Social Media Mining

Data Mining Essentials

Dear instructors/users of these slides:

Please feel free to include these slides in your own material, or modify them as you see fit. If you decide to incorporate these slides into your presentations, please include the following note:

R. Zafarani, M. A. Abbasi, and H. Liu, *Social Media Mining: An Introduction*, Cambridge University Press, 2014.
Free book and slides at **http://socialmediamining.info/**

or include a link to the website:
http://socialmediamining.info/

Introduction

Data production rate has increased dramatically (**Big Data**) and we are able store much more data

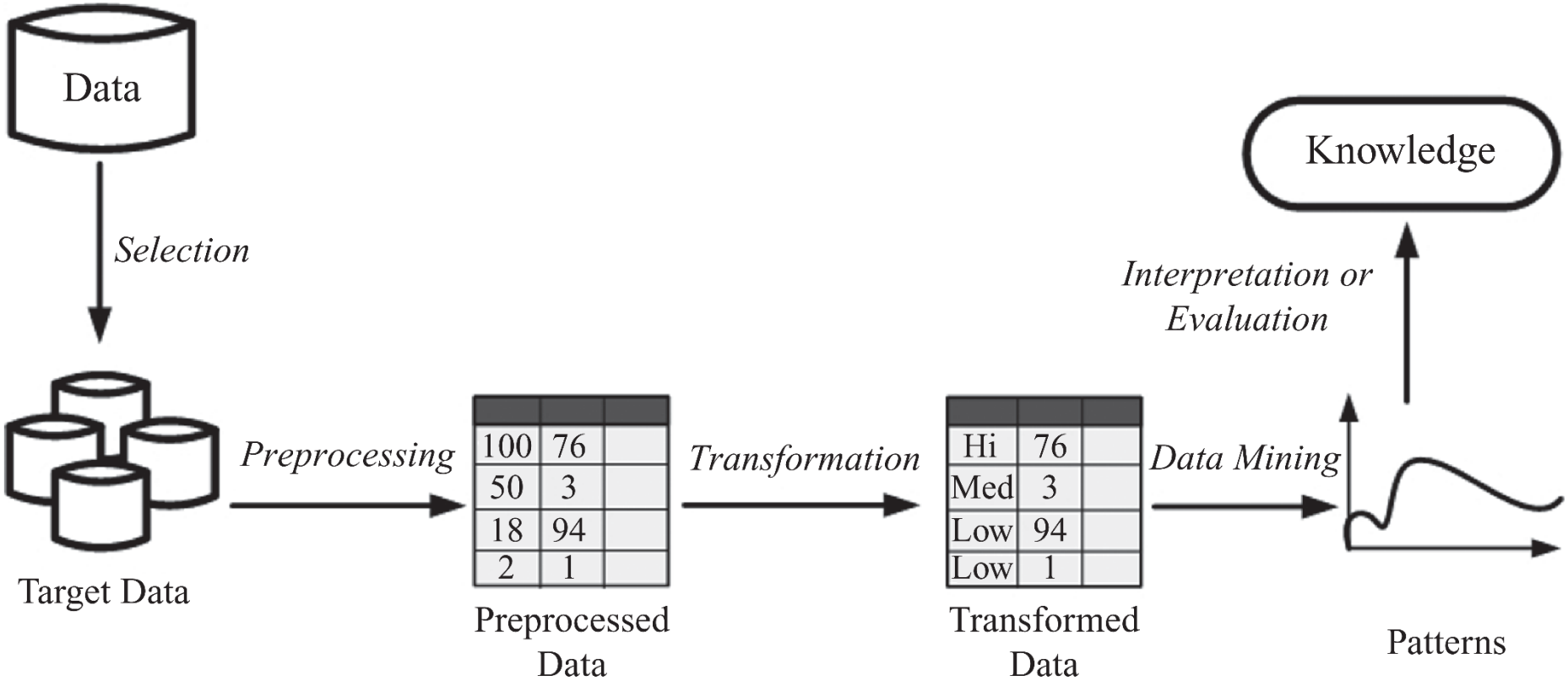
- E.g., purchase data, social media data, cellphone data

Businesses and customers need useful or actionable knowledge to gain insight from raw data for various purposes

- It's not just searching data or databases

The process of extracting useful patterns from raw data is known as Knowledge Discovery in Databases (KDD)

KDD Process


Data
Selection
Target Data
Preprocessing
100 76
50 3
18 94
2 1
Preprocessed Data
Transformation
Hi 76
Med 3
Low 94
Low 1
Transformed Data
Data Mining
Patterns
Interpretation or Evaluation
Knowledge

Collecting Data on Social Media

- Collect Raw Data
  - Use site provided APIs
    - Flickr's: https://www.flickr.com/services/api/
  - Scrape information directly

- Use Provided Repositories
  - http://socialcomputing.asu.edu
  - http://snap.Stanford.edu
  - https://github.com/caesar0301/awesome-public-datasets

Data Mining

Data Mining: the **process** of discovering **hidden** and **actionable** patterns from data

It utilizes methods at the intersection of artificial intelligence, machine learning, statistics, and database systems

- Extracting/"mining" knowledge from large-scale data (big data)
- Data-driven discovery and modeling of hidden patterns in big data
- Extracting information/knowledge from data that is
  - implicit,
  - previously unknown,
  - unexpected, and
  - potentially useful

Data Mining vs. Databases

- **Data mining** is the *process* of extracting hidden and actionable patterns from data

- **Database systems** store and manage data
  - Queries return part of stored data
  - Queries do not extract hidden patterns

- Examples of querying databases
  - Find all employees with income more than $250K
  - Find top spending customers in last month
  - Find all students from *engineering college* with GPA more than average

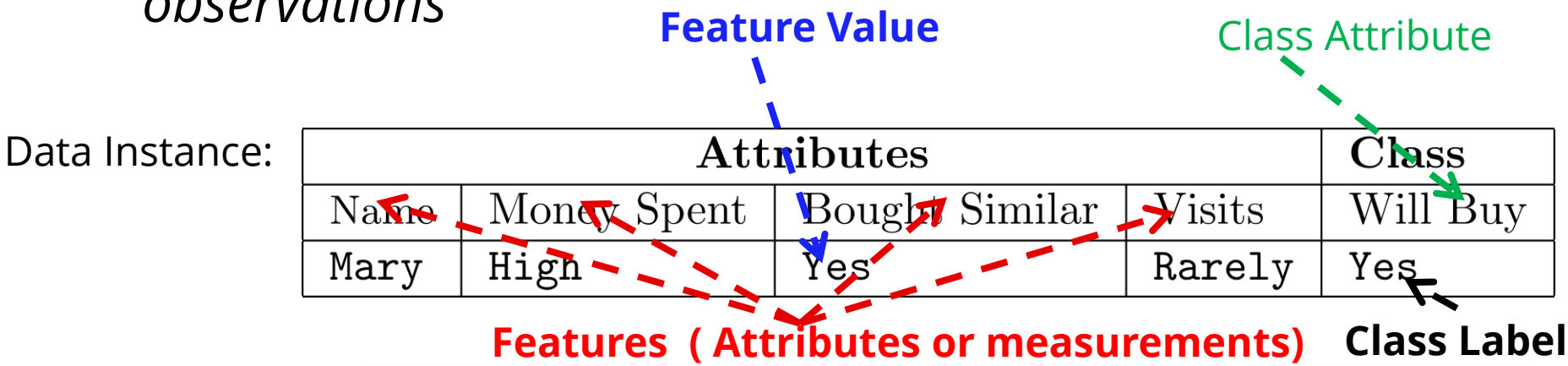
Examples of Data Mining Applications

- **Fraud/Spam Detections:** Identifying fraudulent transactions of a credit card or spam emails
  - You are given a user's purchase history and a new transaction, identify whether the transaction is fraud or not;
  - Determine whether a given email is spam or not
- **Frequent Patterns:** Extracting purchase patterns from existing records
  - beer ⇒ dippers (80%)
- **Forecasting:** Forecasting future sales and needs according to some given samples
- **Finding Like-Minded Individuals:** Extracting groups of like-minded people in a given network

Data

Data Instances

- In the KDD process,
  - Data is in a tabular format (a set of **instances**)
- Each instance is a collection of properties and features related to an object or person
  - A patient's medical record
  - A user's profile
  - A gene's information
- Instances are also called *points*, *data points*, or *observations*



Data Instances

- Predicting whether an individual who visits an online book seller is going to buy a specific book

