Classification

Course Code: CSC 4139 Course Title: Data Warehouse and Data Mining

Dept. of Computer Science
Faculty of Science and Technology

Lecturer No:	**3 & 4**	**Week No:**	**2**	**Semester:**	**Summer 20-21**
Lecturer:	*Dr. Md Mahbub Chowdhury Mishu*				

Lecture Notes Outline

1. Introduction to Classification
2. Naïve Bayes classification
3. Nearest Neighbour Classification

Introduction to Classification

- Lecture Objectives and Outcomes:
  - One of the most common data mining tasks, i.e., classification
  - Two classification algorithms are described in detail: the Naïve Bayes algorithm, which uses probability theory to find the most likely of the possible classifications, and Nearest Neighbour classification, which estimates the classification of an unseen instance using the classification of the instances 'closest' to it.
  - These two methods generally assume that all the attributes are categorical and continuous, respectively.

What is Classification?

- Classification is a task that occurs very frequently in everyday life. Essentially it involves dividing up objects so that each is assigned to one of a number of mutually exhaustive and exclusive categories known as classes. The term ‘mutually exhaustive and exclusive’ simply means that each object must be assigned to precisely one class, i.e. never to more than one and never to no class at all.
- Many practical decision-making tasks can be formulated as classification problems, i.e. assigning people or objects to one of a number of categories, for example:

What is Classification?

- customers who are likely to buy or not buy a particular product in a supermarket
- people who are at high, medium or low risk of acquiring a certain illness
- student projects worthy of a distinction, merit, pass or fail grade
- objects on a radar display which correspond to vehicles, people, buildings or trees
- people who closely resemble, slightly resemble or do not resemble someone seen committing a crime
- houses that are likely to rise in value, fall in value or have an unchanged value in 12 months' time

Introduction to Classification

- If the designated attribute is categorical, the task is called classification.
- Classification is one form of prediction, where the value to be predicted is a label.
- A hospital may want to classify medical patients into those who are at high, medium or low risk
- We may wish to classify a student project as distinction, merit, pass or fail
- Some classifiers deal only with categorical attributes, some only with continuous attributes while some work on both

Naïve Bayes Classifiers

- Lecture Objectives and Outcomes:
  - Familiar with the Naïve Bayes Classifiers
  - Advantages and disadvantages of this classifier
  - Familiar with the Weka tool - open source machine learning software
    - https://www.cs.waikato.ac.nz/ml/weka/

Naïve Bayes Classifiers

- A detailed discussion of probability theory would be substantially outside the scope of this course.
- But you can read specific to Bayes' Theorem: https://en.wikipedia.org/wiki/Bayes%27_theorem
- The *probability* of an *event:*
  - Suppose an event *E* can happen in *r* ways out of a total of *n* possible equally likely ways
  - Then the probability of occurrence of the event (called its success) is denoted by:
  - $P(E) = \frac{r}{n}$

Naïve Bayes Classifiers

- Classifier based on probability theory to find the most probable class. Baye's Theorem:

Bayes' Theorem is named after Thomas Bayes. There are two types of probabilities −

- Posterior Probability [P(H/X)]
- Prior Probability [P(H)]

where X is data tuple and H is some hypothesis.

According to Bayes' Theorem,

$$P(H/X) = P(X/H)P(H) / P(X)$$

- A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes):

$$P(\mathbf{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i)$$

Naïve Bayes Classifiers

- Does not use rules, a decision tree or any other explicit representation of the classifier
- It uses *probability theory* to find the most likely of the possible classifications, when all attributes are categorical as well

Naïve Bayes Classifiers

- We have to predict any one of the four events from the class attribute
- To predict event which is train timing, additional information are needed which are included in the data using additional attributes – *day, season, wind* and *rain*

Naïve Bayes Classifiers

- **Prior Probabilities**
  - *P*(class = on time) = 14/20 = 0.7
  - *P*(class = late) = 2/20 = 0.10
  - *P*(class = very late) = 3/20 = 0.15
  - *P*(class = cancelled) = 1/20 = 0.05

	class = on time	class = late	class = very late	class = cancelled
prior probability	14/20 = 0.7	2/20 = 0.10	3/20 = 0.15	1/20 = 0.05

day	season	wind	rain	class
weekday	spring	none	none	on time
weekday	winter	none	slight	on time
weekday	winter	none	slight	on time
weekday	winter	high	heavy	late
saturday	summer	normal	none	on time
weekday	autumn	normal	none	very late
holiday	summer	high	slight	on time
sunday	summer	normal	none	on time
weekday	winter	high	heavy	very late
weekday	summer	none	slight	on time
saturday	spring	high	heavy	cancelled
weekday	summer	high	slight	on time
saturday	winter	normal	none	late
weekday	summer	high	none	on time
weekday	winter	normal	heavy	very late
saturday	autumn	high	slight	on time
weekday	autumn	none	heavy	on time
holiday	spring	normal	slight	on time
weekday	spring	normal	none	on time
weekday	spring	normal	slight	on time

Naïve Bayes Classifiers

- **Conditional Probabilities**
  - *P*(day = weekday | class = on time) = 9/14 = 0.65

	class = on time	class = late	class = very late	class = cancelled
day = weekday	9/14 = 0.65			

day	season	wind	rain	class
weekday	spring	none	none	on time
weekday	winter	none	slight	on time
weekday	winter	none	slight	on time
weekday	winter	high	heavy	late
saturday	summer	normal	none	on time
weekday	autumn	normal	none	very late
holiday	summer	high	slight	on time
sunday	summer	normal	none	on time
weekday	winter	high	heavy	very late
weekday	summer	none	slight	on time
saturday	spring	high	heavy	cancelled
weekday	summer	high	slight	on time
saturday	winter	normal	none	late
weekday	summer	high	none	on time
weekday	winter	normal	heavy	very late
saturday	autumn	high	slight	on time
weekday	autumn	none	heavy	on time
holiday	spring	normal	slight	on time
weekday	spring	normal	none	on time
weekday	spring	normal	slight	on time

