Data Mining

INFS 4203/7203

Miao Xu

miao.xu@uq.edu.au

The University of Queensland, 2020 Semester 2

Last week

Construction of a decision tree

For a current node:

- Input:
  - Training set $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$
  - Feature set $A = \{a_1, \dots, a_d\}$
- Process: Function **TreeGenerate**(D, A)

1. If **stop criteria** have reached: make the current node a **leaf** node and mark its class, return
2. **Select the optimal** splitting feature a_* from A
3. For all possible values of a_*, generate a branch (child node) and a subset $D_v \subset D$, such that $a_* = v$ for all samples in D_v
4. For all D_v
5. **TreeGenerate**$(D_v, A \backslash \{a_*\})$

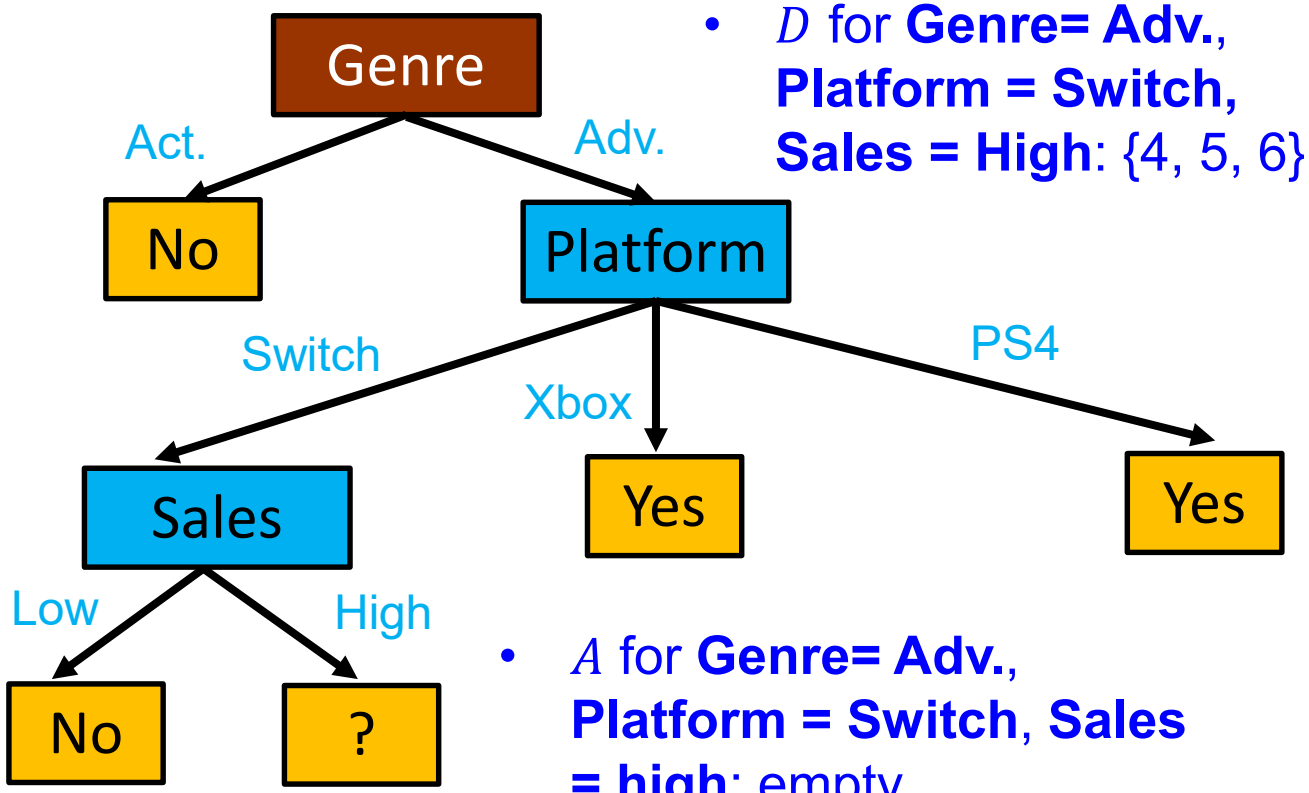
Stop criteria

- A recursive procedure
- It stops when **stop criteria** have reached:
  1. All instances in the current node belong to **the same class** C
     - Mark the leaf node to the class C, return
  2. The **feature set A is empty**, **or** all instances in D have **the same values on A**
     - Mark the leaf node to the **majority** class in D, return
  3. D is **empty**
     - Mark the leaf node to the **majority** class of its **parent** node, return
- Construct a decision tree from the **root node**

Construction of a decision tree: example

- Assume the features are optimally selected in the order
  - Genre , Platform, Sales

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	**High**	**Yes**
5	Adventure	Switch	**High**	**Yes**
6	Adventure	Switch	**High**	**No**
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No



- D for **Genre= Adv., Platform = Switch, Sales = High**: {4, 5, 6}

- A for **Genre= Adv., Platform = Switch**, **Sales = high**: empty

- Criterion 2: The feature set A is empty
- Mark the leaf node to the majority class in D, return

Three measurements of purity

- Information gain (select the largest, ID3)

$$\mathrm{Ent}(D) = -\sum_{i=1}^{m} p_i \log_2 p_i \qquad \mathrm{Gain}(D, a) = \mathrm{Ent}(D) - \sum_{j=1}^{V} \frac{|D_{v_j}|}{|D|} \mathrm{Ent}(D_{v_j})$$

- Gain ratio (select the largest, or as the heuristics in C4.5)

$$\mathrm{IV}(a) = -\sum_{j=1}^{V} \frac{|D_{v_j}|}{|D|} \log_2 \frac{|D_{v_j}|}{|D|} \qquad \mathrm{Gain\_ratio}(D, a) = \frac{\mathrm{Gain}(D,a)}{\mathrm{IV}(D,a)}$$

- Gini index (select the lowest, CART)

$$\mathrm{Gini}(D) = 1 - \sum_{i=1}^{m} p_i^2 \qquad \mathrm{Gini\_index}(D, a) = \sum_{j=1}^{V} \frac{|D_{v_j}|}{|D|} \mathrm{Gini}(D_{v_j})$$

Two pruning approaches

- Pre-pruning (in the construction)
  - Halt tree construction early: do not split a node if this would result in poorer generalization
- Post-pruning (after the construction)
  - Remove branches from a “fully grown” tree, if this would result in better generalization

We use **test data** or **cross validation** to measure the generalization performance.