Attributes				Class
Name	Money Spent	Bought Similar	Visits	Will Buy
John	High	Yes	Frequently	?
Mary	High	Yes	Rarely	Yes

Unlabeled Example

Labeled Example

- Features can be
  - **Continues**: values are numeric values
    - Money spent: $25
  - **Discrete**: can take a number of values
    - Money spent: {high, normal, low}

Data Types + Permissible Operations (statistics)

- **Nominal**
  - **Operations**:
    - Mode (most common feature value), Equality Comparison
  - E.g., {male, female}
- **Ordinal**
  - Feature values have an intrinsic order to them, but the difference is not defined
  - **Operations**:
    - same as nominal, feature value rank
  - E.g., {Low, medium, high}
- **Interval**
  - **Operations:**
    - Addition and subtractions are allowed whereas divisions and multiplications are not
  - E.g., 3:08 PM, calendar dates
- **Ratio**
  - **Operations:**
    - divisions and multiplications are allowed
  - E.g., Height, weight, money quantities

Sample Dataset – Twitter Users

Activity	*Date Joined*	*Number of Followers*	*Verified Account?*	*Has Profile Picture?*
High	2015	50	FALSE	no
High	2013	300	TRUE	no
Average	2011	860000	FALSE	yes
Low	2012	96	FALSE	yes
High	2008	8,000	FALSE	yes
Average	2009	5	TRUE	no
Very High	2010	650,000	TRUE	yes
Low	2010	95	FALSE	no
Average	2011	70	FALSE	yes
Very High	2013	80,000	FALSE	yes
Low	2014	70	TRUE	yes
Average	2013	900	TRUE	yes
High	2011	7500	FALSE	yes
Low	2010	910	TRUE	no

Ordinal **Interval** **Ratio** **Nominal** **Nominal**

Text Representation

- The most common way to represent documents is to transform them into vectors
  - Process them with linear algebraic operations
- This representation is called "***Bag of Words***"
  - Vector Space Model
- Weights for words can be assigned by **TF-IDF**

Vector Space Model

- Consider a set of documents D
- Each document is a set of words

- **Goal**: convert these documents to vectors

$$d_i = (w_{1,i}, w_{2,i}, \ldots, w_{N,i})$$

  - d_i : document i
  - $w_{j,i}$: the weight for word j in document i

How to set $w_{j,i}$

- Set $w_{j,i}$ to 1 when the word j exists in document i and 0 when it does not.
- We can also set $w_{j,i}$ to the number, of times the word j is observed in document i (frequency)

Vector Space Model: An Example

- **Documents:**
  - d_1: social media mining
  - d_2: social media data
  - d_3: financial market data
- Reference vector (**Dictionary**):
  - (social, media, mining, data, financial, market)

- Vector representation:

	social	media	mining	data	financial	market
d_1	1	1	1	0	0	0
d_2	1	1	0	1	0	0
d_3	0	0	0	1	1	1

TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF of term (word) t, document d, and document corpus D is calculated as follows:

$$w_{j,i} = tf_{j,i} \times idf_j$$

$tf_{j,i}$ is the frequency of word j in document i

The total number of documents in the corpus

$$idf_j = \log_2 \frac{|D|}{|\{\text{document} \in D \mid j \in \text{document}\}|}$$

The number of documents where the term j appears

TF-IDF: An Example

Document d_1 contains 100 words

- Word "apple" appears 10 times in d_1
- Word "orange" appears 20 times in d_1

We have $|D| = 20$ documents

- Word "apple" only appears in document d_1
- Word "orange" appears in all 20 documents

$$tf - idf(\text{“}apple\text{”}, d_1) = 10 \times \log_2 \frac{20}{1} = 43.22$$

$$tf - idf(\text{“}orange\text{”}, d_1) = 20 \times \log_2 \frac{20}{20} = 0$$

TF-IDF: An Example

- Documents:
  - d_1: social media mining
  - d_2: social media data
  - d_3: financial market data

$$
\begin{aligned}
idf_{\text{social}} &= \log_2(3/2) = 0.584 \\
idf_{\text{media}} &= \log_2(3/2) = 0.584 \\
idf_{\text{mining}} &= \log_2(3/1) = 1.584 \\
idf_{\text{data}} &= \log_2(3/2) = 0.584 \\
idf_{\text{financial}} &= \log_2(3/1) = 1.584 \\
idf_{\text{market}} &= \log_2(3/1) = 1.584
\end{aligned}
$$

- TF values:

	social	media	mining	data	financial	market
d_1	1	1	1	0	0	0
d_2	1	1	0	1	0	0
d_3	0	0	0	1	1	1

- TF-IDF

	social	media	mining	data	financial	market
d_1	0.584	0.584	1.584	0	0	0
d_2	0.584	0.584	0	0.584	0	0
d_3	0	0	0	0.584	1.584	1.584

Data Quality

When making data ready for mining, data quality needs to be assured

- **Noise**
  - Noise is the distortion of the data
- **Outliers**
  - Outliers are data points that are considerably different from other data points in the dataset
- **Missing Values**
  - Missing feature values in data instances
  - **Solution:**
    - Remove instances that have missing values
    - Estimate missing values, and
    - Ignore missing values when running data mining algorithm
- **Duplicate data**

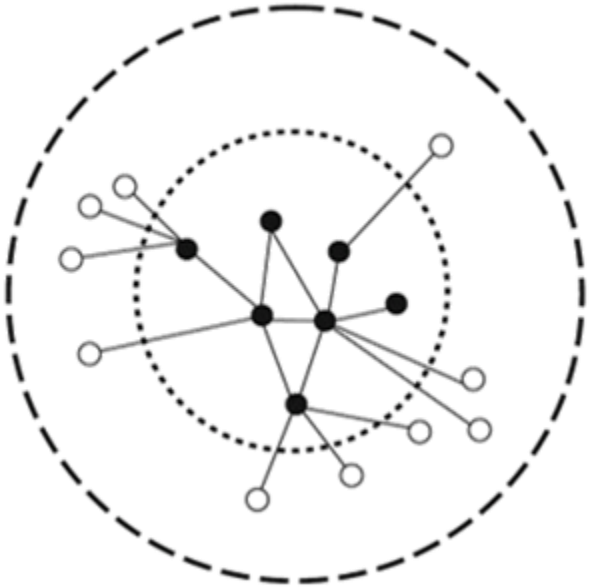
Data Preprocessing

- **Aggregation**
  - It is performed when multiple features need to be combined into a single one or when the scale of the features change
  - Example: image width , image height -> image area (width x height)
- **Discretization**
  - From continues values to discrete values
  - Example: money spent -> {low, normal, high}
- **Feature Selection**
  - Choose relevant features
- **Feature Extraction**
  - Creating new features from original features
  - Often, more complicated than aggregation
- **Sampling**
  - Random Sampling
  - Sampling with or without replacement
  - Stratified Sampling: useful when having class imbalance
  - Social Network Sampling

Data Preprocessing

Sampling social networks:

- Start with a small set of nodes (seed nodes)
- Sample
  - (**a**) the connected components they belong to;
  - (**b**) the set of nodes (and edges) connected to them directly; or
  - (**c**) the set of nodes and edges that are within n-hop distance from them.



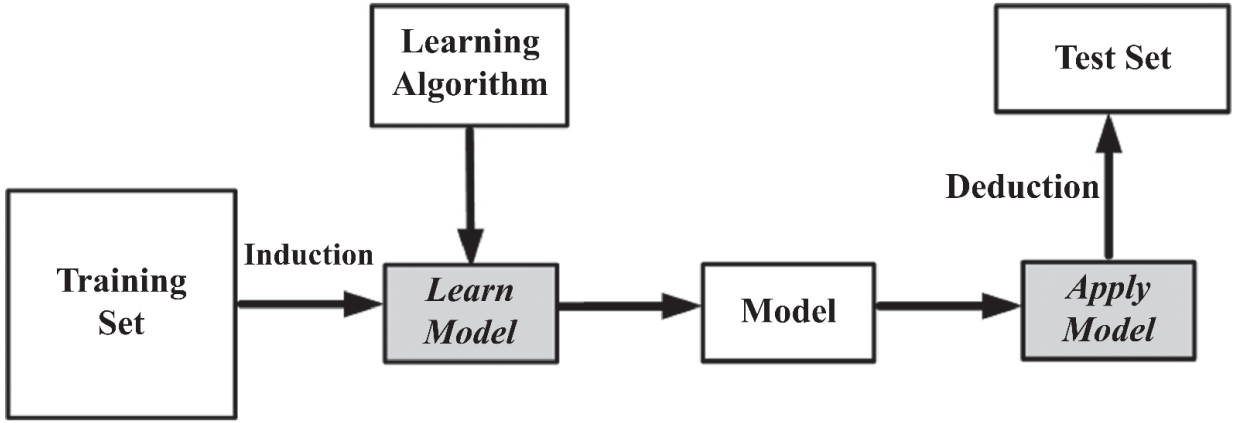
Data Mining Algorithms

Data Mining Algorithms

- **Supervised Learning Algorithm**
  - **Classification (class attribute is discrete)**
    - Assign data into predefined classes
      - Spam Detection
  - **Regression (class attribute takes real values)**
    - Predict a real value for a given data instance
      - Predict the price for a given house
- **Unsupervised Learning Algorithm**
  - Group similar items together into some clusters
    - Detect communities in a given social network