Naïve Bayes Classifiers

- **Conditional Probabilities**
  - *P*(day = weekday | class = on time) = 9/14 = 0.65
  - *P*(day = weekday | class = late) = 1/2 = 0. 5

	class = on time	class = late	class = very late	class = cancelled
day = weekday	9/14 = 0.65	1/2 = 0. 5		

day	season	wind	rain	class
weekday	spring	none	none	on time
weekday	winter	none	slight	on time
weekday	winter	none	slight	on time
weekday	winter	high	heavy	late
saturday	summer	normal	none	on time
weekday	autumn	normal	none	very late
holiday	summer	high	slight	on time
sunday	summer	normal	none	on time
weekday	winter	high	heavy	very late
weekday	summer	none	slight	on time
saturday	spring	high	heavy	cancelled
weekday	summer	high	slight	on time
saturday	winter	normal	none	late
weekday	summer	high	none	on time
weekday	winter	normal	heavy	very late
saturday	autumn	high	slight	on time
weekday	autumn	none	heavy	on time
holiday	spring	normal	slight	on time
weekday	spring	normal	none	on time
weekday	spring	normal	slight	on time

Naïve Bayes Classifiers

- **Conditional Probabilities**
  - *P*(day = weekday | class = on time) = 9/14 = 0.65
  - *P*(day = weekday | class = late) = 1/2 = 0.5
  - *P*(day = weekday | class = very late) = 3/3 = 1
  - *P*(day = weekday | class = cancelled) = 0/1 = 0

	class = on time	class = late	class = very late	class = cancelled
day = weekday	9/14 = 0.65	1/2 = 0. 5	3/3 = 1	0/1 = 0

day	season	wind	rain	class
weekday	spring	none	none	on time
weekday	winter	none	slight	on time
weekday	winter	none	slight	on time
weekday	winter	high	heavy	late
saturday	summer	normal	none	on time
weekday	autumn	normal	none	very late
holiday	summer	high	slight	on time
sunday	summer	normal	none	on time
weekday	winter	high	heavy	very late
weekday	summer	none	slight	on time
saturday	spring	high	heavy	cancelled
weekday	summer	high	slight	on time
saturday	winter	normal	none	late
weekday	summer	high	none	on time
weekday	winter	normal	heavy	very late
saturday	autumn	high	slight	on time
weekday	autumn	none	heavy	on time
holiday	spring	normal	slight	on time
weekday	spring	normal	none	on time
weekday	spring	normal	slight	on time

Naïve Bayes Classifiers

	class = on time	class = late	class = very late	class = cancelled
day = weekday	**9/14 = 0.65**	**1/2 = 0. 5**	**3/3 = 1**	**0/1 = 0**
day = saturday				
day = sunday				
day = holiday				

day	season	wind	rain	class
weekday	spring	none	none	on time
weekday	winter	none	slight	on time
weekday	winter	none	slight	on time
weekday	winter	high	heavy	late
saturday	summer	normal	none	on time
weekday	autumn	normal	none	very late
holiday	summer	high	slight	on time
sunday	summer	normal	none	on time
weekday	winter	high	heavy	very late
weekday	summer	none	slight	on time
saturday	spring	high	heavy	cancelled
weekday	summer	high	slight	on time
saturday	winter	normal	none	late
weekday	summer	high	none	on time
weekday	winter	normal	heavy	very late
saturday	autumn	high	slight	on time
weekday	autumn	none	heavy	on time
holiday	spring	normal	slight	on time
weekday	spring	normal	none	on time
weekday	spring	normal	slight	on time

Naïve Bayes Classifiers

	class = on time	class = late	class = very late	class = cancelled
season = spring	**4/14 = 0.29**	**0/2 = 0**	**0/3 = 0**	**1/1 = 1**
season = summer				
season = autumn				
season = winter				

day	season	wind	rain	class
weekday	spring	none	none	on time
weekday	winter	none	slight	on time
weekday	winter	none	slight	on time
weekday	winter	high	heavy	late
saturday	summer	normal	none	on time
weekday	autumn	normal	none	very late
holiday	summer	high	slight	on time
sunday	summer	normal	none	on time
weekday	winter	high	heavy	very late
weekday	summer	none	slight	on time
saturday	spring	high	heavy	cancelled
weekday	summer	high	slight	on time
saturday	winter	normal	none	late
weekday	summer	high	none	on time
weekday	winter	normal	heavy	very late
saturday	autumn	high	slight	on time
weekday	autumn	none	heavy	on time
holiday	spring	normal	slight	on time
weekday	spring	normal	none	on time
weekday	spring	normal	slight	on time

Naïve Bayes Classifiers

	class = on time	class = late	class = very late	class = cancelled
wind = none	**5/14 = 0.36**	**0/2 = 0**	**0/3 = 0**	**0/1 = 0**
wind = high				
wind = normal				

day	season	wind	rain	class
weekday	spring	none	none	on time
weekday	winter	none	slight	on time
weekday	winter	none	slight	on time
weekday	winter	high	heavy	late
saturday	summer	normal	none	on time
weekday	autumn	normal	none	very late
holiday	summer	high	slight	on time
sunday	summer	normal	none	on time
weekday	winter	high	heavy	very late
weekday	summer	none	slight	on time
saturday	spring	high	heavy	cancelled
weekday	summer	high	slight	on time
saturday	winter	normal	none	late
weekday	summer	high	none	on time
weekday	winter	normal	heavy	very late
saturday	autumn	high	slight	on time
weekday	autumn	none	heavy	on time
holiday	spring	normal	slight	on time
weekday	spring	normal	none	on time
weekday	spring	normal	slight	on time

Naïve Bayes Classifiers

	class = on time	class = late	class = very late	class = cancelled
rain = none	**5/14 = 0.36**	**1/2 = 0.5**	**1/3 = 0.33**	**0/1 = 0**
rain = slight				
rain = heavy				

day	season	wind	rain	class
weekday	spring	none	none	on time
weekday	winter	none	slight	on time
weekday	winter	none	slight	on time
weekday	winter	high	heavy	late
saturday	summer	normal	none	on time
weekday	autumn	normal	none	very late
holiday	summer	high	slight	on time
sunday	summer	normal	none	on time
weekday	winter	high	heavy	very late
weekday	summer	none	slight	on time
saturday	spring	high	heavy	cancelled
weekday	summer	high	slight	on time
saturday	winter	normal	none	late
weekday	summer	high	none	on time
weekday	winter	normal	heavy	very late
saturday	autumn	high	slight	on time
weekday	autumn	none	heavy	on time
holiday	spring	normal	slight	on time
weekday	spring	normal	none	on time
weekday	spring	normal	slight	on time