Deal with continuous values

	0.3	0.4	0.6	0.8	0.9	1.0	1.1	1.2	1.3	1.4
	0.35	**0.5**	**0.7**	**0.85**	**0.95**	**1.05**	**1.15**	**1.25**	**1.35**	
Gini index:	0.4444	0.4000	0.3429	0.2667	0.4000	0.4667	0.4190	0.4750	0.4444	

$$\text{Gini\_index}(\{1-10\}, \text{Genre}) = 0.3429$$

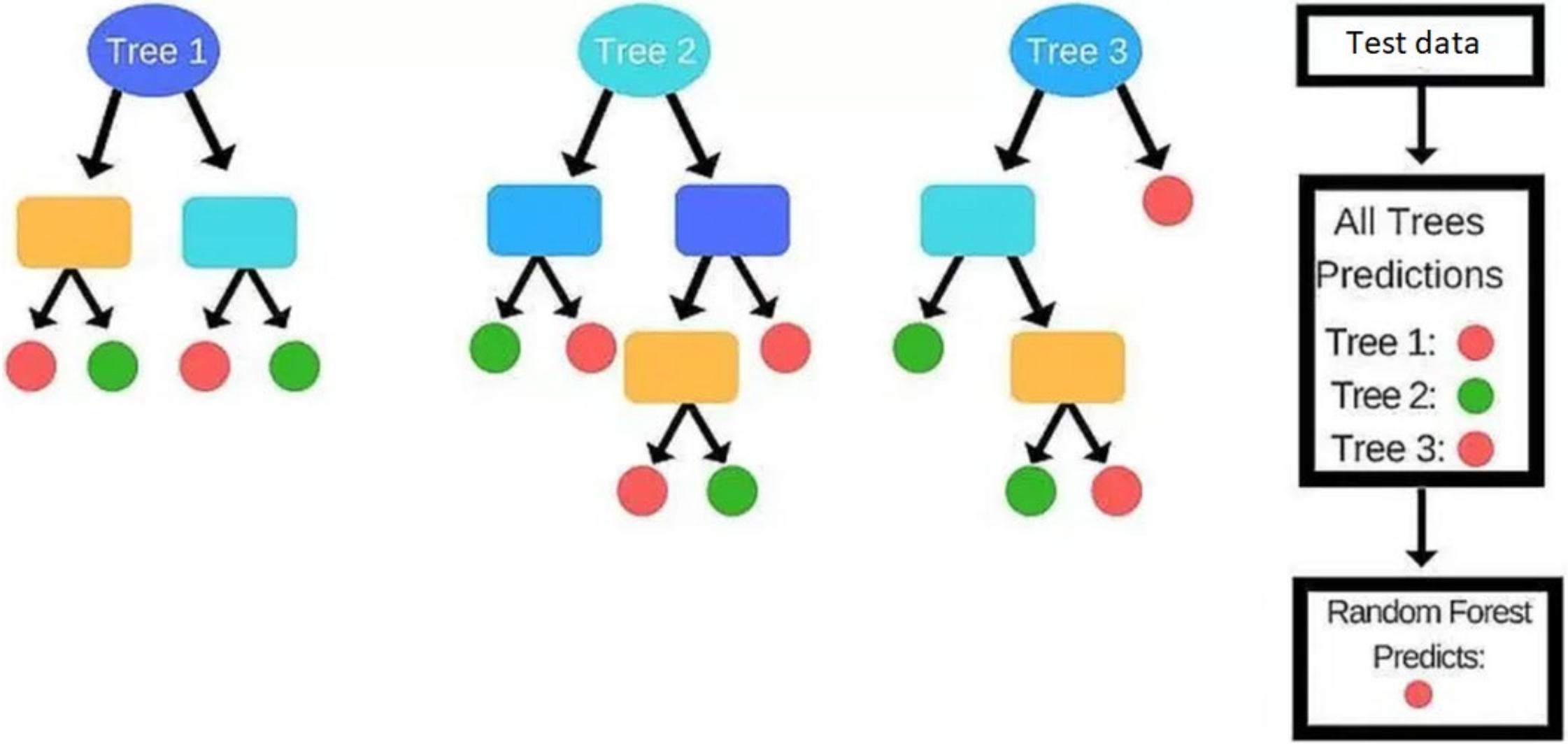
$$\text{Gini\_index}(\{1-10\}, \text{Platform}) = 0.44$$

- Select "Sales = 0.85" as the **first** splitting feature and value!

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	0.8M	No
2	Action	Switch	0.3M	No
3	Action	PS4	0.6M	No
4	Adventure	Switch	0.9M	Yes
5	Adventure	Switch	1.0M	Yes
6	Adventure	Switch	1.1M	No
7	Adventure	Switch	0.4M	No
8	Adventure	PS4	1.2M	Yes
9	Adventure	PS4	1.3M	Yes
10	Adventure	PS4	1.4M	No

Random forest

- Construct several decision trees instead of only one decision tree
- Decision trees are constructed by
  - Determine *k*, which is a positive integer
  - Randomly sample *k* features among all *d* features
  - Construct a decision tree based on the sampled *k* features of all data
- Combine the output of all the trees together
  - E.g. majority voting
- Ensemble learning method: predict with multiple classifiers



https://dataaspirant.com/random-forest-algorithm-machine-learing/

Lecture 9: Classification 3-Naïve Bayesian and *k* Nearest Neighbor

Outline

- Naïve Bayes classifier
- *k*-Nearest Neighbor classifier
- Summary

Classification through conditional probability

- $(\boldsymbol{x}, y)$ are randomly generated from an unknown distribution
- Conditional probability $\mathrm{p}(y|\boldsymbol{x})$: the likelihood of y given $\boldsymbol{x}$
- Example:
  - $\mathrm{p}(\mathrm{Yes}|(\mathrm{Adventure}, \mathrm{Switch}, \mathrm{High}))$
  - $\mathrm{p}(\mathrm{No}|(\mathrm{Action}, \mathrm{Xbox}, \mathrm{Low}))$
- The higher $\mathrm{p}(y|\boldsymbol{x})$ is, the more likely $\boldsymbol{x}$ is classified as y
- Assume there are m classes in total: $C_1, C_2, \ldots, C_m$
  - Predict $\boldsymbol{x}$ as C_j, where $\mathrm{p}(C_j|\boldsymbol{x})$ is the maximum value among $\{\mathrm{p}(C_1|\boldsymbol{x}), \mathrm{p}(C_2|\boldsymbol{x}), \ldots, \mathrm{p}(C_\mathrm{m}|\boldsymbol{x})\}$

How do we estimate $\mathrm{p}(y|\boldsymbol{x})$?

Bayes' theorem

- Joint probability: $\mathrm{p}(\boldsymbol{x}, y)$
- Joint probability can be calculated by:

$$\left.\begin{array}{l} \mathrm{p}(\boldsymbol{x}, y) = \mathrm{p}(\boldsymbol{x}|y) \times \mathrm{p}(y) \\ \text{or} \\ \mathrm{p}(\boldsymbol{x}, y) = \mathrm{p}(y|\boldsymbol{x}) \times \mathrm{p}(\boldsymbol{x}) \end{array}\right\} \mathrm{p}(\boldsymbol{x}|y) \times \mathrm{p}(y) = \mathrm{p}(y|\boldsymbol{x}) \times \mathrm{p}(\boldsymbol{x})$$

- Bayes' theorem:

Class-conditional probability/likelihood

Posterior ← $\mathrm{p}(y|\boldsymbol{x}) = \dfrac{\mathrm{p}(\boldsymbol{x}|y) \times \mathrm{p}(y)}{\mathrm{p}(\boldsymbol{x})}$ → Prior

- For different y, $\mathrm{p}(\boldsymbol{x})$ remains the same
  - To get the maximum value $\mathrm{p}(y|\boldsymbol{x})$, we evaluate $\mathrm{p}(\boldsymbol{x}|y)$ and $\mathrm{p}(y)$

