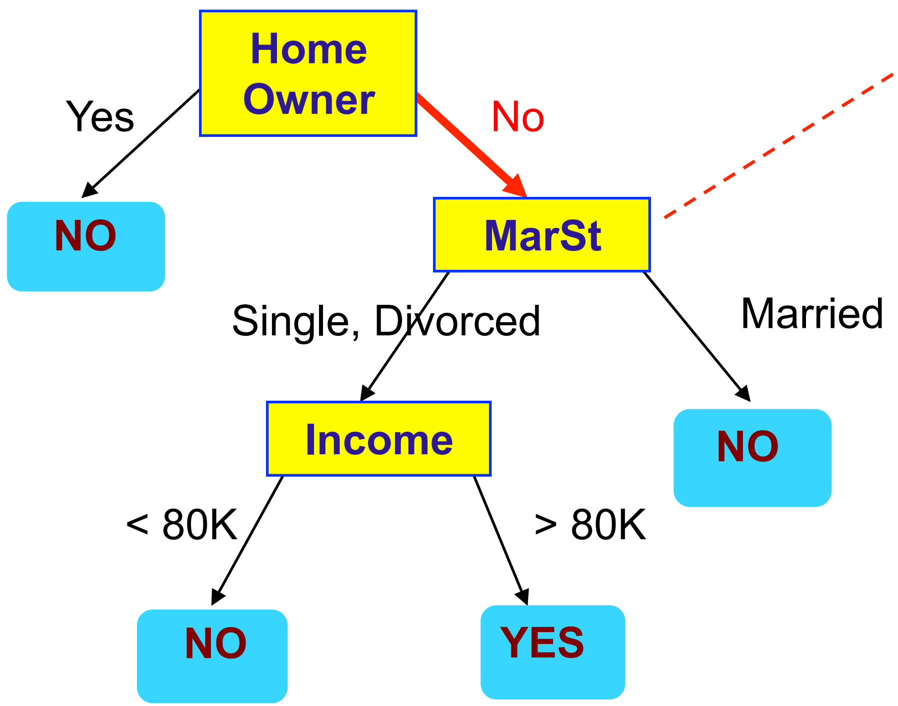
Apply Model to Test Data



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

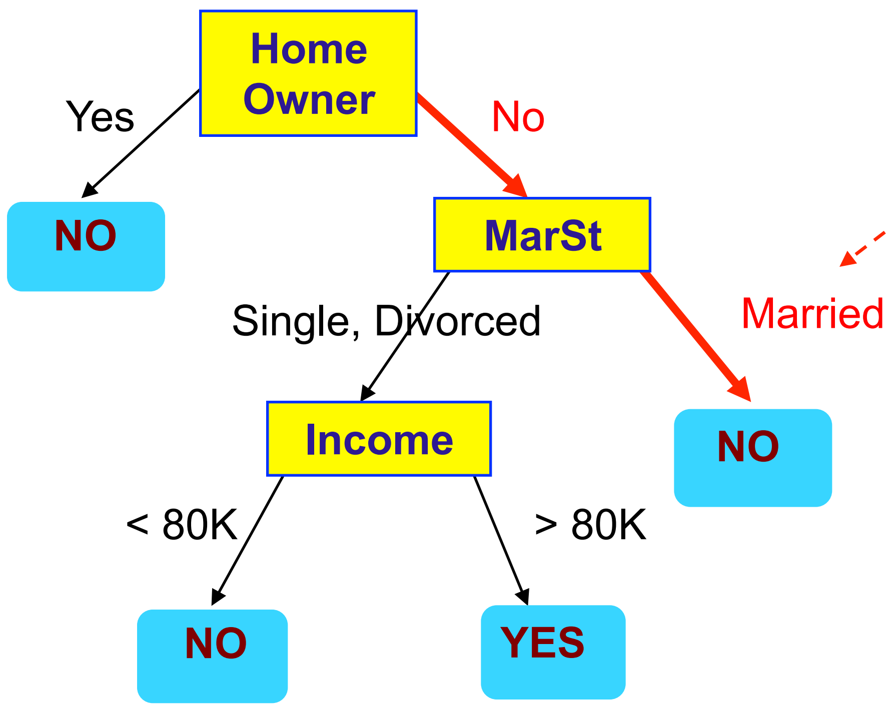
Apply Model to Test Data



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

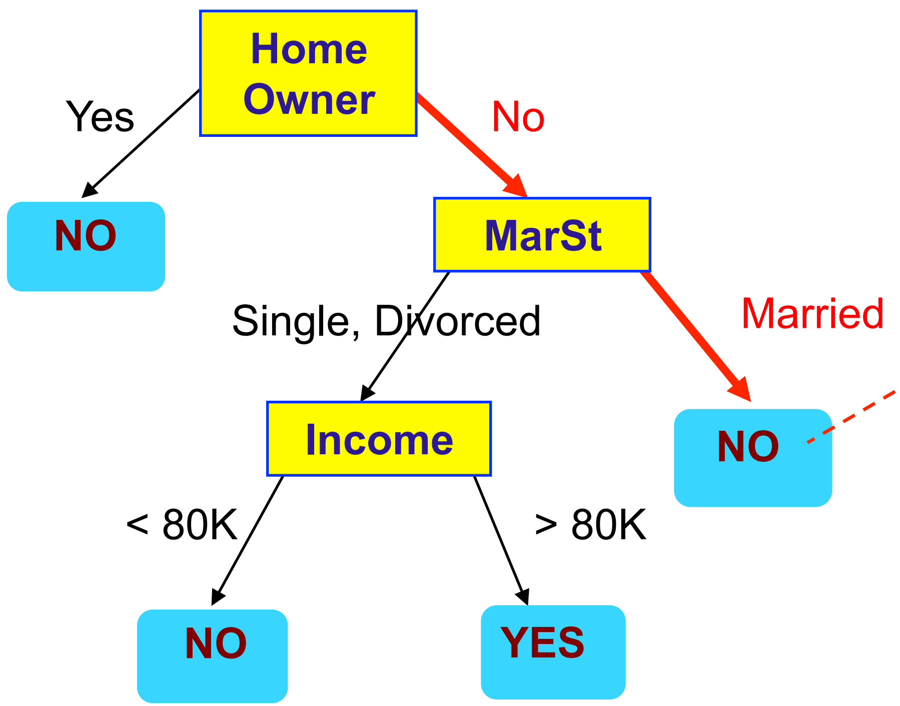
Apply Model to Test Data



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Apply Model to Test Data



Test Data

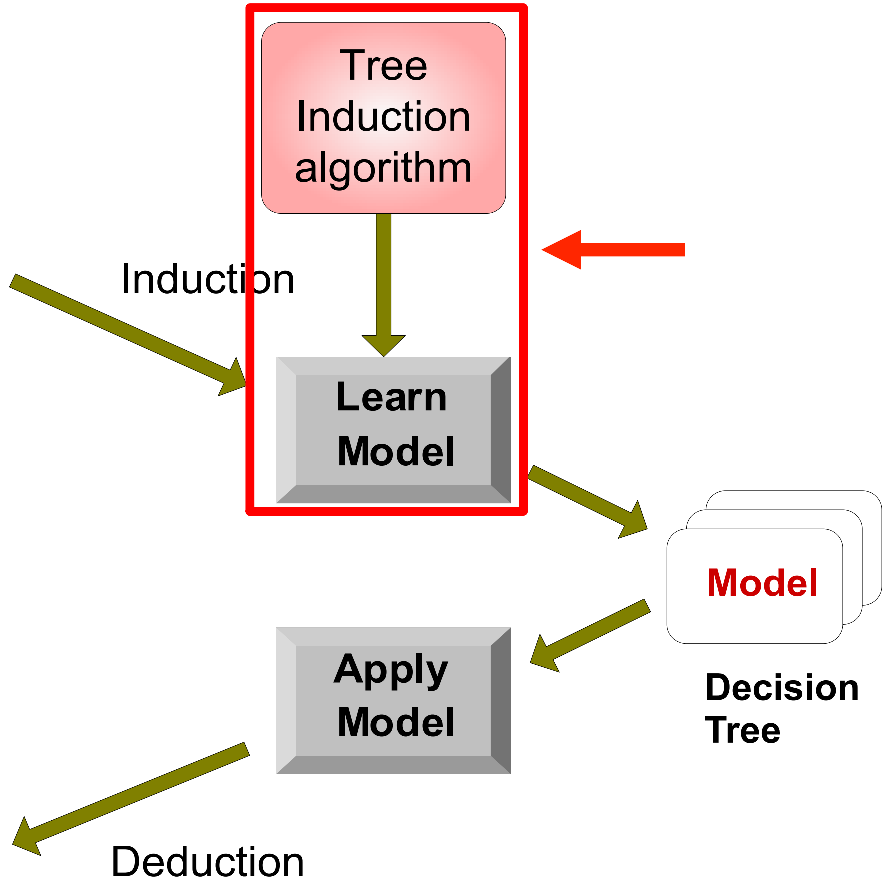
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Assign Defaulted to "No"

Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



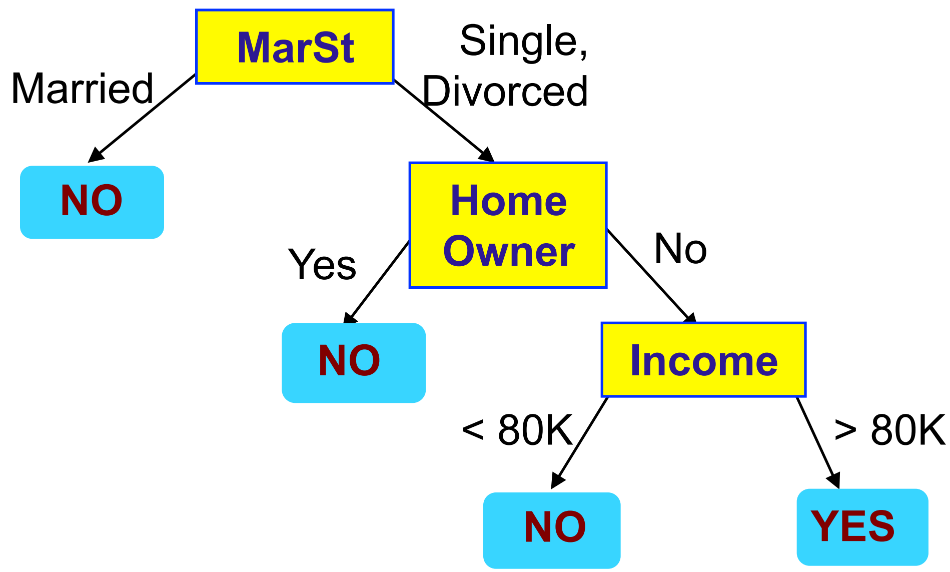
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Another Example of 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

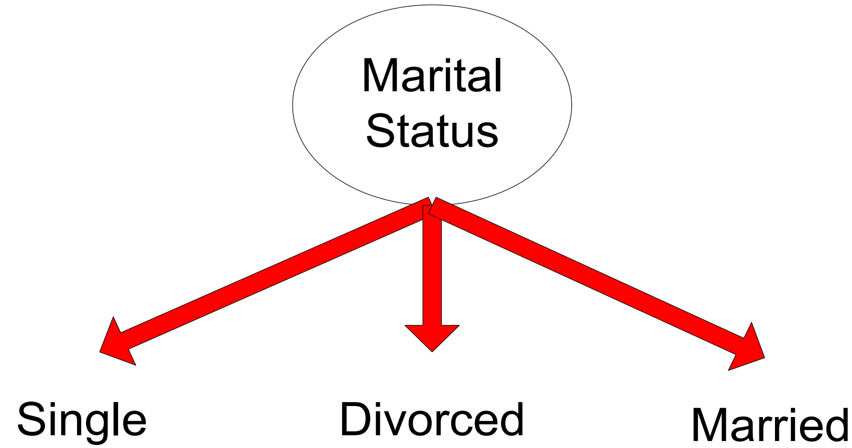


There could be more than one tree that fits the same data!

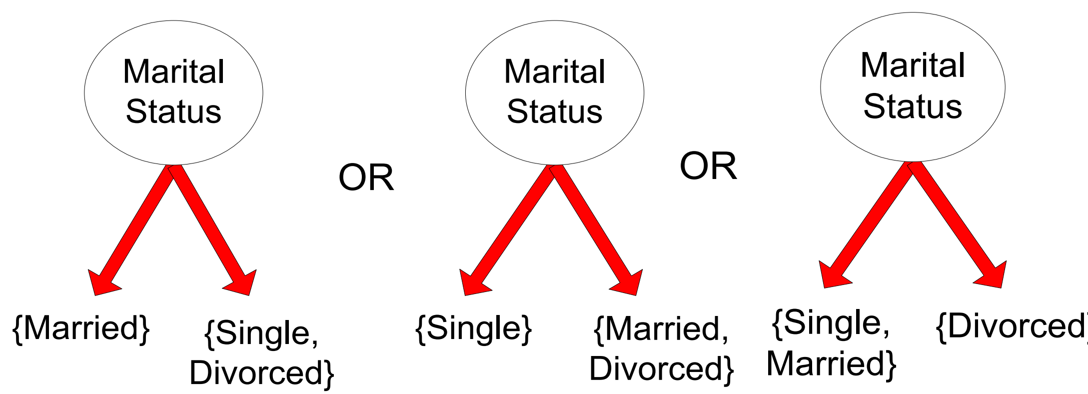
How to specify the attribute test condition?

Test Condition for Nominal Attributes

- Multi-way split:
  - Use as many partitions as distinct values.



- Binary split:
  - Divides values into two subsets



How to determine the best split?

How to determine the Best Split

- Greedy approach:
  - Nodes with purer / homogeneous class distribution are preferred

- Need a measure of node impurity:

C0: 5	C0: 9
C1: 5	C1: 1

High degree of impurity,

Non-homogeneous

Low degree of impurity,

Homogeneous

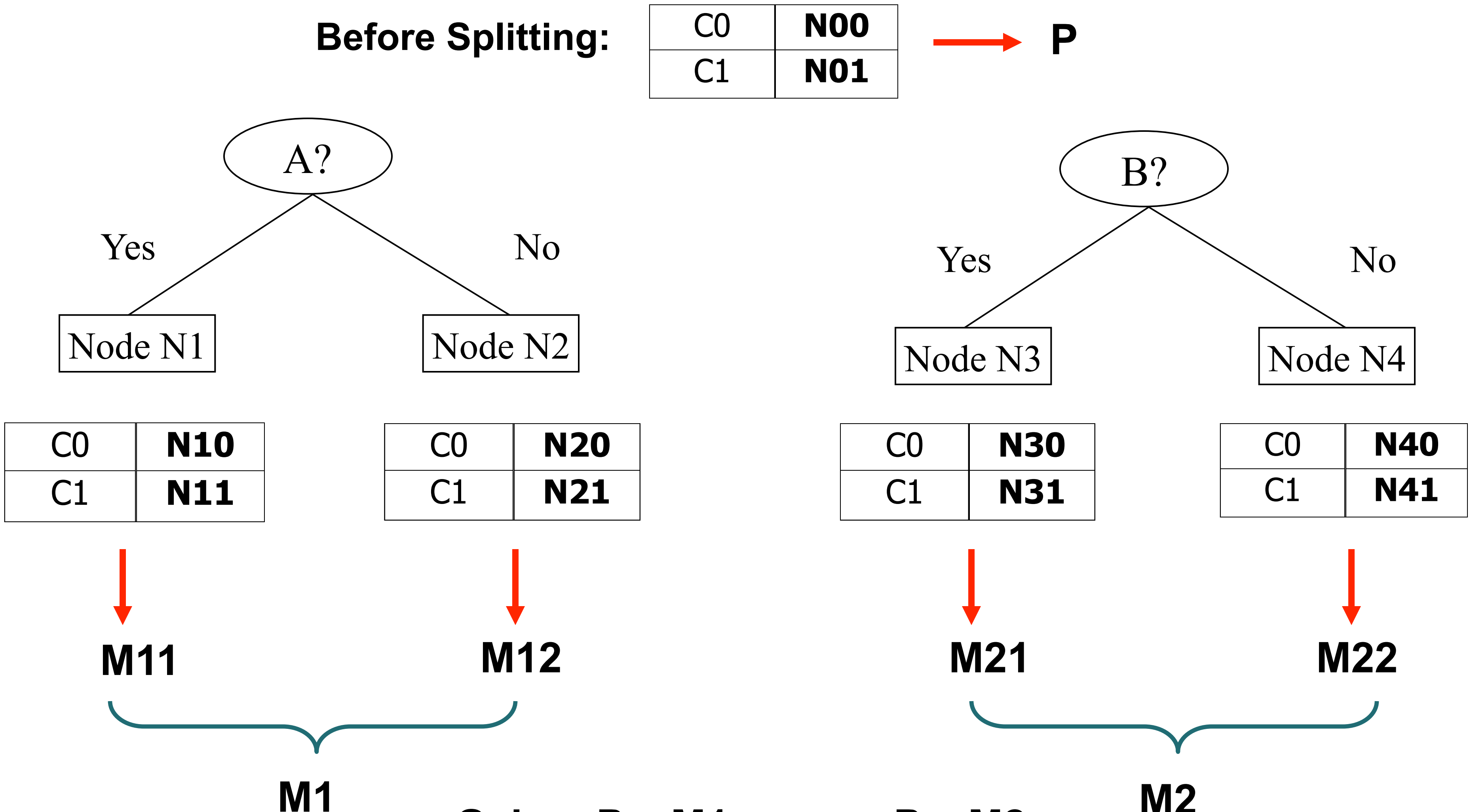
Measures of Node Impurity

- Gini Index $$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

- Entropy $$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

- Misclassification error $$Error(t) = 1 - \max_i P(i \mid t)$$

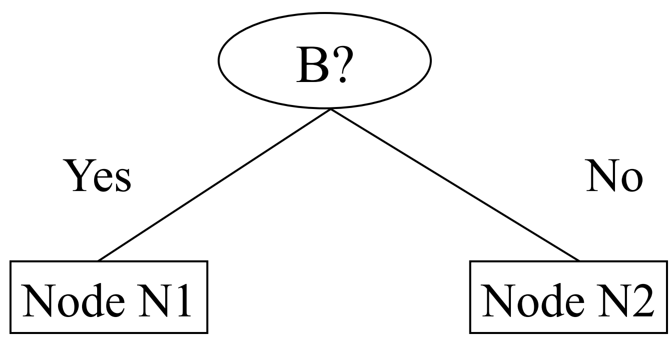
Finding the Best Split



Gain = P – M1 vs P – M2

Binary Attributes: Computing Missclassification

- Splits into two partitions
- Effect of Weighing partitions:
  - Larger and Purer Partitions are sought for.



	Parent
C1	**7**
C2	**5**
ME = 5/12	

ME(N1)
= 1 – (5/6) = 1/6

ME(N2)
= 1 – (4/6)= 2/6

	N1	N2
C1	**5**	**2**
C2	**1**	**4**
ME=3/12		

Weighted ME of N1 N2
= 6/12 * 1/6 +
6/12 * 2/6 = 3/12

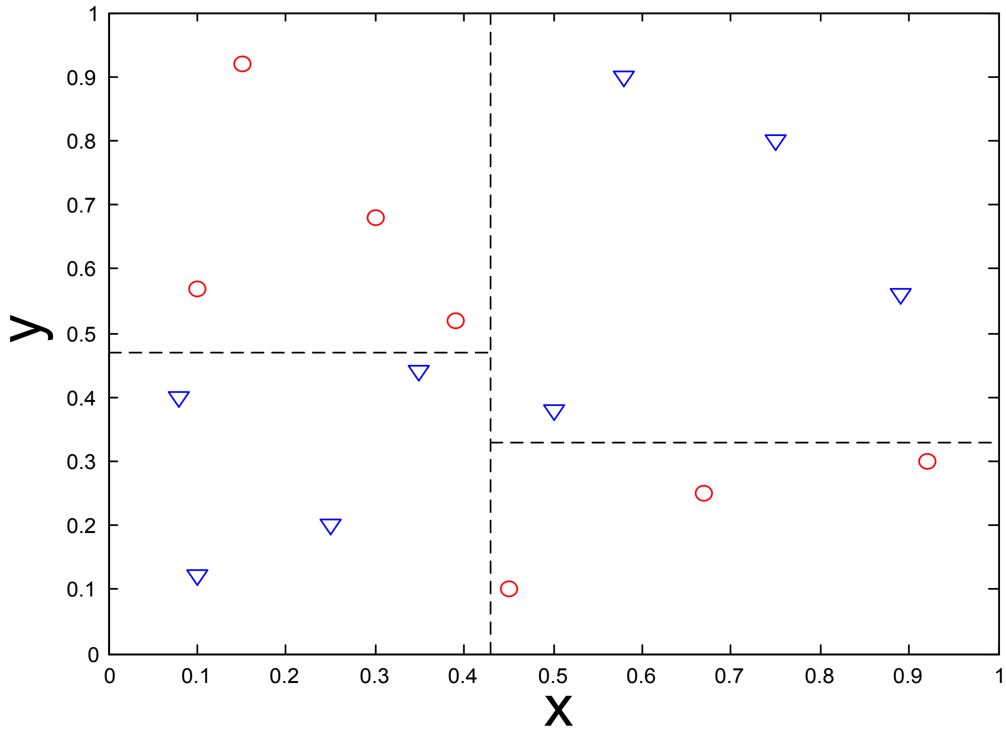
Gain = 5/12 – 3/12= 2/12

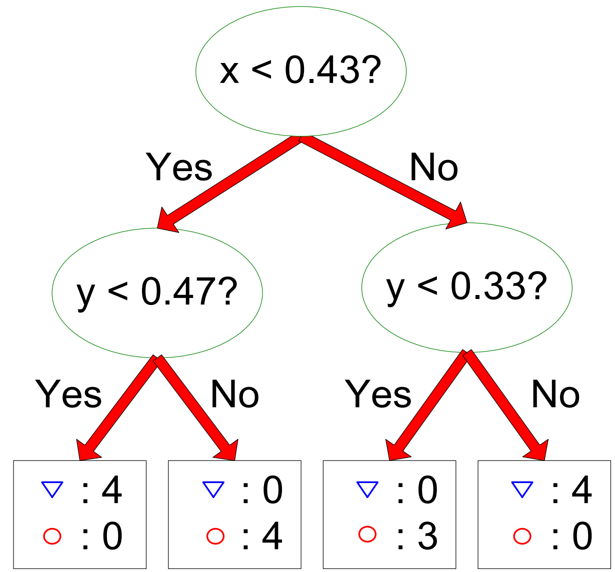
Determine when to stop splitting

Stopping Criteria for Tree Induction

- Stop expanding a node when all the records belong to the same class
- Stop expanding a node when all the records have similar attribute values
- Early termination (to be discussed later)