Supervised Learning

Supervised Learning: The Process



- We are given a set of labeled records/instances
  - In the format $(\boldsymbol{X}, y)$
  - $\boldsymbol{X}$ is a vector of features
  - y is the class attribute (commonly a scalar)

- **[Training]** The supervised learning task is to build model that maps $\boldsymbol{X}$ to y (find a mapping m such that $m(\boldsymbol{X}) = y$)

- **[Testing]** Given an unlabeled instance $(\boldsymbol{X}', ?)$, we compute $m(\boldsymbol{X}')$
  - E.g., spam/non-spam prediction

Classification: An Email Example

- A set of emails is given
  - Users have **manually** labeled them as **spam** / **non-spam**
- Use a set of features (**x**) to identify spam/non-spam status of the email (**y**)
  - We can use words in the email
- In this case, classes are

$$y = \{spam, non-spam\}$$

Subject	Mailbox
{Definitely Spam?} Online. The easiest way to get your doctorat	spam_...
{Spam?} Quick loans	Junk
{Definitely Spam?} Train to become a photographer. Find reputab	Junk
{Definitely Spam?} Custom website designing	spam_...
Spam: {Definitely Spam?} Looking to become a nurse?	spam_...
{Spam?} Start your fairy-tale in Orlando.	spam_...
Spam: {Definitely Spam?} Online doctorate programs in your area	spam_...
{Definitely Spam?} Simple. Secure. Mobile. Email Fax	spam_...
Spam: {Definitely Spam?} Online Education Is Easier Than You Th	Junk
{Definitely Spam?} Train to become a photographer. Find reputab	Junk
Spam: {Definitely Spam?} Design Degrees	spam_...
{Spam?} Local maids want to help you	spam_...
{Spam?} Compare beads and save	spam_...
{Spam?} Talking for free has never been so easy	spam_...
{Spam?} Browse our selection of camera phones	spam_...
{Definitely Spam?} Owning a franchise is easy	spam_...
Definitely Spam?} Convenient Online Options to Earn Your Degree	spam_...
Spam: {Definitely Spam?} We are offering free cell phones	spam_...
{Definitely Spam?} Need a replacement carburetor?	Junk
{Definitely Spam?} It's good to have a lawyer on your side	Junk
{Spam?} {Disarmed} Interior Design Schools	spam_...
{Definitely Spam?} Borrow the money you need with a personal lo	spam_...
{Definitely Spam?} Join one of the fastest growing professions.	spam_...
{Definitely Spam?} It's not too late to earn your degree	spam_...
{Definitely Spam?} Subsidize your mortgage with a VA loan	spam_...
Spam: {Definitely Spam?} No butler? Housekeeping services insid	spam_...
Definitely Spam?} It's a good thing to check your credit report	spam_...
Spam: {Definitely Spam?} An offer for software which manages yo	spam_...
{Definitely Spam?} Affordable lawn care	spam_...
{Definitely Spam?} {Disarmed} Driveways, Patios, and Walks. Fin	spam_...
{Definitely Spam?} Cleaning is easier with hardwood floors	spam_...
{Definitely Spam?} Get awesome deals on hammocks.	spam_...
{Definitely Spam?} Affordable beach vacations	spam_...
Spam: {Definitely Spam?} Let your creativity soar	spam_...
{Definitely Spam?} Join the hospital as a medical biller	spam_...
Definitely Spam?} Blow them away with gorgeous diamond jewelry.	spam_...
{Definitely Spam?} Your career in nursing	spam_...
{Definitely Spam?} This is the moment she has been dreaming abo	spam_...
{Definitely Spam?} Medical transcribers work at home	spam_...
{Definitely Spam?} {Disarmed} All types of fence styles availab	spam_...

A Twitter Example

ID	Celebrity	Verified Account	# Followers	Influential?
1	Yes	No	1.25M	No
2	No	Yes	1M	No
3	No	Yes	600K	No
4	Yes	Unknown	2.2M	No
5	No	No	850K	Yes
6	No	Yes	750K	No
7	No	No	900K	Yes
8	No	No	700K	No
9	Yes	Yes	1.2M	No
10	No	Unknown	950K	Yes

Supervised Learning Algorithms

- **Classification**
  - Decision tree learning
  - Naive Bayes Classifier
  - k-nearest neighbor classifier
  - Classification with Network information
- **Regression**
  - Linear Regression
  - Logistic Regression

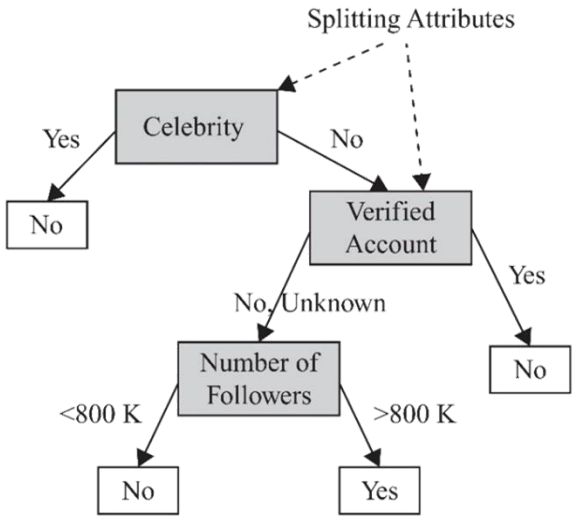
Decision Tree Learning

Decision Tree

- A decision tree is learned from the dataset
  - (training data with known classes)

ID	Celebrity	Verified Account	# Followers	Influential?
1	Yes	No	1.25M	No
2	No	Yes	1M	No
3	No	Yes	600K	No
4	Yes	Unknown	2.2M	No
5	No	No	850K	Yes
6	No	Yes	750K	No
7	No	No	900K	Yes
8	No	No	700K	No
9	Yes	Yes	1.2M	No
10	No	Unknown	950K	Yes

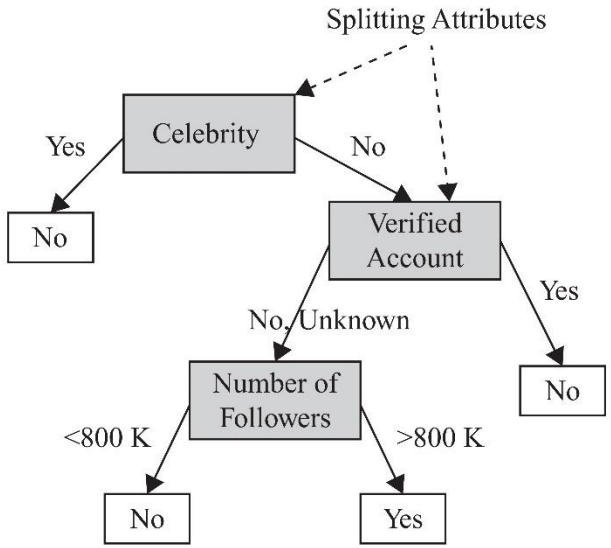
- The learned tree is later applied to predict the class attribute value of new data
  - (test data with unknown classes)
  - Only the feature values are known