Naïve Bayes Classifiers

- We have to calculate posterior probability for each *classification* using the formula

$P(c_i) \times P(a_1 = v_1 \mid c_i) \times$
$P(a_2 = v_2 \mid c_i) \times \ldots \times P(a_n = v_n \mid c_i)$

- Class with the highest posterior probability is the decision made by the Naïve Bayes Classification

	Class = on time	Class = late	Class = very late	Class = cancelled
day = weekday	9/14 = 0.65	1/2 = 0. 5	3/3 = 1	0/1 = 0
day = saturday	2/14 = 0.14	1/2 = 0.5	0/3 = 0	1/1 = 1
day = sunday	1/14 = 0.07	0/2 = 0	0/3 = 0	0/1 = 0
day = holiday	2/14 = 0.14	0/2 = 0	0/3 = 0	0/1 = 0
season = spring	4/14 = 0.29	0/2 = 0	0/3 = 0	1/1 = 1
season = summer	6/14 = 0.43	0/2 = 0	0/3 = 0	0/1 = 0
season = autumn	2/14 = 0.14	0/2 = 0	1/3 = 0.33	0/1 = 0
season = winter	2/14 = 0.14	2/2 = 1	2/3 = 0.67	0/1 = 0
wind = none	5/14 = 0.36	0/2 = 0	0/3 = 0	0/1 = 0
wind = high	4/14 = 0.29	1/2 = 0.5	1/3 = 0.33	1/1 = 1
wind = normal	5/14 = 0.36	1/2 = 0.5	2/3 = 0.67	0/1 = 0
rain = none	5/14 = 0.36	1/2 = 0.5	1/3 = 0.33	0/1 = 0
rain = slight	8/14 = 0.57	0/2 = 0	0/3 = 0	0/1 = 0
rain = heavy	1/14 = 0.07	1/2 = 0.5	2/3 = 0.67	1/1 = 1
prior probability	14/20 = 0.7	2/20 = 0.10	3/20 = 0.15	1/20 = 0.05

Naïve Bayes Classifiers

- **class = on time**

P(class = on time) x

P(day = weekday | class = on time) x

P(season = winter | class = on time) x

P(wind = high | class = on time) x

P(rain = heavy | class = on time)

$0.70 \times 0.64 \times 0.14 \times 0.29 \times 0.07 = 0.0013$

	Class = on time	Class = late	Class = very late	Class = cancelled
day = weekday	9/14 = 0.65	1/2 = 0. 5	3/3 = 1	0/1 = 0
day = saturday	2/14 = 0.14	1/2 = 0.5	0/3 = 0	1/1 = 1
day = sunday	1/14 = 0.07	0/2 = 0	0/3 = 0	0/1 = 0
day = holiday	2/14 = 0.14	0/2 = 0	0/3 = 0	0/1 = 0
season = spring	4/14 = 0.29	0/2 = 0	0/3 = 0	1/1 = 1
season = summer	6/14 = 0.43	0/2 = 0	0/3 = 0	0/1 = 0
season = autumn	2/14 = 0.14	0/2 = 0	1/3 = 0.33	0/1 = 0
season = winter	2/14 = 0.14	2/2 = 1	2/3 = 0.67	0/1 = 0
wind = none	5/14 = 0.36	0/2 = 0	0/3 = 0	0/1 = 0
wind = high	4/14 = 0.29	1/2 = 0.5	1/3 = 0.33	1/1 = 1
wind = normal	5/14 = 0.36	1/2 = 0.5	2/3 = 0.67	0/1 = 0
rain = none	5/14 = 0.36	1/2 = 0.5	1/3 = 0.33	0/1 = 0
rain = slight	8/14 = 0.57	0/2 = 0	0/3 = 0	0/1 = 0
rain = heavy	1/14 = 0.07	1/2 = 0.5	2/3 = 0.67	1/1 = 1
prior probability	14/20 = 0.7	2/20 = 0.10	3/20 = 0.15	1/20 = 0.05

weekday	winter	high	heavy	????

Naïve Bayes Classifiers

- **class = late**

$0.10 \times 0.50 \times 1.00 \times 0.50 \times 0.50 = 0.0125$

	Class = on time	Class = late	Class = very late	Class = cancelled
day = weekday	9/14 = 0.65	1/2 = 0. 5	3/3 = 1	0/1 = 0
day = saturday	2/14 = 0.14	1/2 = 0.5	0/3 = 0	1/1 = 1
day = sunday	1/14 = 0.07	0/2 = 0	0/3 = 0	0/1 = 0
day = holiday	2/14 = 0.14	0/2 = 0	0/3 = 0	0/1 = 0
season = spring	4/14 = 0.29	0/2 = 0	0/3 = 0	1/1 = 1
season = summer	6/14 = 0.43	0/2 = 0	0/3 = 0	0/1 = 0
season = autumn	2/14 = 0.14	0/2 = 0	1/3 = 0.33	0/1 = 0
season = winter	2/14 = 0.14	2/2 = 1	2/3 = 0.67	0/1 = 0
wind = none	5/14 = 0.36	0/2 = 0	0/3 = 0	0/1 = 0
wind = high	4/14 = 0.29	1/2 = 0.5	1/3 = 0.33	1/1 = 1
wind = normal	5/14 = 0.36	1/2 = 0.5	2/3 = 0.67	0/1 = 0
rain = none	5/14 = 0.36	1/2 = 0.5	1/3 = 0.33	0/1 = 0
rain = slight	8/14 = 0.57	0/2 = 0	0/3 = 0	0/1 = 0
rain = heavy	1/14 = 0.07	1/2 = 0.5	2/3 = 0.67	1/1 = 1
prior probability	14/20 = 0.7	2/20 = 0.10	3/20 = 0.15	1/20 = 0.05

weekday	winter	high	heavy	????