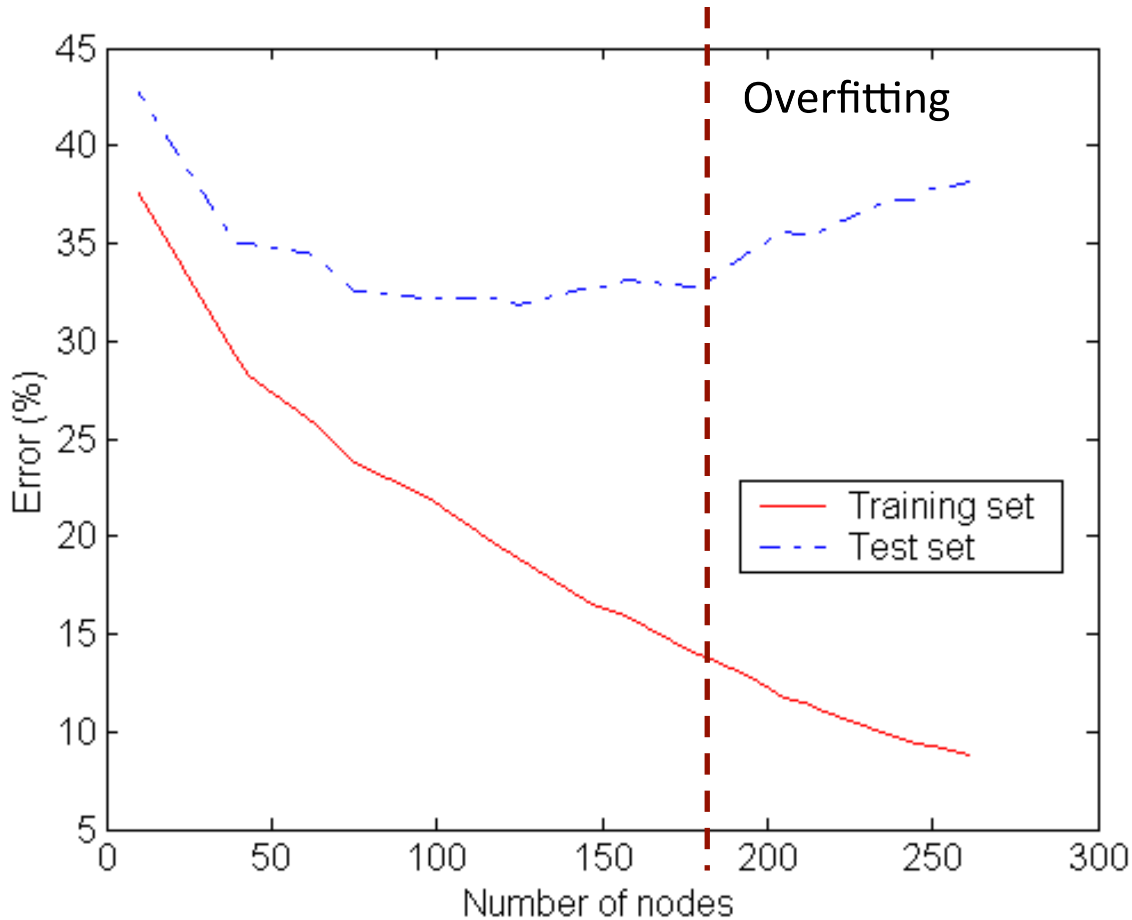
Decision Boundary





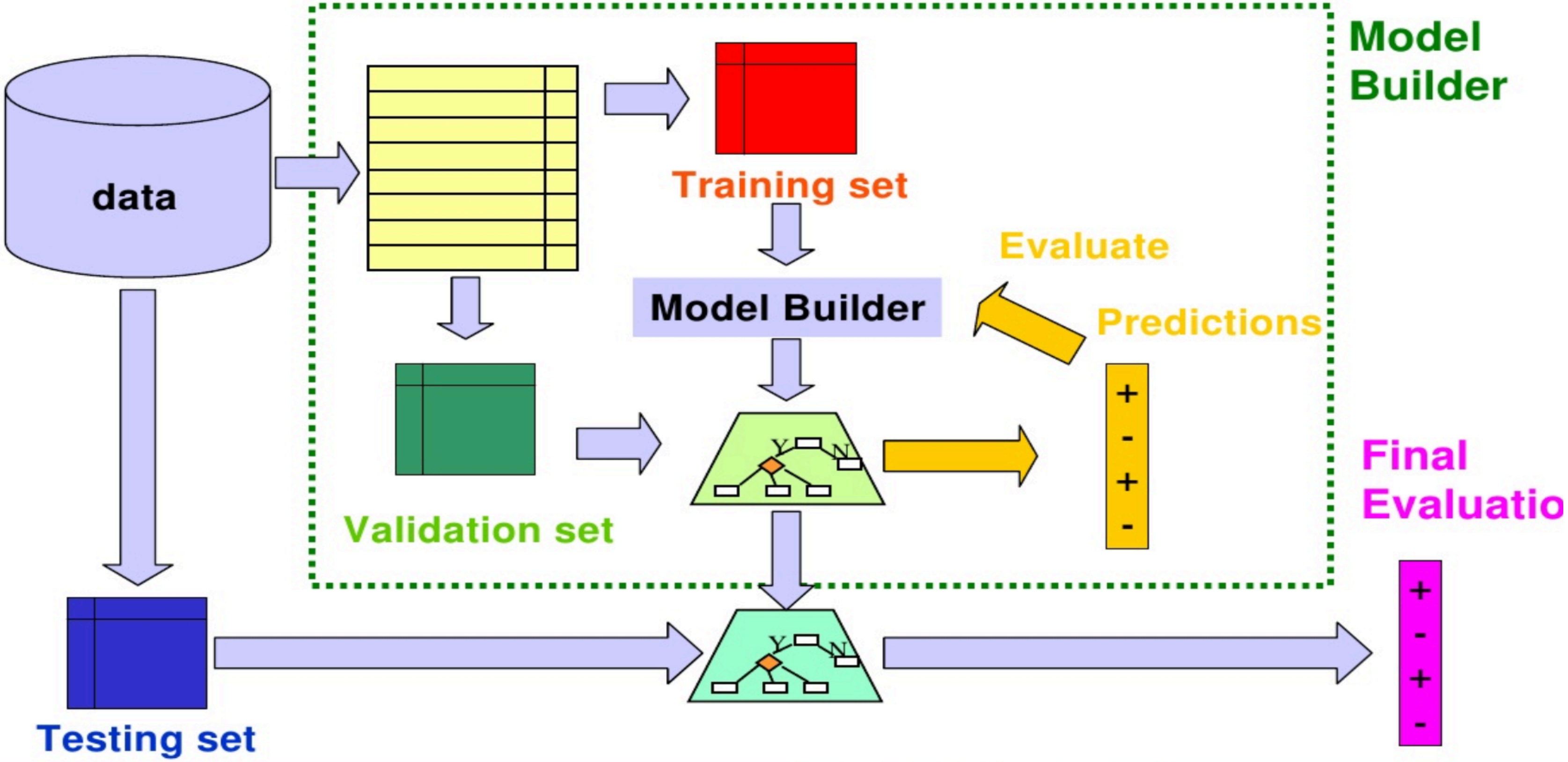
- Border line between two neighboring regions of different classes is known as decision boundary
- Decision boundary is parallel to axes because test condition involves a single attribute at-a-time

Underfitting and Overfitting



Underfitting: when model is too simple, both training and test errors are large

Evaluation: training, validation, test



Parameter Tuning

- It is important that the test data is not used in any way to create the classifier
- Some learning schemes operate in two stages:
  - **Stage 1**: builds the basic structure
  - **Stage 2**: optimizes parameter settings
  - **The test data can't be used for parameter tuning!**
  - Proper procedure uses three sets:
    - training data,
    - validation data,
    - test data
  - **Validation data is used to optimize parameters**
- Once evaluation is complete, all the data can be used to build the final classifier
- Generally, the larger the training data the better the classifier
- The larger the test data the more accurate the error estimate

Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc.
- **Confusion Matrix:**

	PREDICTED CLASS	
	Class=Yes	Class=No
ACTUAL CLASS Class=Yes	a	b
ACTUAL CLASS Class=No	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Metrics for Performance Evaluation…

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

$$\text{Precision (p)} = \frac{TP}{TP + FP}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

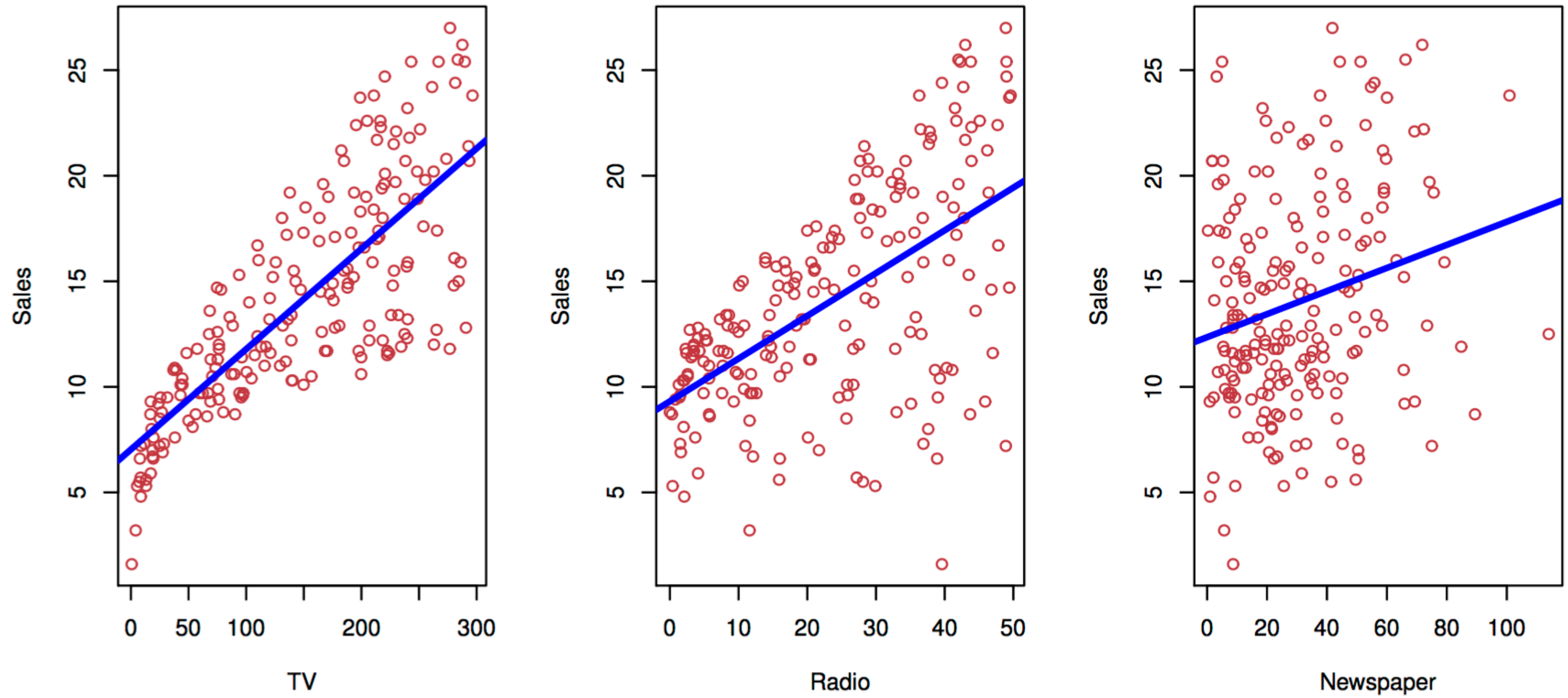
$$\text{Recall (r)} = \frac{TP}{TP + FN}$$

$$\text{F-measure (F)} = \frac{2rp}{r + p} = \frac{2TP}{2TP + FN + FP}$$

Regression

- Given: Dataset X with n tuples
  - x: Object description
  - Y : Numerical target attribute ⇒ **regression problem**
- Find a function f that describes Y as a function of the attributes X