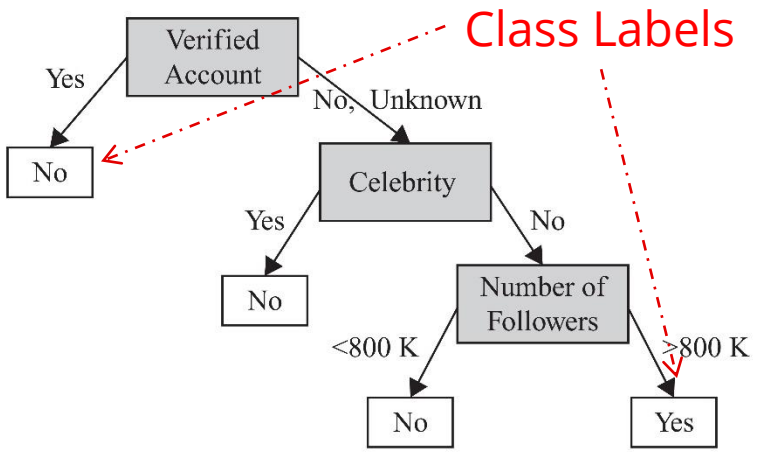
Decision Tree: Example

Multiple decision trees can be learned from the same dataset

ID	Celebrity	Verified Account	# Followers	Influential?
1	Yes	No	1.25M	No
2	No	Yes	1M	No
3	No	Yes	600K	No
4	Yes	Unknown	2.2M	No
5	No	No	850K	Yes
6	No	Yes	750K	No
7	No	No	900K	Yes
8	No	No	700K	No
9	Yes	Yes	1.2M	No
10	No	Unknown	950K	Yes



(a) Learned Decision Tree 1



(b) Learned Decision Tree 2

Decision Tree Construction

- Decision trees are constructed recursively
  - A top-down greedy approach in which features are sequentially selected.
- After selecting a feature for each node,
  - based on its attribute values, different branches are created.
- The training set is then partitioned into subsets based on the feature values,
  - each of which fall under the respective feature value branch;
  - The process is continued for these subsets and other nodes
- When selecting features, we prefer features that partition the set of instances into subsets that are more **pure**.
- A pure subset has instances that all have the same class attribute value.

Decision Tree Construction

- When reaching pure (or highly pure) subsets under a branch,
  - the decision tree construction process no longer partitions the subset,
  - creates a leaf under the branch, and
  - assigns the class attribute value (or the majority class attribute value) for subset instances as the leaf's predicted class attribute value

- To measure purity we can use/minimize entropy.

- Over a subset of training instances, T, with a binary class attribute (values in {+,-}), the entropy of T is defined as:

$$entropy(T) = -p_+ \log p_+ - p_- \log p_-$$

  - p_+ is the proportion of positive examples in T
  - p_- is the proportion of negative examples in T

Entropy Example

Assume there is a subset T that has 10 instances:

- Seven instances have a positive class attribute value
- Three have a negative class attribute value
- Denote T as [7+, 3-]
- The entropy for subset T is

$$entropy(T) = -\frac{7}{10}\log\frac{7}{10} - \frac{3}{10}\log\frac{3}{10} = 0.881$$

In a pure subset, all instances have the same class attribute value (**entropy is 0**)

If the subset contains an unequal number of positive and negative instances

- The entropy is between 0 and 1.

Naïve Bayes Learning

Naive Bayes Classifier

For two random variables $\boldsymbol{X}$ and Y, Bayes theorem states that,

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

class variable — the instance features

Then class attribute value for instance $\boldsymbol{X}$ $\quad \arg\max_{y_i} P(y_i|X)$

We assume that features are independent given the class attribute

$$P(y_i|X) = \frac{P(X|y_i)P(y_i)}{P(X)}$$

$$P(X|y_i) = \Pi_{j=1}^{n} P(x_j|y_i) \Rightarrow P(y_i|X) = \frac{(\Pi_{j=1}^{n} P(x_j|y_i))P(y_i)}{P(X)}$$

NBC: An Example

No.	Outlook (O)	Temperature (T)	Humidity (H)	Play Golf (PG)
1	sunny	hot	high	N
2	sunny	mild	high	N
3	overcast	hot	high	Y
4	rain	mild	high	Y
5	sunny	cool	normal	Y
6	rain	cool	normal	N
7	overcast	cool	normal	Y
8	sunny	mild	high	?

$$
\begin{aligned}
(PG = Y|i_8) &= \frac{P(i_8|PG = Y)P(PG = Y)}{P(i_8)} \\
&= P(O = Sunny, T = mild, H = high|PG = Y) \\
&\quad \times \frac{P(PG = Y)}{P(i_8)} \\
&= P(O = Sunny|PG = Y) \times P(T = mild|PG = Y) \\
&\quad \times P(H = high|PG = Y) \times \frac{P(PG = Y)}{P(i_8)} \\
&= \frac{1}{4} \times \frac{1}{4} \times \frac{2}{4} \times \frac{\frac{4}{7}}{P(i_8)} = \frac{1}{56P(i_8)}.
\end{aligned}
$$

$$
\begin{aligned}
P(PG = N|i_8) &= \frac{P(i_8|PG = N)P(PG = N)}{P(i_8)} \\
&= P(O = Sunny, T = mild, H = high|PG = N) \\
&\quad \times \frac{P(PG = N)}{P(i_8)} \\
&= P(O = Sunny|PG = N) \times P(T = mild|PG = N) \\
&\quad \times P(H = high|PG = N) \times \frac{P(PG = N)}{P(i_8)} \\
&= \frac{2}{3} \times \frac{1}{3} \times \frac{2}{3} \times \frac{\frac{3}{7}}{P(i_8)} = \frac{4}{63P(i_8)}.
\end{aligned}
$$

$$\frac{1}{56P(i_8)} < \frac{4}{63P(i_8)} \Rightarrow Play\ Golf = N$$

Nearest Neighbor Classifier

Nearest Neighbor Classifier

- k-nearest neighbor or k-NN,
  - Utilizes the neighbors of an instance to perform classification.
- It uses the k nearest instances, called **neighbors**, to perform classification.
- The instance being classified is assigned the label (class attribute value) that the majority of its k neighbors are assigned
- When $k = 1$, the closest neighbor's label is used as the predicted label for the instance being classified
- To determine the neighbors of an instance, we need to measure its distance to all other instances based on some distance metric.
  - Often Euclidean distance is employed

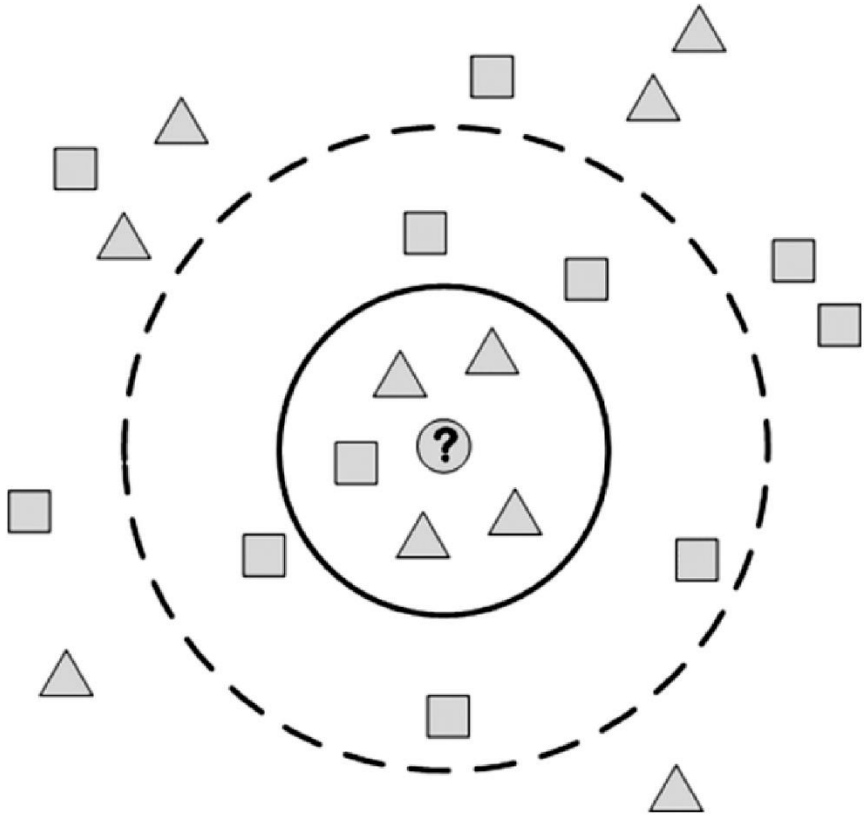
k-NN: Algorithm

Algorithm 5.1 k-Nearest Neighbor Classifier

Require: Instance i, A Dataset of Real-Value Attributes, k (number of neighbors), distance measure d

1: **return** Class label for instance i
2: Compute k nearest neighbors of instance i based on distance mea- sure d.
3: l = the majority class label among neighbors of instance i. If more than one majority label, select one randomly.
4: Classify instance i as class l

k-NN example



- When $k = 5$, the predicted label is: Δ
- When $k = 9$, the predicted label is: □

k-NN: Example 2

No.	Outlook (O)	Temperature (T)	Humidity (H)	Play Golf (PG)
1	sunny	hot	high	N
2	sunny	mild	high	N
3	overcast	hot	high	Y
4	rain	mild	high	Y
5	sunny	cool	normal	Y
6	rain	cool	normal	N
7	overcast	cool	normal	Y
8	sunny	mild	high	?