Naïve Bayes Classifiers

- **class = very late**

$0.15 \times 1.00 \times 0.67 \times 0.33 \times 0.67 = 0.0222$

	Class = on time	Class = late	Class = very late	Class = cancelled
day = weekday	9/14 = 0.65	1/2 = 0. 5	3/3 = 1	0/1 = 0
day = saturday	2/14 = 0.14	1/2 = 0.5	0/3 = 0	1/1 = 1
day = sunday	1/14 = 0.07	0/2 = 0	0/3 = 0	0/1 = 0
day = holiday	2/14 = 0.14	0/2 = 0	0/3 = 0	0/1 = 0
season = spring	4/14 = 0.29	0/2 = 0	0/3 = 0	1/1 = 1
season = summer	6/14 = 0.43	0/2 = 0	0/3 = 0	0/1 = 0
season = autumn	2/14 = 0.14	0/2 = 0	1/3 = 0.33	0/1 = 0
season = winter	2/14 = 0.14	2/2 = 1	2/3 = 0.67	0/1 = 0
wind = none	5/14 = 0.36	0/2 = 0	0/3 = 0	0/1 = 0
wind = high	4/14 = 0.29	1/2 = 0.5	1/3 = 0.33	1/1 = 1
wind = normal	5/14 = 0.36	1/2 = 0.5	2/3 = 0.67	0/1 = 0
rain = none	5/14 = 0.36	1/2 = 0.5	1/3 = 0.33	0/1 = 0
rain = slight	8/14 = 0.57	0/2 = 0	0/3 = 0	0/1 = 0
rain = heavy	1/14 = 0.07	1/2 = 0.5	2/3 = 0.67	1/1 = 1
prior probability	14/20 = 0.7	2/20 = 0.10	3/20 = 0.15	1/20 = 0.05

weekday	winter	high	heavy	????

Naïve Bayes Classifiers

- **class = cancelled**

$0.05 \times 0.00 \times 0.00 \times 1.00 \times 1.00 = 0.0000$

	Class = on time	Class = late	Class = very late	Class = cancelled
day = weekday	9/14 = 0.65	1/2 = 0. 5	3/3 = 1	0/1 = 0
day = saturday	2/14 = 0.14	1/2 = 0.5	0/3 = 0	1/1 = 1
day = sunday	1/14 = 0.07	0/2 = 0	0/3 = 0	0/1 = 0
day = holiday	2/14 = 0.14	0/2 = 0	0/3 = 0	0/1 = 0
season = spring	4/14 = 0.29	0/2 = 0	0/3 = 0	1/1 = 1
season = summer	6/14 = 0.43	0/2 = 0	0/3 = 0	0/1 = 0
season = autumn	2/14 = 0.14	0/2 = 0	1/3 = 0.33	0/1 = 0
season = winter	2/14 = 0.14	2/2 = 1	2/3 = 0.67	0/1 = 0
wind = none	5/14 = 0.36	0/2 = 0	0/3 = 0	0/1 = 0
wind = high	4/14 = 0.29	1/2 = 0.5	1/3 = 0.33	1/1 = 1
wind = normal	5/14 = 0.36	1/2 = 0.5	2/3 = 0.67	0/1 = 0
rain = none	5/14 = 0.36	1/2 = 0.5	1/3 = 0.33	0/1 = 0
rain = slight	8/14 = 0.57	0/2 = 0	0/3 = 0	0/1 = 0
rain = heavy	1/14 = 0.07	1/2 = 0.5	2/3 = 0.67	1/1 = 1
prior probability	14/20 = 0.7	2/20 = 0.10	3/20 = 0.15	1/20 = 0.05

weekday	winter	high	heavy	????

Naïve Bayes Classifiers

- class = on time

$0.70 \times 0.64 \times 0.14 \times 0.29 \times 0.07 = 0.0013$

- class = late

$0.10 \times 0.50 \times 1.00 \times 0.50 \times 0.50 = 0.0125$

- class = very late

$0.15 \times 1.00 \times 0.67 \times 0.33 \times 0.67 = 0.0222$

- class = cancelled

$0.05 \times 0.00 \times 0.00 \times 1.00 \times 1.00 = 0.0000$

- The largest value (0.0222) is for

class = very late

	Class = on time	Class = late	Class = very late	Class = cancelled
day = weekday	9/14 = 0.65	1/2 = 0. 5	3/3 = 1	0/1 = 0
day = saturday	2/14 = 0.14	1/2 = 0.5	0/3 = 0	1/1 = 1
day = sunday	1/14 = 0.07	0/2 = 0	0/3 = 0	0/1 = 0
day = holiday	2/14 = 0.14	0/2 = 0	0/3 = 0	0/1 = 0
season = spring	4/14 = 0.29	0/2 = 0	0/3 = 0	1/1 = 1
season = summer	6/14 = 0.43	0/2 = 0	0/3 = 0	0/1 = 0
season = autumn	2/14 = 0.14	0/2 = 0	1/3 = 0.33	0/1 = 0
season = winter	2/14 = 0.14	2/2 = 1	2/3 = 0.67	0/1 = 0
wind = none	5/14 = 0.36	0/2 = 0	0/3 = 0	0/1 = 0
wind = high	4/14 = 0.29	1/2 = 0.5	1/3 = 0.33	1/1 = 1
wind = normal	5/14 = 0.36	1/2 = 0.5	2/3 = 0.67	0/1 = 0
rain = none	5/14 = 0.36	1/2 = 0.5	1/3 = 0.33	0/1 = 0
rain = slight	8/14 = 0.57	0/2 = 0	0/3 = 0	0/1 = 0
rain = heavy	1/14 = 0.07	1/2 = 0.5	2/3 = 0.67	1/1 = 1
prior probability	14/20 = 0.7	2/20 = 0.10	3/20 = 0.15	1/20 = 0.05

Classroom Exercise-1

1. Using the Naïve Bayes classification algorithm with the *train* dataset, calculate the most likely classification for the following unseen instances.

weekday	summer	high	heavy	????
sunday	summer	normal	slight	????