How to evaluate the prior $\mathrm{p}(y)$

- The Law of Large Numbers (LLN) : (informally) the average of the results obtained from a large number of trials should be close to the expected value
- D_y is the set of all samples belong to class y in training set D
- $\mathrm{p}(y)$ is estimated by the ratio of examples belonging to class y in the training dataset, i.e.,

$$\mathrm{p}(y) = \frac{|D_y|}{|D|}$$

How to evaluate the prior $\mathrm{p}(y)$: example

- $y \in \{\mathrm{Yes}, \mathrm{No}\}$
- Yes occurs 4 times. There are totally 10 training instances

$$\mathrm{p}(\mathrm{Yes}) = \frac{4}{10} = 0.4$$

- No occurs 6 times. There are totally 10 training instances

$$\mathrm{p}(\mathrm{No}) = \frac{6}{10} = 0.6$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	**Yes**
5	Adventure	Switch	High	**Yes**
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	**Yes**
9	Adventure	PS4	High	**Yes**
10	Adventure	PS4	High	No

How to evaluate the likelihood $\mathrm{p}(\boldsymbol{x}|y)$

- If we do similar estimation of $\mathrm{p}(y)$ to estimate $\mathrm{p}(\boldsymbol{x}|y)$
  - The training data may not cover all $\boldsymbol{x}$
    - E.g. (Adv., Xbox, High)
  - When the number of features becomes large, it is hard to estimate all the $\mathrm{p}(\boldsymbol{x}|y)$
- For example, if there are 100 features, each feature has two values, the possible values of $\boldsymbol{x}$ can be 2^{100}

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	**Yes**
5	Adventure	Switch	High	**Yes**
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	**Yes**
9	Adventure	PS4	High	**Yes**
10	Adventure	PS4	High	No

Any other way to estimate $\mathrm{p}(\boldsymbol{x}|y)$?

Naïve Bayes Classifier

- Compute $\mathrm{p}(y)$
- Compute $\mathrm{p}(\boldsymbol{x}|y)$ by assuming all features are independent from each other given the class information y, for all $y \in \{C_1, C_2, \ldots, C_m\}$
  - "Attribute conditional independence" assumption
  - The name "Naïve" is given for this strong assumption
- If $\boldsymbol{x} = [x_1, x_2, \ldots, x_d]$, then

$$\mathrm{p}(\boldsymbol{x}|y) = \mathrm{p}(x_1|y) \times \mathrm{p}(x_2|y) \times \cdots \times \mathrm{p}(x_d|y)$$

- Predict $\boldsymbol{x}$ as C_j, where $\mathrm{p}(C_j)\mathrm{p}(\boldsymbol{x}|C_j)$ is the maximum value among $\{\mathrm{p}(C_1)\mathrm{p}(\boldsymbol{x}|C_1), \mathrm{p}(C_2)\mathrm{p}(\boldsymbol{x}|C_2), \ldots, \mathrm{p}(C_m)\mathrm{p}(\boldsymbol{x}|C_\mathrm{m})\}$

How do we estimate $\mathrm{p}(x_i|y)$?

How to estimate $\mathrm{p}(x_i|y)$

- D_y is the set of all samples belong to class y in training set D
- D_{y,x_i} is the set of samples in D_y with the i-th feature equal to x_i
- $\mathrm{p}(x_i|y)$ is

$$\mathrm{p}(x_i|y) = \frac{|D_{y,x_i}|}{|D_y|}$$

Compute $p(\boldsymbol{x}|y)$ in Naïve Bayes classifier

- For feature Genre
  - Buy? = No: 6 examples
    - Genre = Action: 3 examples

$$p(\text{Genre} = \text{Action}|\text{Buy?} = \text{No}) = \frac{3}{6} = 0.5$$

    - Genre = Adventure: 3 examples

$$p(\text{Genre} = \text{Adventure}|\text{Buy?} = \text{No}) = \frac{3}{6} = 0.5$$

  - Buy? = Yes: 4 examples
    - Genre = Action: 0 examples

$$p(\text{Genre} = \text{Action}|\text{Buy?} = \text{Yes}) = \frac{0}{4} = 0$$

    - Genre = Adventure: 4 examples

$$p(\text{Genre} = \text{Adventure}|\text{Buy?} = \text{Yes}) = \frac{4}{4} = 1$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	**Yes**
5	Adventure	Switch	High	**Yes**
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	**Yes**
9	Adventure	PS4	High	**Yes**
10	Adventure	PS4	High	No

Compute $p(\boldsymbol{x}|y)$ in Naïve Bayes classifier

- For feature Platform
  - Buy? = No: 6 examples
    - Platform = Xbox: 1 examples

$$p(\text{Platform} = \text{Xbox}|\text{Buy?} = \text{No}) = \frac{1}{6} = 0.2$$

    - Platform = Switch: 3 examples

$$p(\text{Platform} = \text{Switch}|\text{Buy?} = \text{No}) = \frac{3}{6} = 0.5$$

    - Platform = PS4: 2 examples

$$p(\text{Platform} = \text{PS4}|\text{Buy?} = \text{No}) = \frac{2}{6} = 0.3$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	**Yes**
5	Adventure	Switch	High	**Yes**
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	**Yes**
9	Adventure	PS4	High	**Yes**
10	Adventure	PS4	High	No

Activity: Compute $p(x|y)$ in Naïve Bayes classifier

- For feature Platform
  - Buy? = Yes: ? examples
    - Platform = Xbox: ? examples

      $p(\text{Platform} = \text{Xbox}|\text{Buy?} = \text{Yes}) = ?$

    - Platform = Switch: ? examples

      $p(\text{Platform} = \text{Switch}|\text{Buy?} = \text{Yes}) = ?$

    - Platform = PS4: ? examples

      $p(\text{Platform} = \text{PS4}|\text{Buy?} = \text{Yes}) = ?$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	**Yes**
5	Adventure	Switch	High	**Yes**
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	**Yes**
9	Adventure	PS4	High	**Yes**
10	Adventure	PS4	High	No

Compute $p(\boldsymbol{x}|y)$ in Naïve Bayes classifier

- For feature Platform
  - Buy? = Yes: 4 examples
    - Platform = Xbox: 0 examples

$$p(\text{Platform} = \text{Xbox}|\text{Buy?} = \text{Yes}) = \frac{0}{4} = 0$$

    - Platform = Switch: 2 examples

$$p(\text{Platform} = \text{Switch}|\text{Buy?} = \text{Yes}) = \frac{2}{4} = 0.5$$

    - Platform = PS4: 2 examples

$$p(\text{Platform} = \text{PS4}|\text{Buy?} = \text{Yes}) = \frac{2}{4} = 0.5$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	**Yes**
5	Adventure	Switch	High	**Yes**
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	**Yes**
9	Adventure	PS4	High	**Yes**
10	Adventure	PS4	High	No