Example



$$\texttt{Sales} \approx f(\texttt{TV}, \texttt{Radio}, \texttt{Newspaper})$$

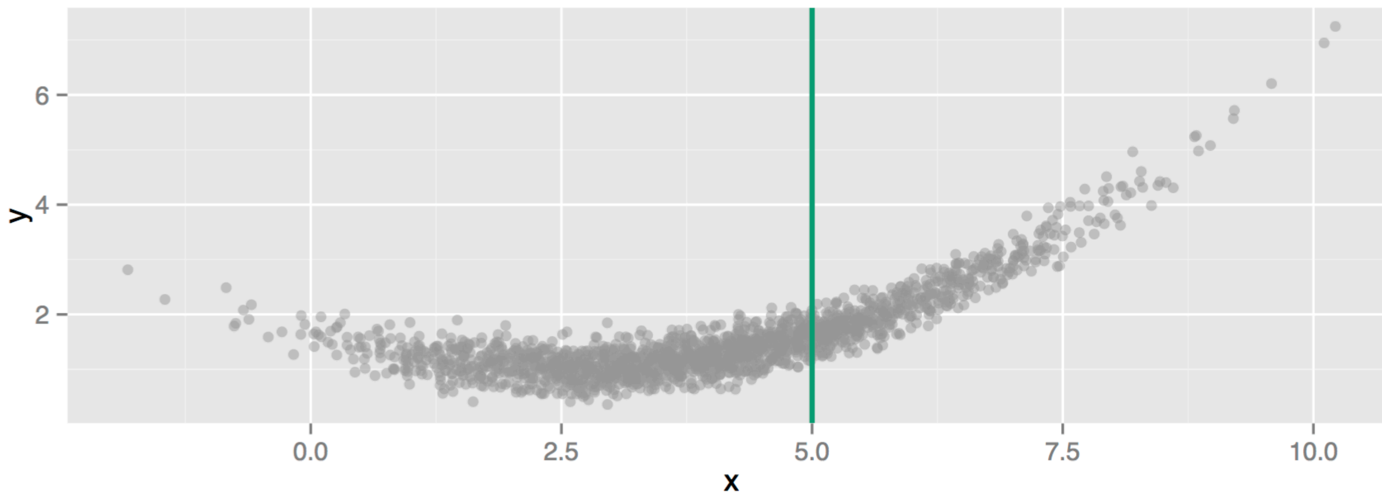
The model

Given the feature X and the target Y we describe the model as

$$Y = f(X) + \varepsilon$$

- where ε captures **measurement errors** and other **discrepancies** between the response Y and the model f(X)

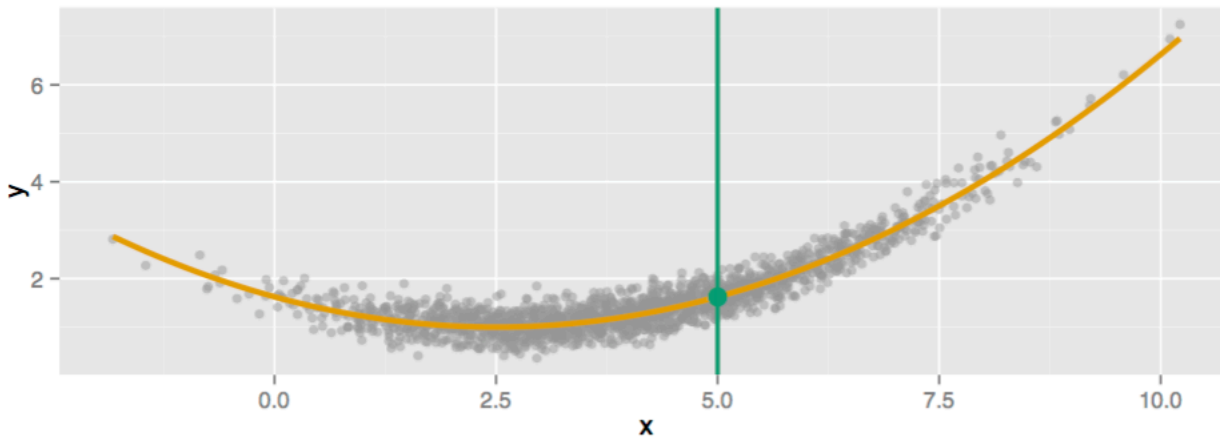
What does it mean to 'predict Y'?



- Look at $X = 5$. There are many different Y values at $X = 5$.
- When we say *predict Y at* $X = 5$, we're really asking:

What is the expected value (average) of Y at $X = 5$?

What does it mean to 'predict Y'?



Definition: Regression function

Formally, the regression function is given by $\mathrm{E}(Y \mid X = x)$. This is the *expected value of* Y at $X = x$.

- The ideal or optimal predictor of Y based on X is thus

$$f(x) = \mathrm{E}(Y \mid X = x)$$

Summary

- The ideal predictor of a response Y given inputs $X = x$ is given by the regression function

$$f(x) = \mathrm{E}(Y \mid X = x)$$

- We *don't know* what f is, so the prediction task is to estimate the regression function from the available data.

- The various prediction methods we will talk about in this class are different ways of using data to construct estimators $\hat{f}$