Avoiding the Zero-Probability Problem

- Naïve Bayesian prediction requires each conditional prob. be **non-zero**. Otherwise, the predicted prob. will be zero

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$

Example: Suppose a dataset with 1000 tuples, income=low (0), income= medium (990), and income = high (10)

- Use Laplacian correction (or Laplacian estimator)
  - *Adding 1 to each case*
    - Prob(income = low) = 1/1003
    - Prob(income = medium) = 991/1003
    - Prob(income = high) = 11/1003
  - The "corrected" prob. estimates are close to their "uncorrected" counterparts

Naïve Bayes Classifier

- Advantages
  - Easy to implement
  - Good results obtained in most of the cases

Naïve Bayes Classifiers

- Disadvantages
  - It relies on all attributes being categorical
    - Datasets have a combination of categorical and continuous attributes, or even only continuous attributes
    - We can Convert the continuous attributes to categorical ones
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., hospitals: patients: Profile: age, family history, etc.
      - Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayes Classifier
- How to deal with these dependencies?
  - Bayesian Belief Networks

Assignment/Exercise

1. Apply the Naïve Bayes classification algorithm.

SoftEng	ARIN	HCI	CSA	Project	Class
A	B	A	B	B	SECOND
A	B	B	B	A	FIRST
A	A	A	B	B	SECOND
B	A	A	B	B	SECOND
A	A	B	B	A	FIRST
B	A	A	B	B	SECOND
A	B	B	B	B	SECOND
A	B	B	B	B	SECOND
A	A	A	A	A	FIRST
B	A	A	B	B	SECOND
B	A	A	B	B	SECOND
A	B	B	A	B	SECOND
B	B	B	B	A	SECOND
A	A	B	A	B	FIRST
B	B	B	B	A	SECOND
A	A	B	B	B	SECOND
B	B	B	B	B	SECOND
A	A	B	A	A	FIRST
B	B	B	A	A	SECOND
B	B	A	A	B	SECOND
B	B	B	B	A	SECOND
B	A	B	A	B	SECOND
A	B	B	B	A	FIRST
A	B	A	B	B	SECOND
B	A	B	B	B	SECOND
A	B	B	B	B	SECOND

Classes
FIRST, SECOND
SoftEng
A,B
ARIN
A,B
HCI
A,B
CSA
A,B
Project
A,B

Assignment/Exercise

- Building and evaluating Naive Bayes classifier with **WEKA**
  1. Apply the Naïve Bayes classifier to the *train* dataset (Principles of Data Mining – Max Bramer).
     1. Prepare data for classification
     2. Building a Naive Bayes model
     3. Evaluate classifier with the test set
  2. Apply the Naïve Bayes classifier to the *degrees* dataset (Principles of Data Mining – Max Bramer).

- Apply the Naïve Bayes classification algorithm with the above dataset (problem 1 and 2 above) dataset, calculate the most likely classification.

Nearest Neighbour Classification

- Lecture Objectives and Outcomes:
  - Familiar with the Nearest Neighbour Classification Algorithm
  - Advantages and disadvantages of this classifier
  - Familiar with the Weka tool - open source machine learning software
    - https://www.cs.waikato.ac.nz/ml/weka/

Nearest Neighbour Classification

- Mainly used when all attribute values are continuous
- It can be modified to deal with categorical attributes
- The idea is to estimate the classification of an unseen instance using the classification of the instance or instances that are *closest* to it, in some sense that we need to define (classifies new cases based on a similarity measure)

Nearest Neighbour Classification

a	b	c	d	e	f	Class
yes	no	no	6.4	8.3	low	negative
yes	yes	yes	18.2	4.7	high	positive

yes	no	no	6.6	8.0	low	???

- What should its classification be?
- Even without knowing what the six attributes represent, it seems intuitively obvious that the unseen instance is *nearer* to the first instance than to the second.

K - Nearest Neighbour (KNN)

- In practice there are likely to be many more instances in the training set but the same principle applies.
- It is usual to base the classification on those of the *k* nearest neighbours, not just the nearest one.
- The method is then known as *k-Nearest Neighbour* or just *k-NN classification*

Basic k-Nearest Neighbour Classification Algorithm

– Find the k training instances that are closest to the unseen instance.

– Take the most commonly occurring classification for these k instances.

KNN

- We can illustrate *k-NN* classification diagrammatically when the *dimension* (i.e. the number of attributes) is small.
- Next, we will see an example which illustrates the case where the dimension is just 2.
- In real-world data mining applications it can of course be considerably larger.

KNN

- A training set with 20 instances, each giving the values of two attributes and an associated classification
- How can we estimate the classification for an 'unseen' instance where the first and second attributes are 9.1 and 11.0, respectively?

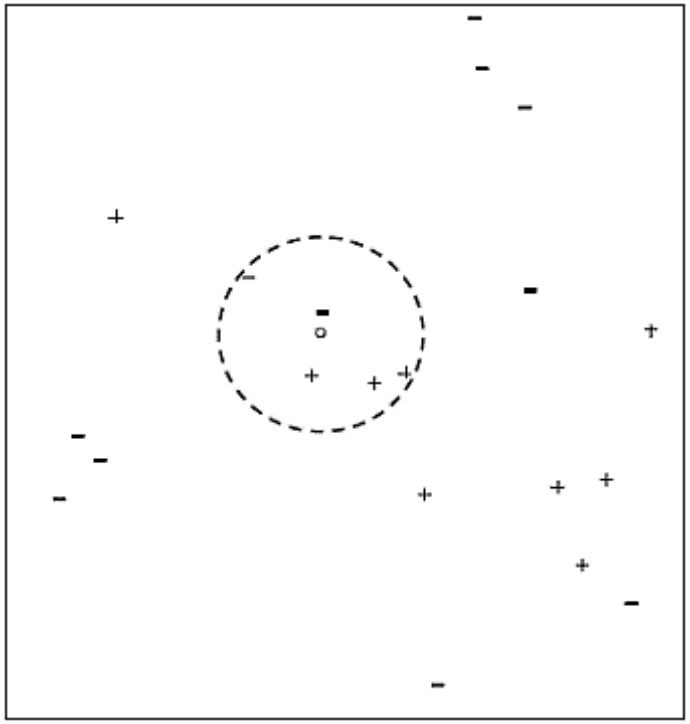
Attribute 1	Attribute 2	Class
0.8	6.3	−
1.4	8.1	−
2.1	7.4	−
2.6	14.3	+
6.8	12.6	−
8.8	9.8	+
9.2	11.6	−
10.8	9.6	+
11.8	9.9	+
12.4	6.5	+
12.8	1.1	−
14.0	19.9	−
14.2	18.5	−
15.6	17.4	−
15.8	12.2	−
16.6	6.7	+
17.4	4.5	+
18.2	6.9	+
19.0	3.4	−
19.6	11.1	+

KNN

- For this small number of attributes we can represent the training set as 20 points on a two-dimensional graph with values of the first and second attributes measured along the horizontal and vertical axes, respectively.
- Each point is labelled with a + or – symbol to indicate that the classification is positive or negative, respectively.