Compute $p(\boldsymbol{x}|y)$ in Naïve Bayes classifier

- For feature Sales
  - Buy? = No: 6 examples
    - Sales = High: 4 examples

$$p(\text{Sales} = \text{High}|\text{Buy?} = \text{No}) = \frac{4}{6} = 0.6$$

    - Sales = Low : 2 examples

$$p(\text{Sales} = \text{Low}|\text{Buy?} = \text{No}) = \frac{2}{6} = 0.3$$

  - Buy? = Yes: 4 examples
    - Sales = High: 4 examples

$$p(\text{Sales} = \text{High}|\text{Buy?} = \text{Yes}) = \frac{4}{4} = 1$$

    - Sales = Low: 0 examples

$$p(\text{Sales} = \text{Low}|\text{Buy?} = \text{Yes}) = \frac{0}{4} = 0$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	**Yes**
5	Adventure	Switch	High	**Yes**
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	**Yes**
9	Adventure	PS4	High	**Yes**
10	Adventure	PS4	High	No

Prediction with Naïve Bayes classifier

- Predict $\boldsymbol{x}$ as C_j, where $\mathrm{p}(C_j)\mathrm{p}(\boldsymbol{x}|C_j)$ is the maximum value among $\{\mathrm{p}(C_1)\mathrm{p}(\boldsymbol{x}|C_1), \mathrm{p}(C_2)\mathrm{p}(\boldsymbol{x}|C_2), \dots, \mathrm{p}(C_m)\mathrm{p}(\boldsymbol{x}|C_m)\}$
- For (Adventure, Switch, High), we have

$$\mathrm{p}(\text{Yes}) \times \mathrm{p}(\text{Genre} = \text{Adventure}, \text{Platform} = \text{Switch}, \text{Sales} = \text{High}|\text{Buy?} = \text{Yes})$$
$$= \mathrm{p}(\text{Yes}) \times \mathrm{p}(\text{Genre} = \text{Adventure}|\text{Buy?} = \text{Yes})\ \mathrm{p}(\text{Platform} = \text{Switch}|\text{Buy?} = \text{Yes})\ \mathrm{p}(\text{Sales} = \text{High}|\text{Buy?} = \text{Yes})$$
$$= 0.4 \times 1 \times 0.5 \times 1 = 0.2$$

$$\mathrm{p}(\text{No}) \times \mathrm{p}(\text{Genre} = \text{Adventure}, \text{Platform} = \text{Switch}, \text{Sales} = \text{High}|\text{Buy?} = \text{No})$$
$$= \mathrm{p}(\text{No}) \times \mathrm{p}(\text{Genre} = \text{Adventure}|\text{Buy?} = \text{No})\ \mathrm{p}(\text{Platform} = \text{Switch}|\text{Buy?} = \text{No})\ \mathrm{p}(\text{Sales} = \text{High}|\text{Buy?} = \text{No})$$
$$= 0.6 \times 0.5 \times 0.5 \times 0.6 = 0.09 < 0.2$$

Prediction: **Yes**

- In this way, (Adventure, Switch, High) is predicted as Yes
- How about (Action, PS4, High)?

Activity: Predict by Naïve Bayes classifier

- What is the prediction of (Action, PS4, High) ?

- $y \in \{\text{Yes}, \text{No}\}$
- Yes occurs 4 times. There are totally 10 training instances

$$p(\text{Yes}) = \frac{4}{10} = 0.4$$

- No occurs 6 times. There are totally 10 training instances

$$p(\text{No}) = \frac{6}{10} = 0.6$$

- For feature Genre
  - Buy? = No: 6 examples
    - Genre = Action: 3 examples

$$p(\text{Genre} = \text{Action}|\text{Buy?} = \text{No}) = \frac{3}{6} = 0.5$$

    - Genre = Adventure: 3 examples

$$p(\text{Genre} = \text{Adventure}|\text{Buy?} = \text{No}) = \frac{3}{6} = 0.5$$

  - Buy? = Yes: 4 examples
    - Genre = Action: 0 examples

$$p(\text{Genre} = \text{Action}|\text{Buy?} = \text{Yes}) = \frac{0}{4} = 0$$

    - Genre = Adventure: 4 examples

$$p(\text{Genre} = \text{Adventure}|\text{Buy?} = \text{Yes}) = \frac{4}{4} = 1$$

- For feature Platform
  - Buy? = Yes: 4 examples
    - Platform = Xbox: 0 examples

$$p(\text{Platform} = \text{Xbox}|\text{Buy?} = \text{Yes}) = \frac{0}{4} = 0$$

    - Platform = Switch: 2 examples

$$p(\text{Platform} = \text{Switch}|\text{Buy?} = \text{Yes}) = \frac{2}{4} = 0.5$$

    - Platform = PS4: 2 examples

$$p(\text{Platform} = \text{PS4}|\text{Buy?} = \text{Yes}) = \frac{2}{4} = 0.5$$

- For feature Platform
  - Buy? = No: 6 examples
    - Platform = Xbox: 1 examples

$$p(\text{Platform} = \text{Xbox}|\text{Buy?} = \text{No}) = \frac{1}{6} = 0.2$$

    - Platform = Switch: 3 examples

$$p(\text{Platform} = \text{Switch}|\text{Buy?} = \text{No}) = \frac{3}{6} = 0.5$$

    - Platform = PS4: 2 examples

$$p(\text{Platform} = \text{PS4}|\text{Buy?} = \text{No}) = \frac{2}{6} = 0.3$$

- For feature Sales
  - Buy? = No: 6 examples
    - Sales = High: 4 examples

$$p(\text{Sales} = \text{High}|\text{Buy?} = \text{No}) = \frac{4}{6} = 0.6$$

    - Sales = Low : 2 examples

$$p(\text{Sales} = \text{Low}|\text{Buy?} = \text{No}) = \frac{2}{6} = 0.3$$

  - Buy? = Yes: 4 examples
    - Sales = High: 4 examples

$$p(\text{Sales} = \text{High}|\text{Buy?} = \text{Yes}) = \frac{4}{4} = 1$$

    - Sales = Low: 0 examples

$$p(\text{Sales} = \text{Low}|\text{Buy?} = \text{Yes}) = \frac{0}{4} = 0$$

Prediction with Naïve Bayes classifier

- For (Action, PS4, High), we have

$$p(\text{Yes}) \times p(\text{Genre} = \text{Action}, \text{Platform} = \text{PS4}, \text{Sales} = \text{High} | \text{Buy?} = \text{Yes})$$
$$= p(\text{Yes}) \times p(\text{Genre} = \text{Action} | \text{Buy?} = \text{Yes})\ p(\text{Platform} = \text{PS4} | \text{Buy?} = \text{Yes})\ p(\text{Sales} = \text{High} | \text{Buy?} = \text{Yes})$$
$$= 0.4 \times 0 \times 0.5 \times 1 = 0$$

$$p(\text{No}) \times p(\text{Genre} = \text{Action}, \text{Platform} = \text{Xbox}, \text{Sales} = \text{High} | \text{Buy?} = \text{No})$$
$$= p(\text{No}) \times p(\text{Genre} = \text{Action} | \text{Buy?} = \text{No})\ p(\text{Platform} = \text{PS4} | \text{Buy?} = \text{No})\ p(\text{Sales} = \text{High} | \text{Buy?} = \text{No})$$
$$= 0.6 \times 0.5 \times 0.3 \times 0.6 = 0.054$$