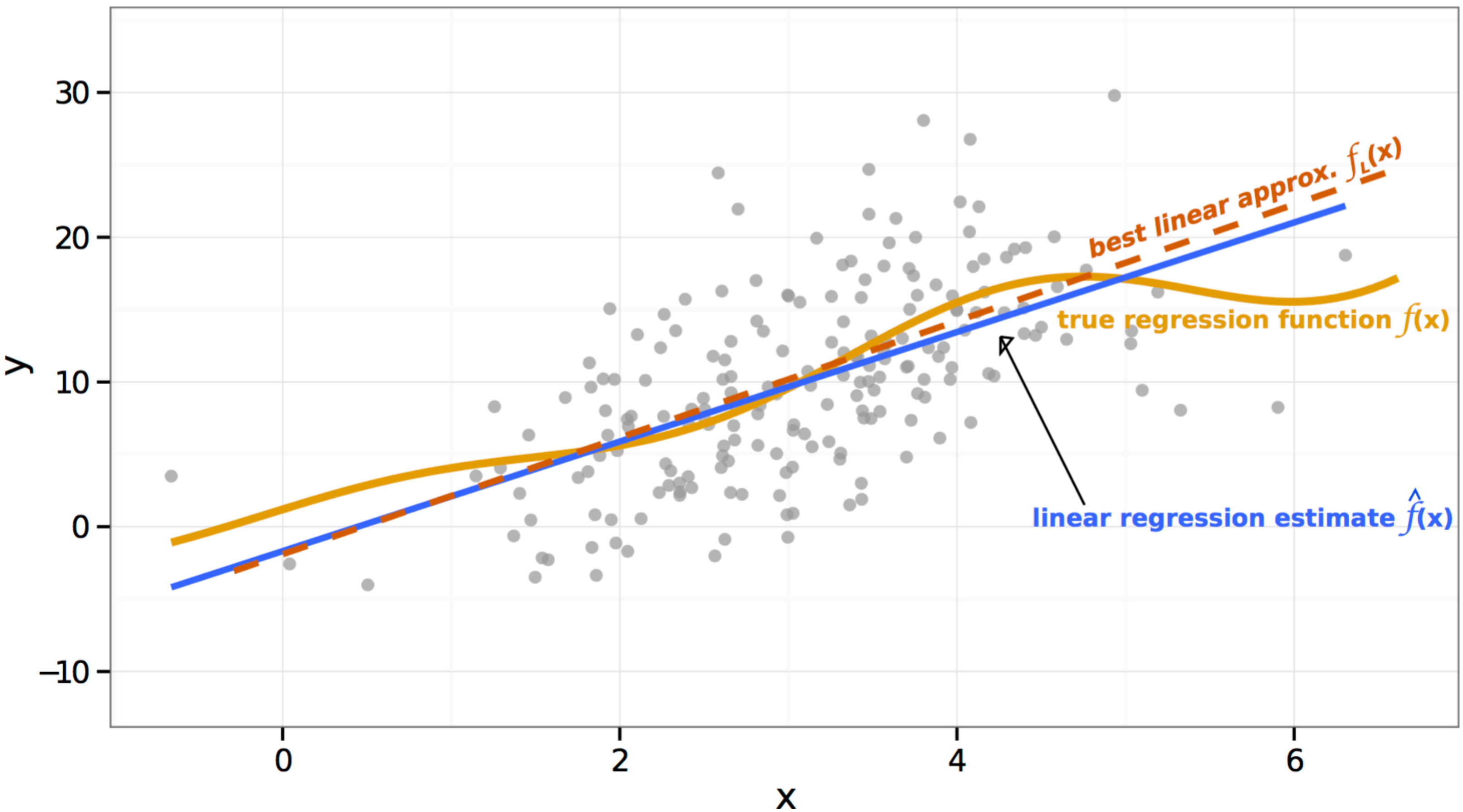
Linear Regression

- Linear regression is a *supervised learning approach* that models the dependence of Y on the covariates $X_1, X_2, \ldots, X_p$ as being linear:

$$\begin{aligned} Y &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon \\ &= \underbrace{\beta_0 + \sum_{j=1}^{p} \beta_j X_j}_{f_L(X)} + \underbrace{\epsilon}_{\text{error}} \end{aligned}$$

- The true regression function $\mathrm{E}(Y \mid X = x)$ might not be linear (it almost never is)
- Linear regression aims to estimate $f_L(X)$: the best linear approximation to the true regression function

Best linear approximation



Linear regression

- Here's the linear regression model again:

$$Y = \beta_0 + \sum_{j=1}^{p} \beta_j X_j + \epsilon$$

- The β_j, $j = 0, \ldots, p$ are called model coefficients or parameters

- Given estimates $\hat{\beta}_j$ for the model coefficients, we can predict the response at a value $x = (x_1, \ldots, x_p)$ via

$$\hat{y} = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_j$$

- The hat symbol denotes values estimated from the data

Estimation of the parameters by least squares

- Suppose that we have data (x_i, y_i), $i = 1, \ldots, n$

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \qquad X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}$$

- Linear regression estimates the parameters β_j by finding the parameter values that *minimize* the residual sum of squares (RSS):

$$\begin{aligned} \mathrm{RSS}(\hat{\beta}) &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n} \left(y_i - \left[\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \beta_p x_{ip} \right] \right)^2 \end{aligned}$$

- The quantity $e_i = y_i - \hat{y}_i$ is called a residual

Least squares picture in 1-dimension

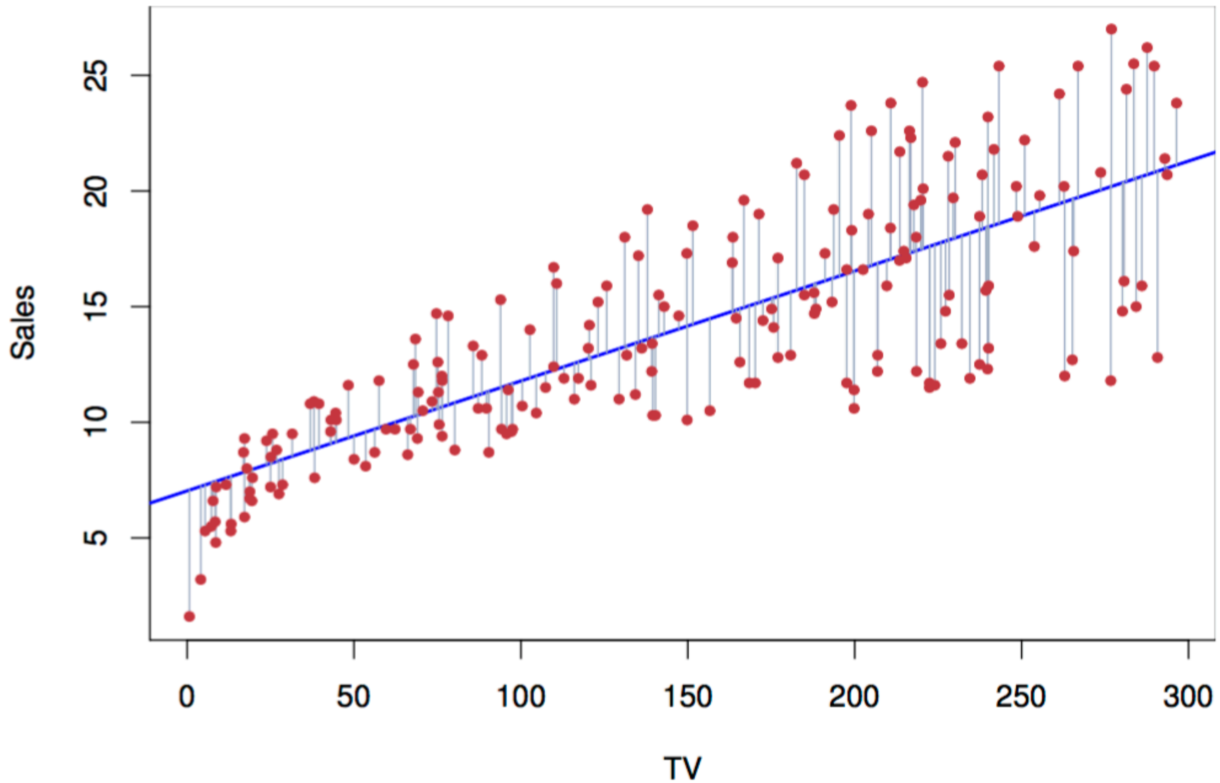
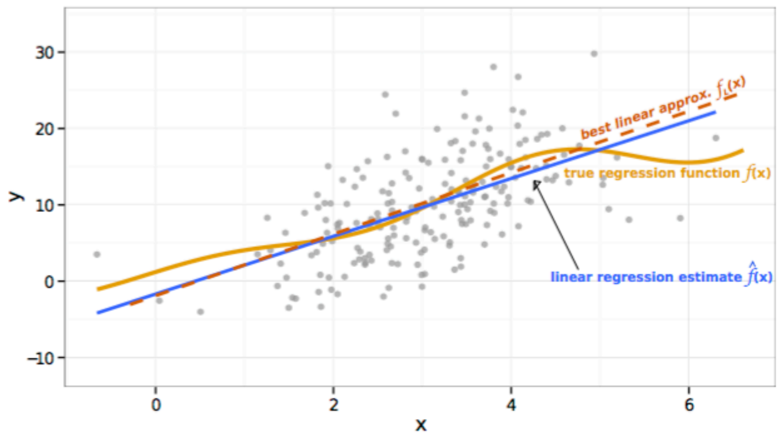
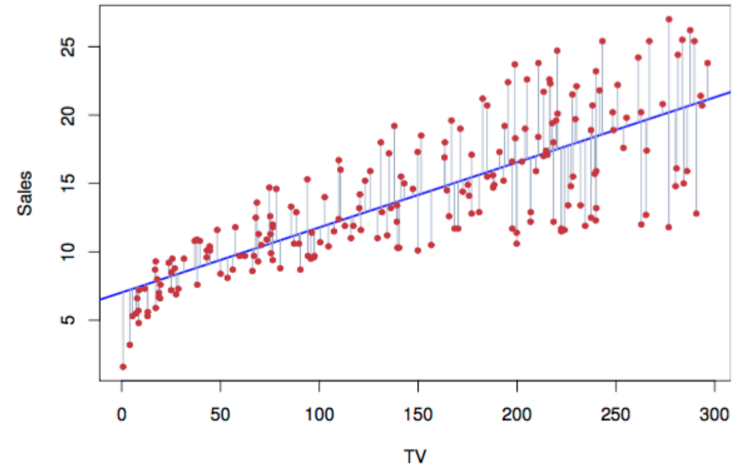


Figure: 3.1 from ISLR. Blue line shows least squares fit for the regression of `Sales` onto `TV`. Lines from observed points to the regression line illustrate the residuals. For any other choice of slope or intercept, the *sum of squared vertical distances* between that line and the observed data would be larger than that of the line shown here.