Similarity between row 8 and other data instances;
(Similarity = 1 if attributes have the same value, otherwise similarity = 0)

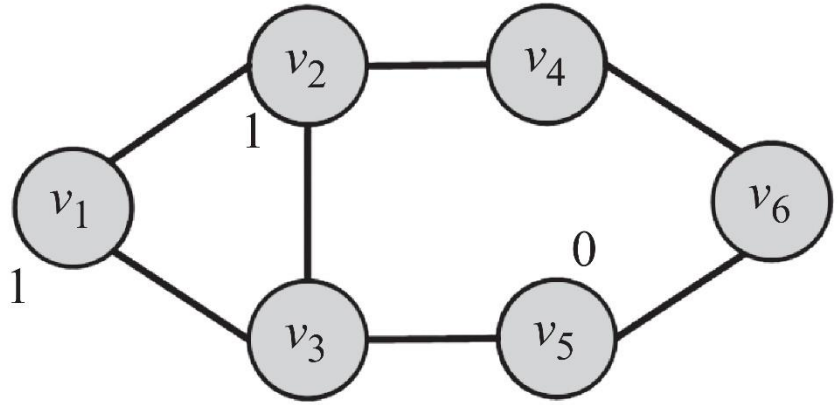
Data instance	Outlook	Temperature	Humidity	Similarity	Label	K	Prediction
2	1	1	1	3	N	1	**N**
1	1	0	1	2	N	2	**N**
4	0	1	1	2	Y	3	**N**
3	0	0	1	1	Y	4	**?**
5	1	0	0	1	Y	5	**Y**
6	0	0	0	0	N	6	**?**
7	0	0	0	0	Y	7	**Y**

Classification with Network Information

Classification with Network Information

- Consider a friendship network on social media and a product being marketed to this network.
- The product seller wants to know who the potential buyers are for this product.
- Assume we are given the network with the list of individuals that decided to buy or not buy the product. Our goal is to predict the decision for the undecided individuals.
- This problem can be formulated as a classification problem based on features gathered from individuals.
- However, in this case, we have additional friendship information that may be helpful in building better classification models

Classification with Network Information



- Let y_i denote the label for node i.
- We can assume that

$$P(y_i = 1) \approx P(y_i = 1|N(v_i))$$

- How can we estimate $\mathrm{P}(y_i = 1|N(v_i))$?

Weighted-vote Relational-Neighbor (wvRN)

wvRN provides an approach to estimate $\mathrm{P}(y_i = 1|N(v_i))$

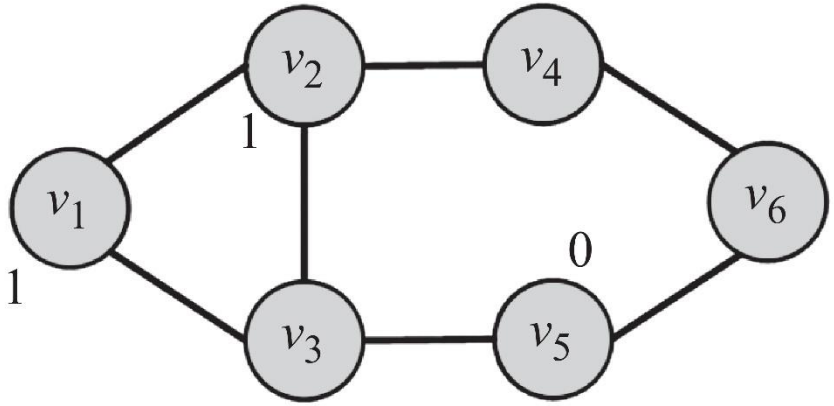
- In wvRN, to find the label of a node, we compute a weighted vote among its neighbors

$$P(y_i = 1|N(v_i)) = \frac{1}{|N(v_i)|} \sum_{v_j \in N(v_i)} P(y_j = 1|N(v_j))$$

- $\mathrm{P}(y_i = 1|N(v_i))$ is only calculated for unlabeled v_i's

- We need to compute these probabilities using **some order** until convergence [i.e., they don't change]
  - What happens for different orders?

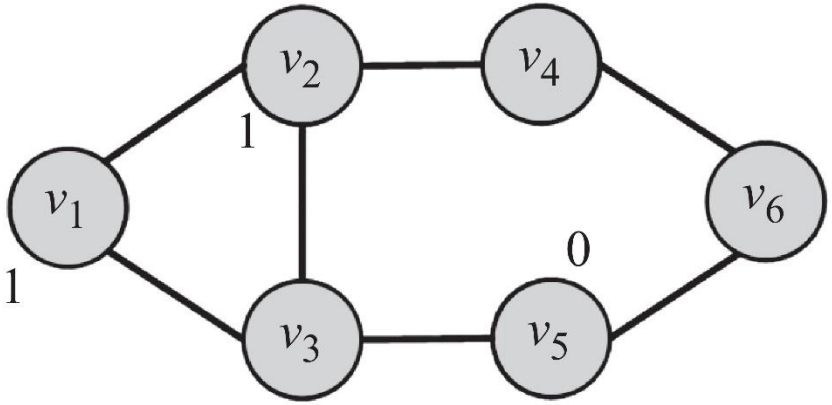
wvRN example



$$P(y_1 = 1|N(v_1)) = 1$$
$$P(y_2 = 1|N(v_2)) = 1$$
$$P(y_5 = 1|N(v_5)) = 0$$

$$P(y_3|N(v_3))$$
$$= \frac{1}{|N(v_3)|} \sum_{v_j \in N(v_3)} P(y_j = 1|N(v_j))$$
$$= \frac{1}{3}(P(y_1 = 1|N(v_1)) + P(y_2 = 1|N(v_2)) + P(y_5 = 1|N(v_5)))$$
$$= \frac{1}{3}(1 + 1 + 0) = 0.67$$
$$P(y_4|N(v_4)) = \frac{1}{2}(1 + 0.5) = 0.75$$
$$P(y_6|N(v_6)) = \frac{1}{2}(0.75 + 0) = 0.38$$

wvRN example



$$P_{(1)}(y_4|N(v_4)) = \frac{1}{2}(1 + 0.38) = 0.69$$

$$P_{(1)}(y_6|N(v_6)) = \frac{1}{2}(0.69 + 0) = 0.35$$

$$P_{(2)}(y_4|N(v_4)) = \frac{1}{2}(1 + 0.35) = 0.68$$

$$P_{(2)}(y_6|N(v_6)) = \frac{1}{2}(0.68 + 0) = 0.34$$

$$P_{(3)}(y_4|N(v_4)) = \frac{1}{2}(1 + 0.34) = 0.67$$

$$P_{(3)}(y_6|N(v_6)) = \frac{1}{2}(0.67 + 0) = 0.34$$

$$P_{(4)}(y_4|N(v_4)) = \frac{1}{2}(1 + 0.34) = 0.67$$

$$P_{(4)}(y_6|N(v_6)) = \frac{1}{2}(0.67 + 0) = 0.34$$

Regression

Regression

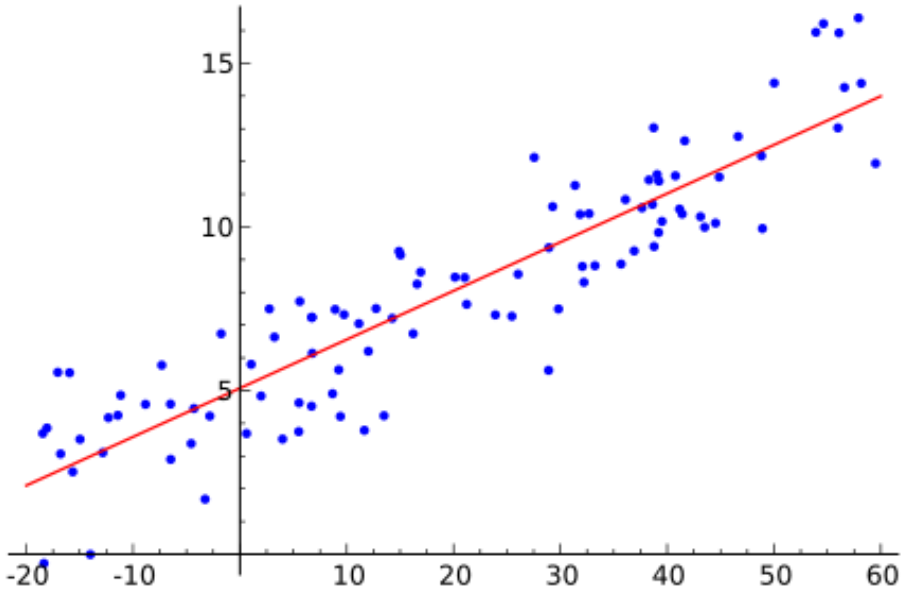
In regression,

- Class values are real numbers as class values
  - In classification, class values are categorical