Attribute 1	Attribute 2	Class
0.8	6.3	–
1.4	8.1	–
2.1	7.4	–
2.6	14.3	+
6.8	12.6	–
8.8	9.8	+
9.2	11.6	–
10.8	9.6	+
11.8	9.9	+
12.4	6.5	+
12.8	1.1	–
14.0	19.9	–
14.2	18.5	–
15.6	17.4	–
15.8	12.2	–
16.6	6.7	+
17.4	4.5	+
18.2	6.9	+
19.0	3.4	–
19.6	11.1	+

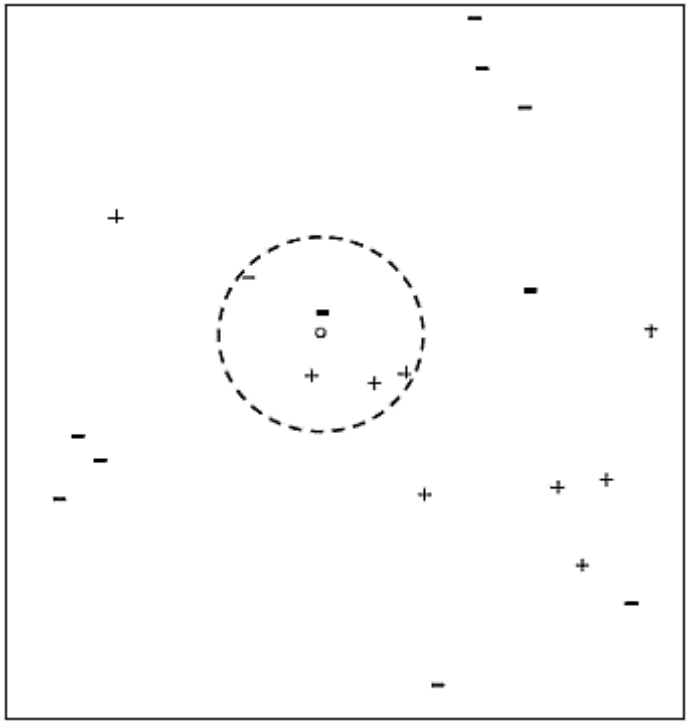
KNN



Attribute 1	Attribute 2	Class
0.8	6.3	−
1.4	8.1	−
2.1	7.4	−
2.6	14.3	+
6.8	12.6	−
8.8	9.8	+
9.2	11.6	−
10.8	9.6	+
11.8	9.9	+
12.4	6.5	+
12.8	1.1	−
14.0	19.9	−
14.2	18.5	−
15.6	17.4	−
15.8	12.2	−
16.6	6.7	+
17.4	4.5	+
18.2	6.9	+
19.0	3.4	−
19.6	11.1	+

- A circle has been added to enclose the five nearest neighbours of the unseen instance, which is shown as a small circle close to the centre of the larger one.

KNN



Attribute 1	Attribute 2	Class
0.8	6.3	−
1.4	8.1	−
2.1	7.4	−
2.6	14.3	+
6.8	12.6	−
8.8	9.8	+
9.2	11.6	−
10.8	9.6	+
11.8	9.9	+
12.4	6.5	+
12.8	1.1	−
14.0	19.9	−
14.2	18.5	−
15.6	17.4	−
15.8	12.2	−
16.6	6.7	+
17.4	4.5	+
18.2	6.9	+
19.0	3.4	−
19.6	11.1	+

- The five nearest neighbours are labelled with three + signs and two − signs
- So a basic *5-NN* classifier would classify the unseen instance as ‘positive’ by a form of majority voting.

KNN

- We can represent two points in two dimensions ('in two-dimensional space' is the usual term) as ($a1$, $a2$) and ($b1$, $b2$)
- When there are three attributes, we can represent the points by ($a1$, $a2$, $a3$) and ($b1$, $b2$, $b3$)
- When there are n attributes, we can represent the instances by the points ($a1$, $a2$, . . . , an) and ($b1$, $b2$, . . . , bn) in 'n-dimensional space'

Distance Measures

- There are many possible ways of measuring the distance between two instances with *n* attribute values, or equivalently between two points in *n*-dimensional space.

- But here distance measurement usually imposes three requirements (let, **dist**(*X, Y*) denotes the distance between two points *X* and *Y*)
  - The distance of any point *A* from itself is zero, i.e. **dist**(*A, A*) = 0
  - The distance from *A* to *B* is the same as the distance from *B* to *A*, i.e.
    - **dist**(*A, B*) = **dist**(*B, A*) (the *symmetry condition*)
  - The third condition is called the *triangle inequality* (Figure 2.7). It corresponds to the intuitive idea that 'the shortest distance between any two points is a straight line'. The condition says that for any points *A*, *B* and *Z*:
    - **dist**(*A, B*) ≤ **dist**(*A, Z*) + **dist**(*Z, B*)

Distance Measures

- There are many possible distance measures
  - Euclidean Distance:

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

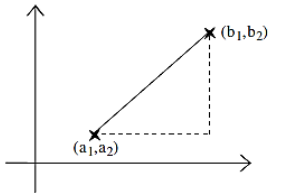


Figure 2.8 Example of Euclidean Distance

  - Manhattan Distance or City Block Distance:

$$\sum_{i=1}^{k}|x_i - y_i|$$

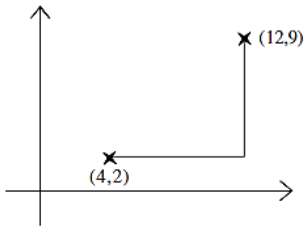


Figure 2.9 Example of City Block Distance

  - Hamming Distance:

$$D_H = \sum_{i=1}^{k}|x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

Distance Measures: Euclidean Distance

- If we denote an instance in the training set by *(a1, a2)* and the unseen instance by *(b1, b2)* the length of the straight line joining the points is $\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$
- If there are two points *(a1, a2, a3)* and *(b1, b2, b3)* in a three-dimensional space the corresponding formula is

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2}$$

- The formula for Euclidean distance between points *(a1, a2, . . . , an)* and *(b1, b2, . . . , bn)* in n-dimensional space is a generalisation of these two results. The Euclidean distance is given by the formula

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \ldots + (a_n - b_n)^2}$$

Distance Measures: Manhattan Distance

- The City Block distance between the points (4, 2) and (12, 9) is $(12 - 4) + (9 - 2) = 8 + 7 = 15$

Example/Assignment of KNN

- A training set with 20 instances, each giving the values of two attributes and an associated classification
- How can we estimate the classification for an 'unseen' instance where the first and second attributes are 9.1 and 11.0, respectively, when K=5 and K=3
- Use Euclidean Distance and other distance measure methods and compare the results.
- You can use WEKA, Python, or any other programming language that you find most suitable for you.
- You can also try Excel as well for this problem.