Summary





- Linear regression aims to predict the response Y by estimating the best linear predictor: the linear function that is closest to the true regression function f.
- The parameter estimates $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ are obtained by minimizing the residual sum of squares

$$\mathrm{RSS}(\hat{\beta}) = \sum_{i=1}^{n} \left(y_i - \left[\hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_{ij} \right] \right)^2$$

- Once we have our parameter estimates, we can predict y

ASSOCIATION RULES

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} → {Eggs,Coke},
{Beer, Bread} → {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. σ({Milk, Bread,Diaper}) = 2
- **Support**
  - Fraction of transactions that contain an itemset
  - E.g. s({Milk, Bread, Diaper}) = 2/5
- **Frequent Itemset**
  - An itemset whose support is greater than or equal to a *minsup* threshold

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

Definition: Association Rule

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

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**

- 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:

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

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

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

Association Rule Mining Task

- Given a set of transactions T, the goal of association rule mining is to find all rules having
  - support ≥ *minsup* threshold
  - confidence ≥ *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

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 Rules:

{Milk,Diaper} → {Beer} (s=0.4, c=0.67)
{Milk,Beer} → {Diaper} (s=0.4, c=1.0)
{Diaper,Beer} → {Milk} (s=0.4, c=0.67)
{Beer} → {Milk,Diaper} (s=0.4, c=0.67)
{Diaper} → {Milk,Beer} (s=0.4, c=0.5)
{Milk} → {Diaper,Beer} (s=0.4, c=0.5)

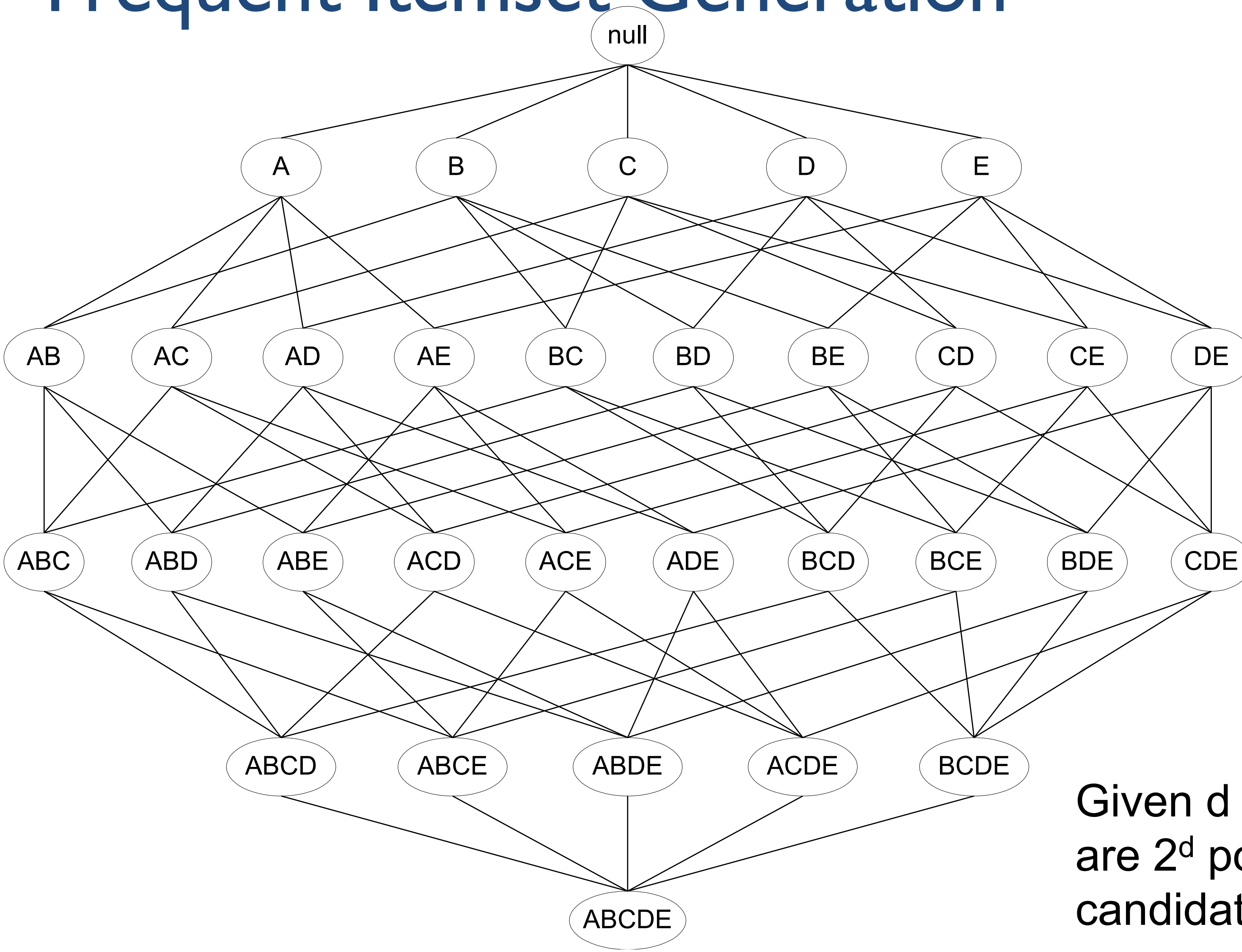
Observations:

- All the above rules are binary partitions of the same itemset:
  {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 ≥ 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

Reducing Number of Candidates

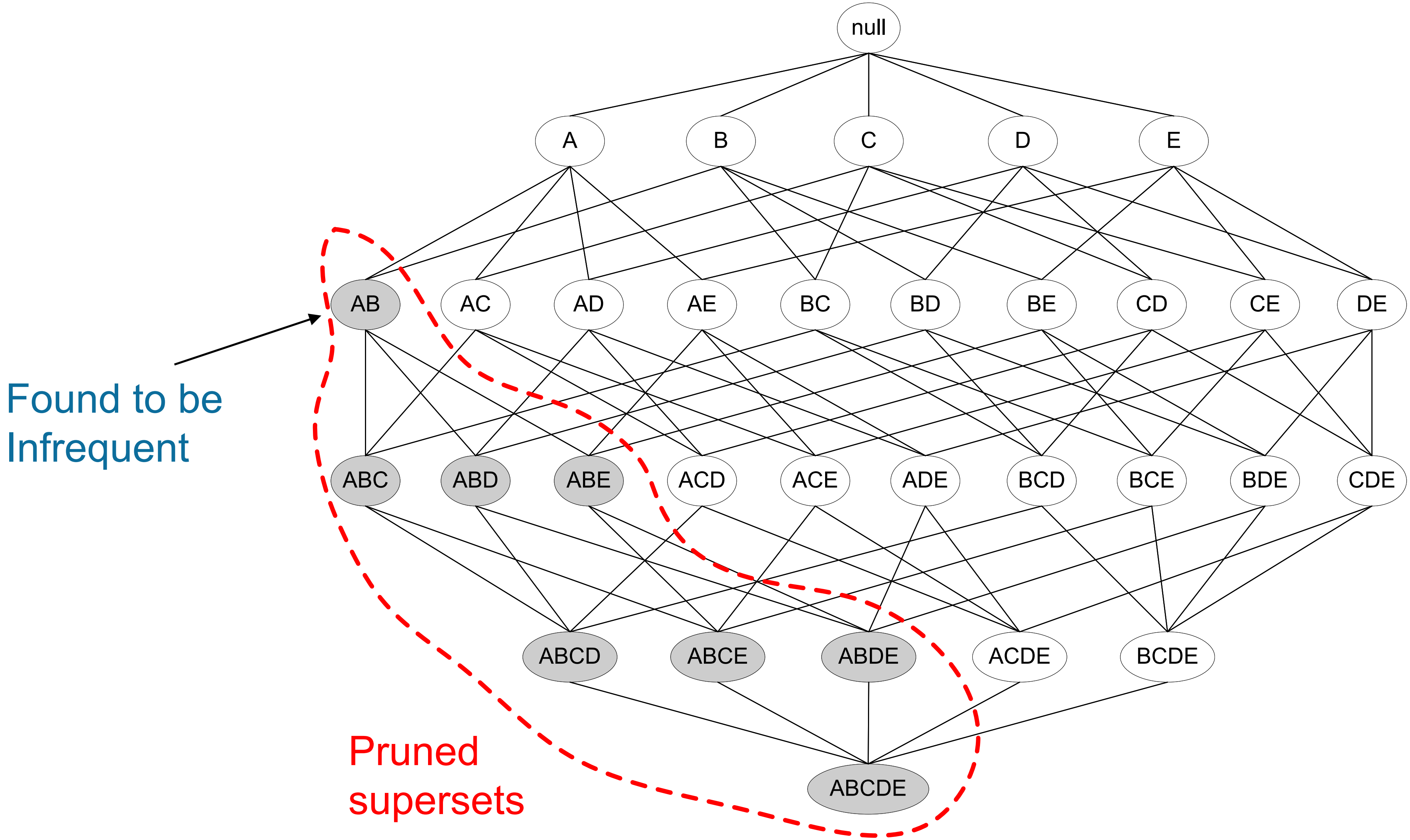
- Apriori principle:
  - If an itemset is frequent, then all of its subsets must also be frequent