$$y \approx f(X)$$

Class attribute
(dependent variable)
$y \in R$

Features
(regressors)
$X = (x_1, x_2, \ldots, x_m)$



Goal: find the relation between y and vector $X = (x_1, x_2, \ldots, x_m)$

Linear Regression

- **Linear regression**: we assume the relation between the class attribute Y and feature set $\boldsymbol{X}$ is linear

$$Y = XW + \epsilon$$

- W represents the vector of regression coefficients

- Regression can be solved by estimating W and ε using the provided dataset and the labels Y
  - "**Least squares**" is a popular method to solve regression
  - The goal is to minimize $\epsilon^2 = ||\epsilon^2|| = ||Y - XW||^2$

Least Squares

Find W such that minimizing $\|Y - XW\|^2$ for regressors $\boldsymbol{X}$ and labels Y

$$||X||^2 = X^T X$$

$$min||Y - XW||^2$$

$$\frac{\partial}{\partial W}||Y - XW||^2 = 0$$

$$||X||^2 = X^T X \Rightarrow \frac{\partial}{\partial W}(Y - XW)^T(Y - XW) = 0$$

$$\frac{\partial}{\partial W}(Y^T - W^T X^T)(Y - XW) = 0$$

$$\frac{\partial}{\partial W}(Y^T Y - Y^T XW - W^T X^T Y + W^T X^T XW) = 0$$

$$-2X^T Y + 2X^T XW = 0$$

$$2X^T Y = 2X^T XW$$

$$W = (X^T X)^{-1} X^T Y$$

Simplifying W with SVD

$$X = U\Sigma V^T$$

SVD of X

$$\begin{aligned} W &= (X^T X)^{-1} X^T Y \\ &= (V\Sigma U^T U\Sigma V^T)^{-1} V\Sigma U^T Y \\ &= (V\Sigma^2 V^T)^{-1} V\Sigma U^T Y \\ &= V\Sigma^{-2} V^T V\Sigma U^T Y \\ &= V\Sigma^{-1} U^T Y \end{aligned}$$

Logistic Regression

- A probabilistic view of regression
- Assuming the class attribute is binary, logistic regression finds the probability p

$$P(Y = 1|X) = p$$

X: Feature vector

- We can assume that p can be obtained from X

$$p = \beta X$$

- **Unbounded**: X can take any values and β is also unconstrained

Logistic Regression

- **Solution**: transform p using $g(p)$, such that $g(p)$ is unbounded
  - Fit $g(p)$ using βX

$$g(p) = \ln \frac{p}{1-p}$$

- Known an as the **logit** function
- For any p between [0,1]
  - $G(p)$ is in range $[-\infty, +\infty]$

$$\beta X = \ln \frac{p}{1-p} \rightarrow e^{\beta X} = \frac{p}{1-p} \rightarrow p = \frac{e^{\beta X}}{e^{\beta X}+1}$$

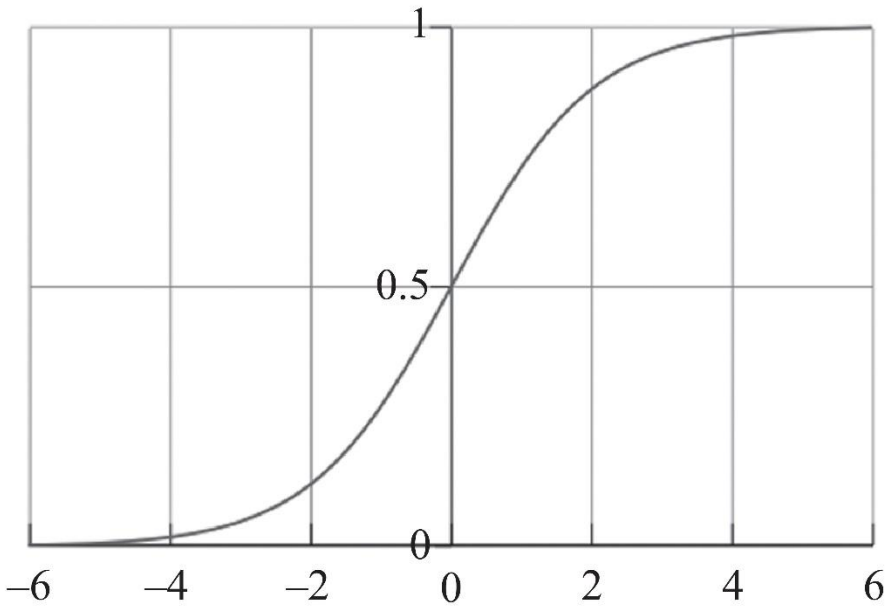
$$p = \frac{1}{e^{-\beta X}+1}$$

Logistic Regression

$$p = \frac{1}{e^{-\beta X} + 1}$$

Logistic Function

- Acts as a probability



- **Goal:** Find β such that $P(Y|X)$ is maximized
  - No closed form solution
  - Iterative maximum likelihood
- Prediction: once β is determined compute $P(Y|X)$
  - For a binary class problem if it is more than 0.5, predict 1

Supervised Learning Evaluation

Evaluating Supervised Learning

- Training/Testing Framework:
  - A training dataset (i.e., the labels are known) is used to train a model
  - the model is evaluated on a test dataset.

- The correct labels of the test dataset are unknown,
  - In practice, the training set is divided into two parts,
    - One used for training and
    - The other used for testing.

- When testing, the labels from this test set are removed.
  - After these labels are predicted using the model, the predicted labels are compared with the masked labels (**ground truth**).

Evaluating Supervised Learning

- Dividing the training set into train/test sets
  - **Leave-one-out training**
    - Divide the training set into k equally sized partitions
      - Often called **folds**
    - Use all folds but one to train and the one left out for testing
  - **k-fold cross validation training**
    - Divide the training set into k equally sized sets
    - Run the algorithm k times
    - In round i, we use all folds but fold i for training and fold i for testing.
    - The average performance of the algorithm over k rounds measures the performance of the algorithm.

Evaluating Supervised Learning

- As the class labels are discrete, we can measure the accuracy by dividing number of correctly predicted labels (C) by the total number of instances (N)

$$\text{accuracy} = \frac{C}{N}$$

$$\text{error rate} = 1 \text{ - accuracy}$$

- More sophisticated approaches of evaluation
  - AUC
  - F-Measure

Evaluating Regression Performance

- The labels cannot be predicted precisely

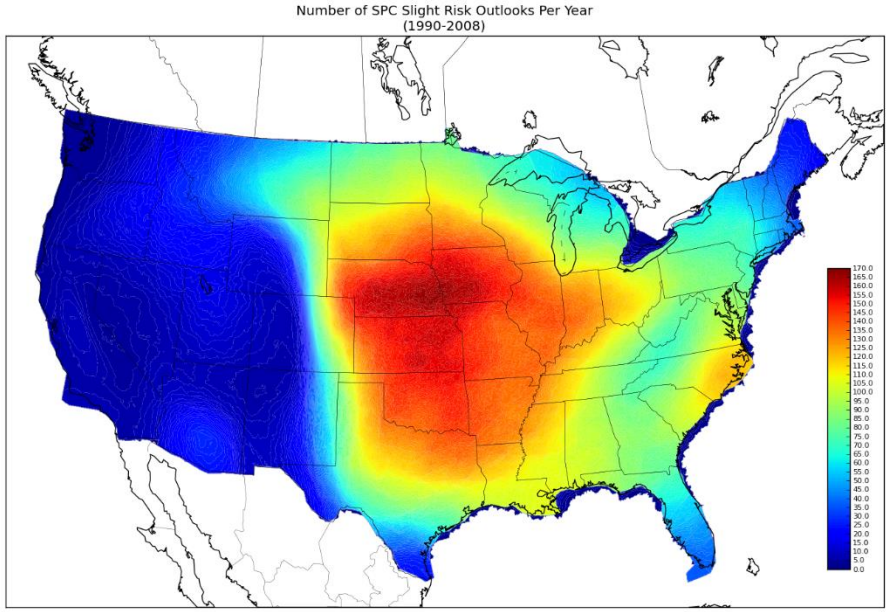
- We need to set a margin to accept or reject the predictions
  - **Example.** When the observed temperature is 71, any prediction in the range of 71±0.5 can be considered as a correct prediction
- Or, we can use correlation between predicted labels and the ground truth.