Prediction: **No**

- How to handle the zero probability?
  - If any zero-probability exists, we **have to** use "Laplacian correction" in Naïve Bayes classifier

Laplacian correction

- Assume there are m classes in total: $C_1, C_2, \ldots, C_m$
- Assume for the i-th feature, there are V_i possible values:
  - $\{v_1, v_2, \ldots, v_{V_i}\}$
- D_y, D_{y,x_i} defined as before
- After Laplacian correction:

$$\hat{\mathrm{p}}(y) = \frac{|D_y| + 1}{|D| + m} \qquad \hat{\mathrm{p}}(x_i|y) = \frac{|D_{y,x_i}| + 1}{|D_y| + V_i}$$

- It fixes the problem of zero probability. When dataset becomes large, the effect of correction on non-zero probability can be negligible

Laplacian correction: example

- $y \in \{\text{Yes}, \text{No}\}$, 2 classes
- Yes occurs 4 times, there are total 10 training instances

$$\hat{\text{p}}(\text{Yes}) = \frac{4+1}{10+2} = 0.4$$

- No occurs 6 times, there are total 10 training instances

$$\hat{\text{p}}(\text{No}) = \frac{6+1}{10+2} = 0.6$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	**Yes**
5	Adventure	Switch	High	**Yes**
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	**Yes**
9	Adventure	PS4	High	**Yes**
10	Adventure	PS4	High	No

Laplacian correction: example

- For feature Genre, 2 values
  - Buy? = No: 6 examples
    - Genre = Action: 3 examples

$$\hat{p}(\text{Genre} = \text{Action}|\text{Buy?} = \text{No}) = \frac{3+1}{6+2} = 0.5$$

    - Genre = Adventure: 3 examples

$$\hat{p}(\text{Genre} = \text{Adventure}|\text{Buy?} = \text{No}) = \frac{3+1}{6+2} = 0.5$$

  - Buy? = Yes: 4 examples
    - Genre = Action: 0 examples

$$\hat{p}(\text{Genre} = \text{Action}|\text{Buy?} = \text{Yes}) = \frac{0+1}{4+2} = 0.2$$

    - Genre = Adventure: 4 examples

$$\hat{p}(\text{Genre} = \text{Adventure}|\text{Buy?} = \text{Yes}) = \frac{4+1}{4+2} = 0.8$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	**Yes**
5	Adventure	Switch	High	**Yes**
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	**Yes**
9	Adventure	PS4	High	**Yes**
10	Adventure	PS4	High	No

Non-zero now

Activity: Laplacian correction

- For feature Platform, ? values
  - Buy? = No: ? examples
    - Platform = Xbox: ? examples

$$\hat{p}(\text{Platform} = \text{Xbox}|\text{Buy?} = \text{No}) = ?$$

    - Platform = Switch: ? examples

$$\hat{p}(\text{Platform} = \text{Switch}|\text{Buy?} = \text{No}) = ?$$

    - Platform = PS4: ? examples

$$\hat{p}(\text{Platform} = \text{PS4}|\text{Buy?} = \text{No}) = ?$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	**Yes**
5	Adventure	Switch	High	**Yes**
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	**Yes**
9	Adventure	PS4	High	**Yes**
10	Adventure	PS4	High	No

Laplacian correction: example

- For feature Platform, 3 values
  - Buy? = No: 6 examples
    - Platform = Xbox: 1 examples

$$\hat{p}(\text{Platform} = \text{Xbox}|\text{Buy?} = \text{No}) = \frac{1+1}{6+3} = 0.2$$

    - Platform = Switch: 3 examples

$$\hat{p}(\text{Platform} = \text{Switch}|\text{Buy?} = \text{No}) = \frac{3+1}{6+3} = 0.4$$

    - Platform = PS4: 2 examples

$$\hat{p}(\text{Platform} = \text{PS4}|\text{Buy?} = \text{No}) = \frac{2+1}{6+3} = 0.3$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	**Yes**
5	Adventure	Switch	High	**Yes**
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	**Yes**
9	Adventure	PS4	High	**Yes**
10	Adventure	PS4	High	No

Laplacian correction: example

- For feature Platform, 3 values
  - Buy? = Yes: 4 examples
    - Platform = Xbox: 0 examples

Non-zero now

$$\hat{p}(\text{Platform} = \text{Xbox}|\text{Buy?} = \text{Yes}) = \frac{0+1}{4+3} = 0.1$$

    - Platform = Switch: 2 examples

$$\hat{p}(\text{Platform} = \text{Switch}|\text{Buy?} = \text{Yes}) = \frac{2+1}{4+3} = 0.4$$

    - Platform = PS4: 2 examples

$$\hat{p}(\text{Platform} = \text{PS4}|\text{Buy?} = \text{Yes}) = \frac{2+1}{4+3} = 0.4$$

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	**Yes**
5	Adventure	Switch	High	**Yes**
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	**Yes**
9	Adventure	PS4	High	**Yes**
10	Adventure	PS4	High	No

Laplacian correction: example

- For feature Sales, 2 values
  - Buy? = No: 6 examples
    - Sales = High: 4 examples

$$\hat{p}(\text{Sales} = \text{High}|\text{Buy?} = \text{No}) = \frac{4+1}{6+2} = 0.6$$

    - Sales = Low : 2 examples

$$\hat{p}(\text{Sales} = \text{Low}|\text{Buy?} = \text{No}) = \frac{2+1}{6+2} = 0.4$$

  - Buy? = Yes: 4 examples
    - Sales = High: 4 examples

$$\hat{p}(\text{Sales} = \text{High}|\text{Buy?} = \text{Yes}) = \frac{4+1}{4+2} = 0.8$$

    - Sales = Low: 0 examples

$$\hat{p}(\text{Sales} = \text{Low}|\text{Buy?} = \text{Yes}) = \frac{0+1}{4+2} = 0.2$$

← Non-zero now

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	**Yes**
5	Adventure	Switch	High	**Yes**
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	**Yes**
9	Adventure	PS4	High	**Yes**
10	Adventure	PS4	High	No

Laplacian correction: prediction

- For (Adventure, Switch, High), we have

$$\hat{p}\,(\text{Yes}) \times \hat{p}(\text{Genre} = \text{Adventure}, \text{Platform} = \text{Switch}, \text{Sales} = \text{High}|\text{Buy?} = \text{Yes})$$
$$= \hat{p}(\text{Yes}) \times \hat{p}(\text{Genre} = \text{Adventure}|\text{Buy?} = \text{Yes})\ \hat{p}\,(\text{Platform} = \text{Switch}|\text{Buy?} = \text{Yes})\ \hat{p}\,(\text{Sales} = \text{High}|\text{Buy?} = \text{Yes})$$
$$= 0.4 \times 0.8 \times 0.4 \times 0.8 = 0.1024$$

$$\hat{p}\,(\text{No}) \times \hat{p}(\text{Genre} = \text{Adventure}, \text{Platform} = \text{Switch}, \text{Sales} = \text{High}|\text{Buy?} = \text{No})$$
$$= \hat{p}(\text{No}) \times \hat{p}(\text{Genre} = \text{Adventure}|\text{Buy?} = \text{No})\ \hat{p}\,(\text{Platform} = \text{Switch}|\text{Buy?} = \text{No})\ \hat{p}\,(\text{Sales} = \text{High}|\text{Buy?} = \text{No})$$
$$= 0.6 \times 0.5 \times 0.4 \times 0.6 = 0.072 < 0.1024$$