Attribute 1	Attribute 2	Class
0.8	6.3	−
1.4	8.1	−
2.1	7.4	−
2.6	14.3	+
6.8	12.6	−
8.8	9.8	+
9.2	11.6	−
10.8	9.6	+
11.8	9.9	+
12.4	6.5	+
12.8	1.1	−
14.0	19.9	−
14.2	18.5	−
15.6	17.4	−
15.8	12.2	−
16.6	6.7	+
17.4	4.5	+
18.2	6.9	+
19.0	3.4	−
19.6	11.1	+

Example/Assignment of KNN-Solution

- Estimate the classification for an ‘unseen’ instance using Euclidean distance, where the first and second attributes are 10 and 15, respectively, when K=5.

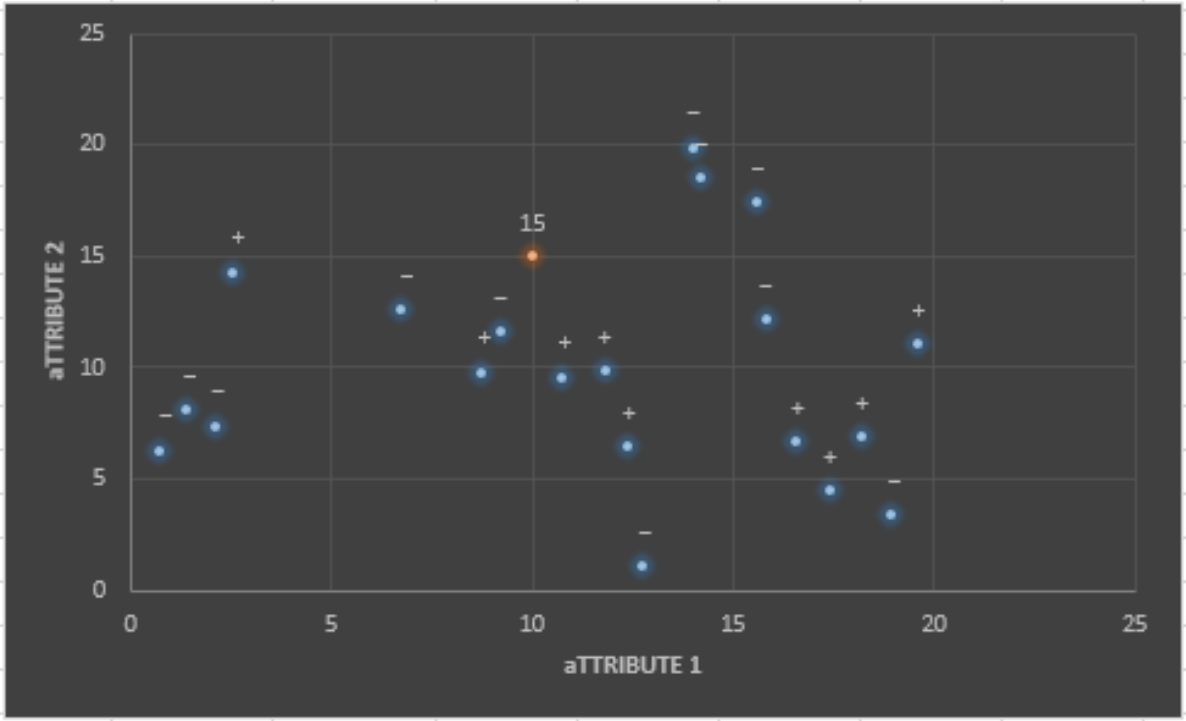
Attribute 1	Aattribute 2	Class (+/-)	Distance
0.8	6.3	–	12.6621
1.4	8.1	–	11.0259
2.1	7.4	–	10.9622
2.6	14.3	+	7.43303
6.8	12.6	–	4
8.8	9.8	+	5.33667
9.2	11.6	–	3.49285
10.8	9.6	+	5.45894
11.8	9.9	+	5.40833
12.4	6.5	+	8.83233
12.8	1.1	–	14.1792
14	19.9	–	6.32535
14.2	18.5	–	5.46717
15.6	17.4	–	6.09262
15.8	12.2	–	6.4405
16.6	6.7	+	10.6042
17.4	4.5	+	12.8456
18.2	6.9	+	11.5261
19	3.4	–	14.682
19.6	11.1	+	10.3619

Example/Assignment of KNN-Solution

Attribute 1	Attribute 2	Class
0.8	6.3	−
1.4	8.1	−
2.1	7.4	−
2.6	14.3	+
6.8	12.6	−
8.8	9.8	+
9.2	11.6	−
10.8	9.6	+
11.8	9.9	+
12.4	6.5	+
12.8	1.1	−
14.0	19.9	−
14.2	18.5	−
15.6	17.4	−
15.8	12.2	−
16.6	6.7	+
17.4	4.5	+
18.2	6.9	+
19.0	3.4	−
19.6	11.1	+

Test Data

Attribute 1	Attribute 2	KNN	Class
10	15	5	+



Normalisation

- A major problem when using the Euclidean distance formula (and many other distance measures) is that the large values frequently swamp the small ones.

Mileage (miles)	Number of doors	Age (years)	Number of owners
18,457	2	12	8
26,292	4	3	1

- When the distance of these instances from an unseen one is calculated, the *mileage* attribute will almost certainly contribute a value of several thousands squared, i.e. several millions, to the sum of squares total.