Unsupervised Learning

Unsupervised Learning

Unsupervised division of instances into groups of similar objects

- Clustering is a form of **unsupervised learning**
- Clustering algorithms group together **similar items**
  - The algorithm does not have examples showing how the samples should be grouped together (unlabeled data)

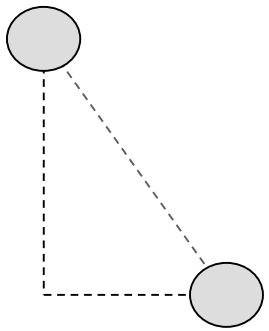


Kernel Density Estimation

Measuring Distance/Similarity in Clustering

- **Clustering Goal:** Group together similar items
- Instances are put into different clusters based on the distance to other instances
- **Any clustering algorithm requires a distance measure**

The most popular (dis)similarity measure for continuous features are ***Euclidean Distance*** and ***Pearson Linear Correlation***



Euclidean Distance

$$d(X,Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Similarity Measures

- X and Y are n-dimensional vectors

$$X = (x_1, x_2, \ldots, x_n)$$
$$Y = (y_1, y_2, \ldots, y_n)$$

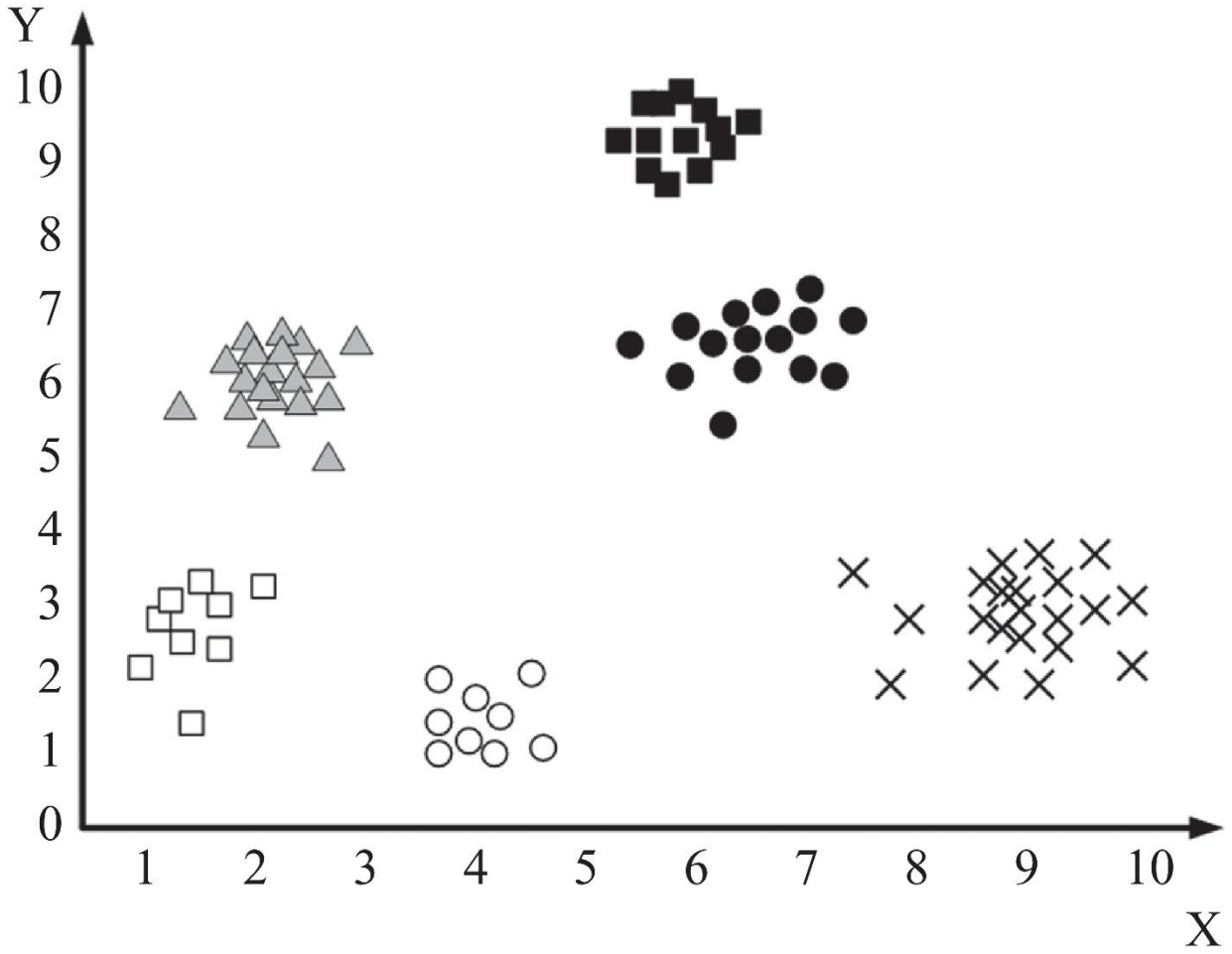
Measure Name	Formula	Description
Mahalanobis	$d(X,Y) = \sqrt{(X-Y)^T \Sigma^{-1} (X-Y)}$	X, Y are features vectors and Σ is the covariance matrix of the dataset
Manhattan (L_1 norm)	$d(X,Y) = \sum_i \lvert x_i - y_i \rvert$	X, Y are features vectors
L_p-norm	$d(X,Y) = (\sum_i \lvert x_i - y_i \rvert^n)^{\frac{1}{n}}$	X, Y are features vectors

Once a distance measure is selected, instances are grouped using it.

Clustering

- Clusters are usually represented by compact and abstract notations.
- "Cluster centroids" are one common example of this abstract notation.
- Partitional Algorithms (most common type)
  - Partition the dataset into a set of clusters
  - Each instance is assigned to a cluster exactly once
  - No instance remains unassigned to clusters.
    - **Example:** k-means

k-means for $k = 6$



k-means

The most commonly used clustering algorithm

- Related to Expectation Maximization (**EM**) in statistics.

Algorithm 5.2 k-Means Algorithm

Require: A Dataset of Real-Value Attributes, k (number of Clusters)

1: **return** A Clustering of Data into k Clusters
2: Consider k random instances in the data space as the initial cluster centroids.
3: **while** centroids have not converged **do**
4: Assign each instance to the cluster that has the closest cluster centroid.
5: If all instances have been assigned then recalculate the cluster centroids by averaging instances inside each cluster
6: **end while**

k-means: Algorithm

Given data points x_i and an initial set of k centroids $m_1^1, m_2^1, \ldots, m_k^1$ the algorithm proceeds as follows:

- **Assignment step**: Assign each data point to the cluster S_i^t with the closest centroid
  - Each data point goes into exactly one cluster

$$S_i^t = \{x_p : ||x_p - m_i^t|| \leq ||x_p - m_j^t|| \forall 1 \leq j \leq k\}$$

- **Update step**: Calculate the new means to be the centroid of the data points in the cluster
  - After all points are assigned

When do we stop?

The procedure is repeated until **convergence**

- **Convergence:**
  - Whether centroids are no longer changing
  - Equivalent to clustering assignments not changing
  - The algorithm can be stopped when the Euclidean distance between the centroids in two consecutive steps is less than some small positive value

k-means (alternative!)

- As an alternative, k-means implementations try to minimize an objective function.
  - **Example:** the squared distance error:

$$\sum_{i=1}^{k} \sum_{j=1}^{n(i)} ||x_j^i - c_i||^2$$

  - x_j^i is the jth instance of cluster i
  - $n(i)$ is the number of instances in cluster i
  - c_i is the centroid of cluster i.