Prediction: **Yes**

- For (Action, PS4, High), we have

$$\hat{p}\,(\text{Yes}) \times \hat{p}(\text{Genre} = \text{Action}, \text{Platform} = \text{PS4}, \text{Sales} = \text{High}|\text{Buy?} = \text{Yes})$$
$$= \hat{p}(\text{Yes}) \times \hat{p}(\text{Genre} = \text{Action}|\text{Buy?} = \text{Yes})\ \hat{p}\,(\text{Platform} = \text{PS4}|\text{Buy?} = \text{Yes})\ \hat{p}\,(\text{Sales} = \text{High}|\text{Buy?} = \text{Yes})$$
$$= 0.4 \times 0.2 \times 0.4 \times 0.8 = 0.0256$$

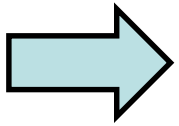
$$\hat{p}\,(\text{No}) \times \hat{p}(\text{Genre} = \text{Action}, \text{Platform} = \text{PS4}, \text{Sales} = \text{High}|\text{Buy?} = \text{No})$$
$$= \hat{p}(\text{No}) \times \hat{p}(\text{Genre} = \text{Action}|\text{Buy?} = \text{No})\ \hat{p}\,(\text{Platform} = \text{PS4}|\text{Buy?} = \text{No})\ \hat{p}\,(\text{Sales} = \text{High}|\text{Buy?} = \text{No})$$
$$= 0.6 \times 0.5 \times 0.3 \times 0.6 = 0.054 > 0.0256$$

Prediction: **No**

Continuous values

Training data

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	High	No
2	Action	Switch	Low	No
3	Action	PS4	High	No
4	Adventure	Switch	High	Yes
5	Adventure	Switch	High	Yes
6	Adventure	Switch	High	No
7	Adventure	Switch	Low	No
8	Adventure	PS4	High	Yes
9	Adventure	PS4	High	Yes
10	Adventure	PS4	High	No



Training data

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	0.8M	No
2	Action	Switch	0.3M	No
3	Action	PS4	0.6M	No
4	Adventure	Switch	0.9M	Yes
5	Adventure	Switch	1.0M	Yes
6	Adventure	Switch	1.1M	No
7	Adventure	Switch	0.4M	No
8	Adventure	PS4	1.2M	Yes
9	Adventure	PS4	1.3M	Yes
10	Adventure	PS4	1.4M	No

- Probability density estimation
  - Assume the values of the feature follow an univariate gaussian distribution

$$x_i \sim N(\mu, \sigma^2): \mathrm{p}(x_i|y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{(x_i-\mu)^2}{2\sigma^2} \right\}$$

  - Use data of class y to estimate parameters of distribution (e.g., mean μ and variance σ^2)
  - Use the probability distribution to estimate $\mathrm{p}(x_i|y)$

Continuous value: example

Sales	Buy?
0.8M	No
0.3M	No
0.6M	No
1.1M	No
0.4M	No
1.4M	No

- Data of class No:
  - μ: $\frac{0.8+0.3+0.6+1.1+0.4+1.4}{6} = 0.77$
  - σ^2: $\frac{[(0.8-0.77)^2+(0.3-0.8)^2+(0.6-0.8)^2+(1.1-0.8)^2+(0.4-0.8)^2+(1.4-0.8)^2]}{5} = 0.18$
  - σ: 0.42
  - $\mathrm{p}(x_i|\mathrm{No}) = \frac{1}{\sqrt{2\pi}\times 0.42}\exp\left\{-\frac{(x_i-0.77)^2}{2\times 0.18}\right\}$
  - For example

$$\mathrm{p}(0.5\mathrm{M}|\mathrm{No}) = \frac{1}{\sqrt{2\pi}\times 0.42}\exp\left\{-\frac{(x_i-0.77)^2}{2\times 0.18}\right\}=0.78$$

Activity: continuous value

- What is the following value?

$$\mathrm{p}(0.5\mathrm{M}|\mathrm{Yes}) = ?$$

Sales	Buy?
0.9M	Yes
1.0M	Yes
1.2M	Yes
1.3M	Yes

$$\mathrm{p}(x_i|y) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(x_i-\mu)^2}{2\sigma^2}\right\}$$

Continuous value: example

Sales	Buy?
0.9M	Yes
1.0M	Yes
1.2M	Yes
1.3M	Yes

- Data of class Yes:
  - μ: (0.9+1.0+1.2+1.3)/4 = 1.1
  - σ^2: $\frac{[(0.9-1.1)^2+(1.0-1.1)^2+(1.2-1.1)^2+(1.3-1.1)^2]}{3} = 0.0333$
  - σ: 0.1826
  - $p(x_i|\text{Yes}) = \frac{1}{\sqrt{2\pi}\times 0.1826}\exp\left\{-\frac{(x_i-1.1)^2}{2\times 0.0333}\right\}$
  - For example

$$p(0.5\text{M}|\text{Yes}) = \frac{1}{\sqrt{2\pi}\times 0.1826}\exp\left\{-\frac{(x_i-1.1)^2}{2\times 0.0333}\right\} = 0.0099$$

Summarize of NB classifier

- Training:
  - Calculate all $\mathrm{p}(y)$
  - Calculate all possible $\mathrm{p}(x_i|y)$ (assume continuous values follows univariate gaussian distribution)
- Prediction:
  - If $\boldsymbol{x} = [x_1, x_2, \ldots, x_d]$, then calculate for all possible y

$$\mathrm{p}(y) \times \mathrm{p}(x_1|y) \times \mathrm{p}(x_2|y) \times \cdots \times \mathrm{p}(x_d|y) = \mathrm{p}(y) \times \mathrm{p}(\boldsymbol{x}|y)$$

  - Predict $\boldsymbol{x}$ as C_j, where $\mathrm{p}(C_j)\mathrm{p}(\boldsymbol{x}|C_j)$ is the **maximum** value among $\{\mathrm{p}(C_1)\mathrm{p}(\boldsymbol{x}|C_1), \mathrm{p}(C_2)\mathrm{p}(\boldsymbol{x}|C_2), \ldots, \mathrm{p}(C_m)\mathrm{p}(\boldsymbol{x}|C_\mathrm{m})\}$

Beyond NB: other ways to estimate $p(\boldsymbol{x}|y)$

- Maximum likelihood estimation (MLE)
- Semi-naïve Bayes classifier
- Bayesian network (or called belief network)
- Expectation–maximization algorithm (EM algorithm) to consider latent variables

Outline

- Naïve Bayes classifier
- *k*-Nearest Neighbor classifier
- Summary

Basic idea

- Find from historical record those as similar as possible to the new record

Niffler

Nearest Neighbor Classifier

Requires three input:

- The set of **training data**
- **Distance metric** to compute distances between data points
- The value of k, the number of nearest neighbours

Nearest Neighbor Classifier: Prediction

For a test example

- Compute its distance to all training samples
- Identify k nearest neighbours
- Use labels of nearest neighbours to determine the label of test example (e.g., by taking majority vote—the majority class among the nearest neighbours)

*k-Nearest Neighbour (k-NN for short) classifier is a **lazy learning** approach (or called lazy learner), because the training data is just memorized, without any model built in the training phase.*

Nearest Neighbor Classifier: example

Customer	Age	Income (K)	Games played	Buy?
John	35	35	3	Yes
Rachel	22	50	2	No
Ruth	63	200	1	No
Tom	59	170	1	No
Neil	25	40	4	Yes
David	37	50	2	?