- 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

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

Items (1-itemsets)

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

Minimum Support = 3

If every subset is considered,

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

$6 + 15 + 20 = 41$

With support-based pruning,

$6 + 6 + 4 = 16$

Illustrating Apriori Principle

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

Items (1-itemsets)

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

Minimum Support = 3

If every subset is considered,

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

6 + 15 + 20 = 41

With support-based pruning,

6 + 6 + 4 = 16

Illustrating Apriori Principle

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

Items (1-itemsets)

Itemset
{Bread,Milk}
{Bread, Beer }
{Bread,Diaper}
{Beer, Milk}
{Diaper, Milk}
{Beer,Diaper}

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

If every subset is considered,

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

$6 + 15 + 20 = 41$

With support-based pruning,

$6 + 6 + 4 = 16$

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**
{Beer, Bread}	**2**
{Bread,Diaper}	**3**
{Beer,Milk}	**2**
{Diaper,Milk}	**3**
{Beer,Diaper}	**3**

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

If every subset is considered,

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

$6 + 15 + 20 = 41$

With support-based pruning,

$6 + 6 + 4 = 16$

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)

Minimum Support = 3

Triplets (3-itemsets)

Itemset
{ Beer, Diaper, Milk}
{ Beer,Bread,Diaper}
{Bread, Diaper, Milk}
{ Beer, Bread, Milk}

If every subset is considered,

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

$6 + 15 + 20 = 41$

With support-based pruning,

$6 + 6 + 4 = 16$

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)

Minimum Support = 3

Triplets (3-itemsets)

Itemset	Count
{ Beer, Diaper, Milk}	2
{ Beer,Bread, Diaper}	2
{Bread, Diaper, Milk}	2
{Beer, Bread, Milk}	1

If every subset is considered,

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

$6 + 15 + 20 = 41$

With support-based pruning,

$6 + 6 + 4 = 16$

$6 + 6 + 1 = 13$

Multidimensional AR

Associations between values of different attributes :

CID	**nationality**	**age**	**income**
1	Italian	50	low
2	French	40	high
3	French	30	high
4	Italian	50	medium
5	Italian	45	high
6	French	35	high

RULES:

nationality = French $\Rightarrow$ **income** = high [50%, 100%]

income = high $\Rightarrow$ **nationality** = French [50%, 75%]

age = 50 $\Rightarrow$ **nationality** = Italian [33%, 100%]

Single-dimensional vs Multi-dimensional AR

Multi-dimensional

<1, Italian, 50, low>

<2, French, 45, high>

Single-dimensional

<1, {nat/Ita, age/50, inc/low}>

<2, {nat/Fre, age/45, inc/high}>

Quantitative Association Rules

Problem: too many distinct values for numerical attributes

Solution: transform quantitative attributes in categorical ones via discretization

CID	Age	Married	NumCars
1	23	No	1
2	25	Yes	1
3	29	No	0
4	34	Yes	2
5	38	Yes	2

[Age: 30..39] and [Married:Yes] $\Rightarrow$ [NumCars:2]

support = 40%
confidence = 100%

SEQUENTIAL PATTERNS

Sequence Databases

- A sequence database consists of ordered elements or events
- Transaction databases vs. sequence databases

A *transaction database*

TID	itemsets
10	a, b, d
20	a, c, d
30	a, d, e
40	b, e, f

A *sequence database*

SID	sequences
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

Applications

- Applications of sequential pattern mining
  - Customer shopping sequences:
    - First buy computer, then CD-ROM, and then digital camera, within 3 months.
  - Medical treatments, natural disasters (e.g., earthquakes), science & eng. processes, stocks and markets, etc.
  - Telephone calling patterns, Weblog click streams
  - DNA sequences and gene structures

Subsequence vs. super sequence

- A sequence is an ordered list of events, denoted $< e_1\ e_2 \ldots e_l >$
- Given two sequences $\alpha =< a_1\ a_2 \ldots a_n >$ and $\beta =< b_1\ b_2 \ldots b_m >$
- α is called a subsequence of β, denoted as $\alpha \subseteq \beta$, if there exist integers $1 \leq j_1 < j_2 < \ldots < j_n \leq m$ such that $a_1 \subseteq b_{j1}, a_2 \subseteq b_{j2}, \ldots, a_n \subseteq b_{jn}$
- β is a super sequence of α
  - E.g. α=< (ab), d> and β=< (abc), (de)>

What Is Sequential Pattern Mining?

- Given a set of sequences and support threshold, find the complete set of *frequent* subsequences

A sequence : < (ef) (ab) (df) c b >

A sequence database

SID	sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

An element may contain a set of items. Items within an element are unordered and we list them alphabetically.

<a(bc)dc> is a subsequence of <a(abc)(ac)d(cf)>

Given support threshold min_sup =2, <(ab)c> is a sequential pattern

The Apriori Property of Sequential Patterns

- A basic property: Apriori (Agrawal & Sirkant' 94)
  - If a sequence *S* is not frequent, then none of the super-sequences of *S* is frequent
  - E.g, <hb> is infrequent so do <hab> and <(ah)b>

Seq. ID	Sequence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

Given support threshold
min_sup =2

GSP—Generalized Sequential Pattern Mining

- GSP (Generalized Sequential Pattern) mining algorithm
- Outline of the method
  - Initially, every item in DB is a candidate of length-1
  - for each level (i.e., sequences of length-k) do
    - scan database to collect support count for each candidate sequence
    - generate candidate length-(k+1) sequences from length-k frequent sequences using Apriori
  - repeat until no frequent sequence or no candidate can be found
- Major strength: Candidate pruning by Apriori

Finding Length-1 Sequential Patterns

- Initial candidates:
  - <a>, <b>, <c>, <d>, <e>, <f>, <g>, <h>
- Scan database once, count support for candidates

min_sup =2

Seq. ID	Sequence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

Cand	Sup
<a>	**3**
<b>	**5**
<c>	**4**
<d>	**3**
<e>	**3**
<f>	**2**
~~<g>~~	1
~~<h>~~	1

Generating Length-2 Candidates

51 length-2 Candidates

	<a>	<b>	<c>	<d>	<e>	<f>
<a>	<aa>	<ab>	<ac>	<ad>	<ae>	<af>
<b>	<ba>	<bb>	<bc>	<bd>	<be>	<bf>
<c>	<ca>	<cb>	<cc>	<cd>	<ce>	<cf>
<d>	<da>	<db>	<dc>	<dd>	<de>	<df>
<e>	<ea>	<eb>	<ec>	<ed>	<ee>	<ef>
<f>	<fa>	<fb>	<fc>	<fd>	<fe>	<ff>

	<a>	<b>	<c>	<d>	<e>	<f>
<a>		<(ab)>	<(ac)>	<(ad)>	<(ae)>	<(af)>
<b>			<(bc)>	<(bd)>	<(be)>	<(bf)>
<c>				<(cd)>	<(ce)>	<(cf)>
<d>					<(de)>	<(df)>
<e>						<(ef)>
<f>						

Apriori prunes 44.57% candidates

Finding Lenth-2 Sequential Patterns

- Scan database one more time, collect support count for each length-2 candidate
- There are 19 length-2 candidates which pass the minimum support threshold
  - They are length-2 sequential patterns

The GSP Mining Process

Cand. cannot pass sup. threshold

Cand. not in DB at all

5th scan: 1 cand. 1 length-5 seq. pat. — <(bd)cba>

4th scan: 8 cand. 6 length-4 seq. pat. — <abba> <(bd)bc> …

3rd scan: 46 cand. 19 length-3 seq. pat. 20 cand. not in DB at all — <abb> <aab> <aba> <baa> <bab> …

2nd scan: 51 cand. 19 length-2 seq. pat. 10 cand. not in DB at all — <aa> <ab> … <af> <ba> <bb> … <ff> <(ab)> … <(ef)>

1st scan: 8 cand. 6 length-1 seq. pat. — <a> <b> <c> <d> <e> <f> <g> <h>

min_sup =2

Seq. ID	Sequence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>