- **Stopping Criterion:**
  - when the difference between the objective function values of two consecutive iterations of the k-means algorithm is less than some small value .

k-means

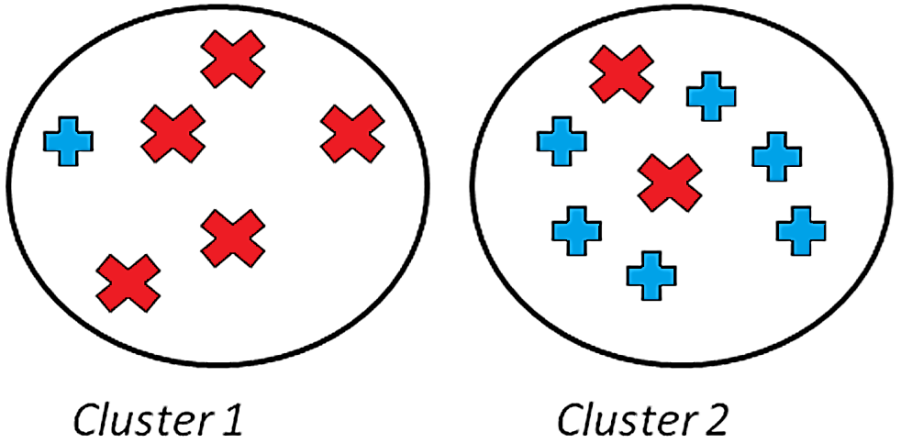
- Finding the global optimum of the k partitions is computationally expensive (**NP**-hard).
- This is equivalent to finding the optimal centroids that minimize the objective function
- **Solution:** efficient heuristics
  - **Outcome:** converge quickly to a local optimum that might not be global
  - **Example:** running k-means multiple times
    - Select the clustering assignment that is observed most often or
    - Select the clustering that is more desirable based on an objective function, such as the squared error.

Unsupervised Learning Evaluation

Evaluating the Clusterings

We are **given** two types of objects

- In perfect clustering, objects of the same type are clustered together.



- Evaluation **with ground truth**
- Evaluation **without ground truth**

Evaluation with Ground Truth

When ground truth is available,

- We have prior knowledge on what the clustering should be (the correct clustering assignments)

- We will discuss these methods in community analysis chapter

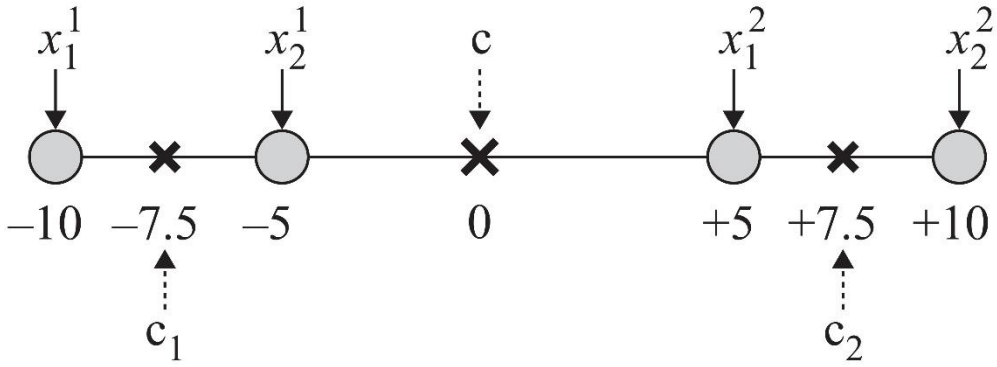
Evaluation without Ground Truth

- **Cohesiveness**
  - In clustering, we are interested in clusters that exhibit cohesiveness
  - In cohesive clusters, instances inside the clusters are close to each other
- **Separateness**
  - We are also interested in clusterings of the data that generates clusters that are well separated from one another

Cohesiveness

- **Cohesiveness**
  - **In statistics**: having a small standard deviation, i.e., being close to the mean value
  - **In clustering**: being close to the centroid of the cluster

$$cohesiveness = \sum_{i=1}^{k} \sum_{j=1}^{n(i)} ||x_j^i - c_i||^2$$

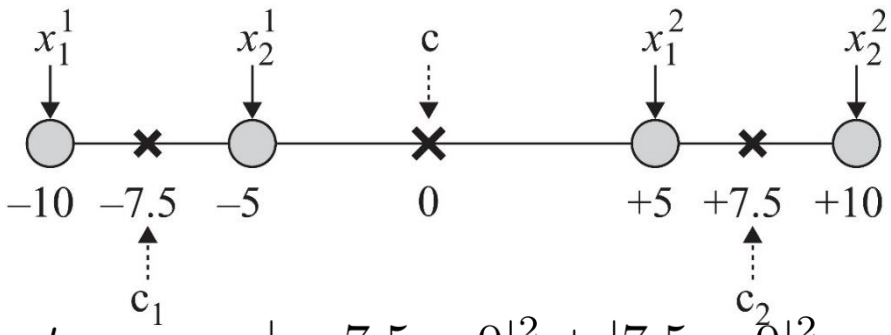


$$cohesiveness = |-10-(-7.5)|^2+|-5-(-7.5)|^2+|5-7.5)|^2+|10-7.5|^2 = 25$$

Separateness

- **Separateness**
  - **In statistics:** separateness can be measured by standard deviation
    - Standard deviation is maximized when instances are far from the mean
  - **In clustering:** cluster centroids being far from the mean of the entire dataset

$$separateness = \sum_{i=1}^{k} ||c - c_i||^2$$



$$separateness = |-7.5-0|^2 + |7.5-0|^2 = 112.5$$

Silhouette Index

- We are interested in clusters that are both **cohesive** and **separate**
  - *Silhouette index*

- It compares
  - *the average distance value between instances in the **same** cluster*

  To
  - *the average distance value between instances in **different** clusters*

- In a well-clustered dataset,
  - the average distance between instances in the same cluster is small (**cohesiveness**) and
  - the average distance between instances in different clusters is large (**separateness**).

Silhouette Index

- For any instance x that is a member of cluster C
- Compute the within-cluster average distance

$$a(x) = \frac{1}{|C|-1} \sum_{y \in C, y \neq x} ||x - y||^2$$

- Compute the average distance between x and instances in cluster G
  - G is closest to x in terms of the average distance between x and members of G

$$b(x) = \min_{G \neq C} \frac{1}{|G|} \sum_{y \in G} ||x - y||^2$$

Silhouette Index

- Our interest: clusterings where $a(x) < b(x)$

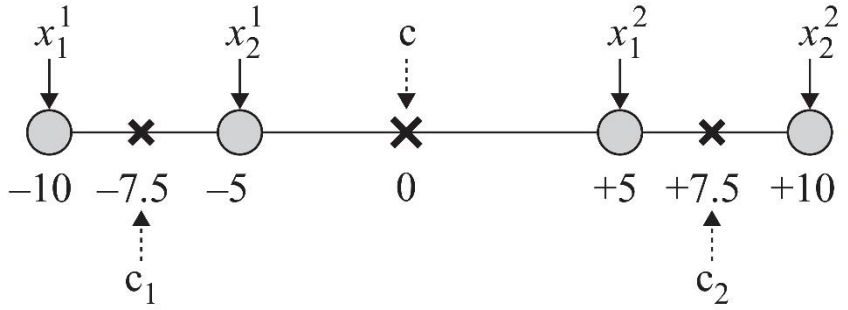
$$s(x) = \frac{b(x) - a(x)}{\max(b(x), a(x))}$$

$$silhouette = \frac{1}{n} \sum_{x} s(x)$$

- Silhouette can take values between $[-1,1]$

- The best case happens when for all x,
  - $a(x) = 0, b(x) > a(x)$

Silhouette Index - Example



$$a(x_1^1) = |-10-(-5)|^2 = 25 \qquad a(x_1^2) = |5-10|^2 = 25$$

$$b(x_1^1) = \frac{1}{2}(|-10-5|^2+|-10-10|^2) = 312.5 \qquad b(x_1^2) = \frac{1}{2}(|5-(-10)|^2+|5-(-5)|^2) = 162.5$$

$$s(x_1^1) = \frac{312.5-25}{312.5} = 0.92 \qquad s(x_1^2) = \frac{162.5-25}{162.5} = 0.84$$

$$a(x_2^1) = |-5-(-10)|^2 = 25 \qquad a(x_2^2) = |10-5|^2 = 25$$

$$b(x_2^1) = \frac{1}{2}(|-5-5|^2+|-5-10|^2) = 162.5 \qquad b(x_2^2) = \frac{1}{2}(|10-(-5)|^2+|10-(-10)|^2) = 312.5$$

$$s(x_2^1) = \frac{162.5-25}{162.5} = 0.84 \qquad s(x_2^2) = \frac{312.5-25}{312.5} = 0.92.$$