Euclidean Distance from David
$\sqrt{(35-37)^2+(35-50)^2+(3-2)^2} = \mathbf{15.16}$
$\sqrt{(22-37)^2+(50-50)^2+(2-2)^2} = \mathbf{15}$
$\sqrt{(63-37)^2+(200-50)^2+(1-2)^2} = \mathbf{152.23}$
$\sqrt{(59-37)^2+(170-50)^2+(1-2)^2} = \mathbf{122}$
$\sqrt{(25-37)^2+(40-50)^2+(4-2)^2} = \mathbf{15.74}$

If k = 1, the nearest neighbour is Rachel, and we predict it as “No”
If k = 3, the nearest neighbours are Rachel, John, and Neil, we predict it as “Yes”
If k =5, the nearest neighbours are Rachel, John, Neil, Tom and Ruth. We predict is as “No”

Data normalization

- Attribute values may have to be scaled to prevent distance measures from being dominated by one of the attributes
- Example:
  - Age of a person may vary from 0 to 120
  - Income of a person may vary from $10K to $1M
  - Number of games played of a person may vary from 0 to 1000

Normalization: example of standardization

Customer	Age	Income (K)	Games played	Buy?
John	35	35	3	Yes
Rachel	22	50	2	No
Ruth	63	200	1	No
Tom	59	170	1	No
Neil	25	40	4	Yes
David	37	50	2	?

Standardization:

$$x' = \frac{x - \mu}{\sigma}$$

1. Calculate the mean of each feature of the training data
2. Calculate the variance of each feature of the training data
3. Normalize the training data feature by feature
4. When test data come, normalize the test data in the same way the training data is normalized, and do prediction

Normalization: example of standardization

Customer	Age	Income (K)	Games played	Buy?
John	35	35	3	Yes
Rachel	22	50	2	No
Ruth	63	200	1	No
Tom	59	170	1	No
Neil	25	40	4	Yes
David	37	50	2	?

Customer	Age	Income (K)	Games played	Buy?
John	-0.30	-0.81	0.61	Yes
Rachel	-0.98	-0.62	-0.15	No
Ruth	1.16	1.27	-0.92	No
Tom	0.95	0.89	-0.92	No
Neil	-0.83	-0.74	1.38	Yes
David	-0.20	-0.62	-0.15	?

Normalization: example of standardization

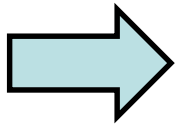
Customer	Age	Income (K)	Games played	Buy?
John	-0.30	-0.81	0.61	Yes
Rachel	-0.98	-0.62	-0.15	No
Ruth	1.16	1.27	-0.92	No
Tom	0.95	0.89	-0.92	No
Neil	-0.83	-0.74	1.38	Yes
David	-0.20	-0.62	-0.15	?

Euclidean Distance from David
0.7897
0.7800
2.4525
2.0483
1.6590

If k = 1, the nearest neighbour is Rachel, and we predict it as “No”
If k = 3, the nearest neighbours are Rachel, John, and Neil, we predict it as “Yes”
If k =5, the nearest neighbours are Rachel, John, Neil, Tom and Ruth. We predict is as “No”

Non-numerical features

TID	Genre	Platform	Sales	Buy?
1	Action	Xbox	0.8M	No
2	Action	Switch	0.3M	No
3	Action	PS4	0.6M	No
4	Adventure	Switch	0.9M	Yes
5	Adventure	Switch	1.0M	Yes
6	Adventure	Switch	1.1M	No
7	Adventure	Switch	0.4M	No
8	Adventure	PS4	1.2M	Yes
9	Adventure	PS4	1.3M	Yes
10	Adventure	PS4	1.4M	No



TID	Genre	Xbox	Switch	PS4	Sales	Buy?
1	1	1	0	0	0.8M	No
2	1	0	1	0	0.3M	No
3	1	0	0	1	0.6M	No
4	0	0	1	0	0.9M	Yes
5	0	0	1	0	1.0M	Yes
6	0	0	1	0	1.1M	No
7	0	0	1	0	0.4M	No
8	0	0	0	1	1.2M	Yes
9	0	0	0	1	1.3M	Yes
10	0	0	0	1	1.4M	No

- Followed by the k-NN method.
- Normalization could also be applied if necessary.

Summarize k-NN

- Transfer non-numerical features into numerical
- Normalize the training data if necessary
- Specify k and distance metric, and store all training data
- For a test example, do normalization if necessary; then calculate its distance to all training instances
- Take all labels of the k-nearest neighbours in the training data, and then make prediction by majority voting (or other voting methods)

Determine k

Choosing the value of k:

- If k is too small, the prediction can easily be "wrong" if there is noisy point (be sensitive to noisy points)
- If k is too large, neighbourhood may include too many points from other classes

If k = 1, the nearest neighbour is Rachel, and we predict it as "No"
If k = 3, the nearest neighbours are Rachel, John, and Neil, we predict it as "Yes"
If k = 5, the nearest neighbours are Rachel, John, Neil, Tom and Ruth. We predict is as "No"

It is preferred to choose odd numbers for k. Why?

Because we do majority voting—no ties using odd numbers for binary classification

Determine k by cross-validation: example of 3-CV

- Randomly partition the data into three folders:

	TID	Genre	Xbox	Switch	PS4	Sales	Buy?
Folder 1	1	1	1	0	0	0.8M	No
	2	1	0	1	0	0.3M	No
	3	1	0	0	1	0.6M	No
Folder 2	4	0	0	1	0	0.9M	Yes
	5	0	0	1	0	1.0M	Yes
	6	0	0	1	0	1.1M	No
Folder 3	7	0	0	1	0	0.4M	No
	8	0	0	0	1	1.2M	Yes
	9	0	0	0	1	1.3M	Yes
	10	0	0	0	1	1.4M	No

- Select an evaluation metric, e.g. accuracy

Activity: k-NN prediction

- Using folder 1 and folder 2 as training data, and folder 3 as test data
- Euclidean distances from training data and test data

TID	1	2	3	4	5	6
7	1.78	1.00	1.74	0.50	0.60	0.70
8	1.78	1.95	1.17	1.45	1.43	1.42
9	1.80	2.00	1.22	1.47	1.45	1.43
10	1.83	2.05	1.28	1.50	1.47	1.45

TID	Buy?
1	No
2	No
3	No
4	Yes
5	Yes
6	No

- Prediction

TID	k = 1	k = 3	k = 5	Ground Truth
7	4 Yes	4,5,6 Yes	4,5,6,2,3 No	No
8	?	?	?	Yes
9				Yes
10				No