Normalisation

- It is clear that in practice the only attribute that will matter when deciding which neighbours are the nearest using the Euclidean distance formula is the mileage.
- We could have chosen an alternative measure of distance travelled such as millimetres or perhaps light years. Similarly we might have measured age in some other unit such as milliseconds or millennia. The units chosen should not affect the decision on which are the nearest neighbours.

Mileage (miles)	Number of doors	Age (years)	Number of owners
18,457	2	12	8
26,292	4	3	1

Normalisation

- To overcome this problem we generally *normalise* the values of continuous attributes.
- The idea is to make the values of each attribute run from 0 to 1.
- In general if the lowest value of attribute *A* is *min* and the highest value is *max*, we convert each value of *A*,
  say *a*, to $(a - min)/(max - min)$.
- Using this approach all continuous attributes are converted to small numbers from 0 to 1, so the effect of the choice of unit of measurement on the outcome is greatly reduced.

Normalisation

- Note that it is possible that an unseen instance may have a value of *A* that is less than *min* or greater than *max*. If we want to keep the adjusted numbers 38 Principles of Data Mining in the range from 0 to 1 we can just convert any values of *A* that are less than *min* or greater than *max* to 0 or 1, respectively.

Normalisation

- Another issue that occurs with measuring the distance between two points is the *weighting* of the contributions of the different attributes.
- We may believe that the mileage of a car is more important than the number of doors it has.
- To achieve this we can adjust the formula for Euclidean distance to

$$\sqrt{w_1(a_1 - b_1)^2 + w_2(a_2 - b_2)^2 + \ldots + w_n(a_n - b_n)^2}$$

where w_1, w_2, . . . , w_n are the weights. It is customary to scale the weight values so that the sum of all the weights is one.

Dealing with Categorical Attributes

- One of the weaknesses of the nearest neighbour approach to classification is that there is no entirely satisfactory way of dealing with categorical attributes.
- One possibility is to say that the difference between any two identical values of the attribute is zero and that the difference between any two different values is 1. (Hamming Distance)
- Effectively this amounts to saying (for a colour attribute) red − red = 0, red − blue = 1, blue − green = 1, etc.

Dealing with Categorical Attributes

- Sometimes there is an ordering (or a partial ordering) of the values of an attribute (Ordinal Attribute), for example we might have values *good*, *average* and *bad*.
- We could treat the difference between *good* and *average* or between *average* and *bad* as 0.5 and the difference between *good* and *bad* as 1.
- This still does not seem completely right but may be the best we can do in practice.

K-NN Applications

- Handwritten character classification
- Recommender system
- Finding similar documents (documents containing similar topics) or Text Mining

Classroom Exercise-1

Age	Loan	Default	Distance
25	40000	N	
35	60000	N	
45	80000	N	
20	20000	N	
35	120000	N	
52	18000	N	
23	95000	Y	
40	62000	Y	
60	100000	Y	
48	220000	Y	
33	150000	Y	
48	142000	??	

Classroom Exercise-1

Age	Loan	Default	Distance
25	40000	N	102000
35	60000	N	82000
45	80000	N	62000
20	20000	N	122000
35	120000	N	22000
52	18000	N	124000
23	95000	Y	47000
40	62000	Y	80000
60	100000	Y	42000
48	220000	Y	78000
33	150000	Y	8000
48	142000	??	

Classroom Exercise-2

Age	Loan	Default	Distance
0.125	0.11	N	
0.375	0.21	N	
0.625	0.31	N	
0	0.01	N	
0.375	0.5	N	
0.8	0	N	
0.075	0.38	Y	
0.5	0.22	Y	
1	0.41	Y	
0.7	1	Y	
0.325	0.65	Y	
0.7	0.61	??	

Classroom Exercise-2

Age	Loan	Default	Distance
0.125	0.11	N	0.762
0.375	0.21	N	0.5154
0.625	0.31	N	0.3092
0	0.01	N	0.922
0.375	0.5	N	0.3431
0.8	0	N	0.6181
0.075	0.38	Y	0.666
0.5	0.22	Y	0.4383
1	0.41	Y	0.3606
0.7	1	Y	0.39
0.325	0.65	Y	0.3771
0.7	0.61	??	

Classroom Exercise-3

Outlook	Temp (°F)	Humidity (%)	Windy	Class
sunny	75	70	true	play
sunny	80	90	true	don't play
sunny	85	85	false	don't play
sunny	72	95	false	don't play
sunny	69	70	false	play
overcast	72	90	true	play
overcast	83	78	false	play
overcast	64	65	true	play
overcast	81	75	false	play
rain	71	80	true	don't play
rain	65	70	true	don't play
rain	75	80	false	play
rain	68	80	false	play
rain	70	96	false	play

Classes
play, don't play
Outlook
sunny, overcast, rain
Temperature
numerical value
Humidity
numerical value
Windy
true, false

- How can we use this dataset to apply Naïve Bayes and KNN??

Assignment/Exercise

- Building and evaluating KNN classifier with WEKA/Python/Other
  1. Use the dataset for *k*-Nearest Neighbour example in the book (Principles of Data Mining – Max Bramer (2nd Edition)
  2. Use the *glass* dataset and apply the KNN algorithms
     1. Dataset: http://archive.ics.uci.edu/ml/datasets/glass+identification
     2. Apply different distance measure formula and compare the outcome with different value for K as well.

Lecture Reference

- Principles of Data Mining – Max Bramer (2nd Edition)
  - Chapter – 2 (*Introduction to Classification: Naïve Bayes and Nearest Neighbour*)
- Weka - open source machine learning software
  - https://www.cs.waikato.ac.nz/ml/weka/
- Bayes' Theorem
  - https://en.wikipedia.org/wiki/Bayes%27_theorem

References

- Principles of Data Mining – Max Bramer (2nd Edition)
- Data Mininig Concepts and Techniques – Jiawei Han, Michaline Kamber, Jian Pei, 3rd Ed.
- Weka - open source machine learning software
  - https://www.cs.waikato.ac.nz/ml/weka/
- The UCI Repository of Datasets
  - https://archive.ics.uci.edu/ml/index.php
- Kaggle: Your Machine Learning and Data Science Community
  - https://www.kaggle.com/