Trial 1

- Using folder 1 and folder 2 as training data, and folder 3 as test data
- Euclidean distances from training data and test data

TID	1	2	3	4	5	6
7	1.78	1.00	1.74	0.50	0.60	0.70
8	1.78	1.95	1.17	1.45	1.43	1.42
9	1.80	2.00	1.22	1.47	1.45	1.43
10	1.83	2.05	1.28	1.50	1.47	1.45

TID	Buy?
1	No
2	No
3	No
4	Yes
5	Yes
6	No

- Prediction

TID	k = 1	k = 3	k = 5	Ground Truth
7	4 Yes	4,5,6 Yes	4,5,6,2,3 No	No
8	3 No	3,6,5 No	3,6,5,4,1 No	Yes
9	3 No	3,6,5 No	3,6,5,4,1 No	Yes
10	3 No	3,6,5 No	3,6,5,4,1 No	No

Accuracy for k = 1: 25%
Accuracy for k = 3: 25%
Accuracy for k = 5: 50%

Trial 2

- Using folder 1 and folder 3 as training data, and folder 2 as test data
- Euclidean distances from training data and test data

TID	1	2	3	7	8	9	10
4	1.73	1.17	1.76	0.50	1.45	1.47	1.50
5	1.74	1.22	1.78	0.60	1.43	1.45	1.47
6	1.76	1.28	1.80	0.70	1.42	1.43	1.45

- Prediction

TID	k = 1	k = 3	k = 5	Ground Truth
4	7 No	7,2,8 No	7,2,8,9,10 No	Yes
5	7 No	7,2,8 No	7,2,8,9,10 No	Yes
6	7 No	7,2,8 No	7,2,8,9,10 No	No

Accuracy for k = 1: 33%
Accuracy for k = 3: 33%
Accuracy for k = 5: 33%

Trial 3

- Using folder 2 and folder 3 as training data, and folder 1 as test data
- Euclidean distances from training data and test data

TID	4	5	6	7	8	9	10
1	1.73	1.74	1.76	1.78	1.78	1.80	1.83
2	1.17	1.22	1.28	1.00	1.95	2.00	2.05
3	1.76	1.78	1.80	1.74	1.17	1.22	1.28

- Prediction

TID	k = 1	k = 3	k = 5	Ground Truth
1	4 Yes	4,5,6 Yes	4,5,6,7,8 Yes	No
2	7 No	7,4,5 Yes	7,4,5,6,8 Yes	No
3	8 Yes	8,9,10 Yes	8,9,10,7,4 Yes	No

Accuracy for k = 1: 33%
Accuracy for k = 3: 0%
Accuracy for k = 5: 0%

- Summarize the results of 3-CV

	K = 1	K = 3	K = 5
Trial 1	25%	25%	50%
Trial 2	33%	33%	33%
Trial 3	33%	0%	0%
Average	30%	19%	28%

  - We select k = 1 for it has the highest average accuracy
- If data is enough, we usually 10-CV or even 30-CV; to save time, we may use 3-CV or 5-CV; however, it still depends on the data

Evaluation by cross-validation: example of 3-CV

- Similar methods can be used to evaluate the classifier if we do not have any test data
- E.g., we use 3-CV to evaluate 1-NN classifier on the games-buying dataset, and the average accuracy of all 3 trials is 30%

Advanced k-NN method (1)

- Avoid having to compute distance to all objects in the training set
  - Multi-dimensional access methods (k-d trees)
  - Fast approximate similarity search
  - Locality Sensitive Hashing (LSH)
- Condensing
  - Determine a smaller set of objects that give the same performance
- Editing
  - Remove objects to improve classification accuracy

Advanced k-NN method (2)

- Assign a weigh to each individual attribute

- Assign a weight to each training instance

Outline

- Naïve Bayes classifier
- *k*-Nearest Neighbor classifier
- Summary

Three classification models

- Decision Tree
- Naïve Bayes classifier
- *k*-Nearest Neighbor classifier

Advantages of decision tree

- Relatively inexpensive to construct
- Relatively fast at classifying unknown records
- Easy to interpret for small-sized trees
- Robust to noise (especially when methods to avoid overfitting are employed)
- Can handle redundant or irrelevant attributes

Disadvantages of decision tree

- Due to the greedy nature of splitting criterion, interacting attributes (that can distinguish between classes together but not individually) may be passed over in favour of other attributed that are less discriminating.
- Each decision boundary involves only a single attribute

Advantages of Naïve Bayes

- Efficient and easy to construct
- The building process is parallelized
- Relatively fast at classifying unknown records
- Robust to isolated noise points
- Robust to irrelevant attributes

Disadvantages of Naïve Bayes

- The oversimplified assumption of features are conditionally independent with each other
  - Cannot handle redundant features
- Based on numerical computation, no rules generated – less interpretable

Advantages of k-NN

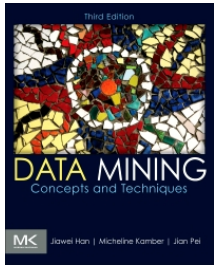
- Very simple implementation
- Few assumption on the distribution of the data
- Classifier can be updated online at very little cost as new instances with unknown classes are presented

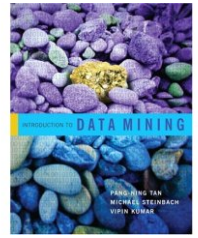
Disadvantages of k-NN

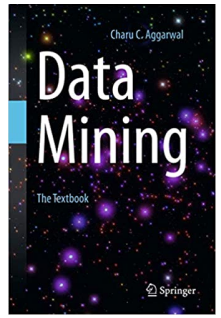
- Expensive testing of each instance
- Sensitiveness to noisy or irrelevant attributes
- Sensitiveness to imbalanced datasets, where most entities belong to one or a few classes

Recommended reading

- [Han et al., 2012]
  - Chapter 8.3
- [Tan et al., 2005]
  - Chapter 5.2, 5.3
- [Aggarwal, 2015]
  - Chapter 10.5, 10.8

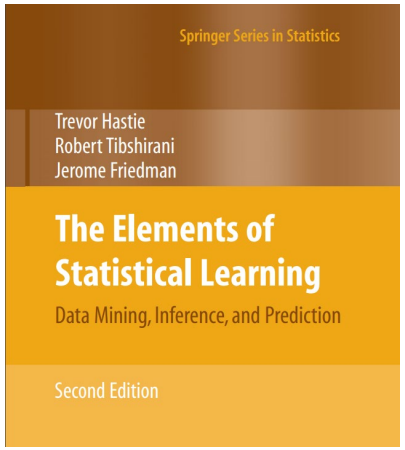






- All other references (links) within the slides

Other classification models*



- Logistic regression
  - Chapter 4.4
- Support vector machine
  - An introduction to Support Vector Machine by John Shawe-Taylor
- Neural network
  - Deep Learning by Ian Goodfellow, Yoshua Bengio and Aaron Courville
- …

Next two weeks

- Mining Text, Web, Stream, Time-series and Graph data

- Final-term revision