THE UNIVERSITY OF MELBOURNE

COMP20008 Elements of Data Processing

Semester 2 2018

Lecture 12: Classification: Decision Tree (cont.) K-Nearest Neighbor

---

THE UNIVERSITY OF MELBOURNE

Announcements

2018 Week 7, 3rd Sep – 9th Sep
- Ph. 2 release: Mon-3rd Sep 11:59 am

2018 Week 8, 10th Sep – 16th Sep
- Ph. 3 release: Mon-10th Sep 11:59 am
- Guest Lect.# 1, Prof. Richard Sinnott, Fri-14th Sep

2018 Week 9, 17th Sep – 23rd Sep
- Ph. 2 Due: Fri-21st Sep 11:59 am

2018 Week 10, 1st Oct – 7th Oct
- Guest Lect.# 2, Prof. James Bailey, Fri-5th Oct
- Ph. 3-A Due: Fri-5th Oct 11:59 am

2018 Week 11, 8th Oct – 14th Oct
- Guest Lect.# 3, Scott Thomson, Wed-10th Oct
- Ph. 3 presentations during workshops

2018 Week 12, 15th Oct – 21st Oct

THE UNIVERSITY OF MELBOURNE

Plan today

- Classification
  - Decision tree classification – finish off from last lecture
  - k nearest neighbor classification

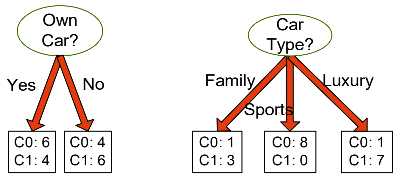
THE UNIVERSITY OF MELBOURNE

Tree Induction

- Issues
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

How to determine the Best Split

Before Splitting: 10 records of class 0,
10 records of class 1



Which test condition is the best?

How to determine the Best Split

- Greedy approach:
  - Nodes with homogeneous class distribution are preferred
- Need a measure of node impurity:

C0: 5 C1: 5	C0: 9 C1: 1
Non-homogeneous, High degree of impurity	Homogeneous, Low degree of impurity

Measures of Node Impurity

- Entropy
  - We have seen entropy in the feature correlation section, where it was used to measure the amount of uncertainty in an outcome
  - *Entropy can also be viewed as an impurity measure*
    - The set {A,B,C,A,A,A,A,A} has low entropy: low uncertainty and **high purity**
    - The set {A,B,C,D,B,E,A,F} has high entropy: high uncertainty and **low purity**

Node Impurity Criteria based on Entropy

- Entropy (H) at a given node t:

$$\boxed{H(t) = -\sum_j p(j \mid t) \log p(j \mid t)}$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

  - Measures homogeneity of a node.
    - Maximum ($\log n_c$) when records are equally distributed among all classes (n_c is number of classes)
    - Minimum (0.0) when all records belong to one class

THE UNIVERSITY OF MELBOURNE

Examples for computing Entropy

$$H(t) = -\sum_j p(j|t)\log_2 p(j|t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Entropy = – 0 $\log_2$ 0 – 1 $\log_2$ 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65

C1	**2**
C2	**4**

?

THE UNIVERSITY OF MELBOURNE

Examples for computing Entropy

$$H(t) = -\sum_j p(j|t)\log_2 p(j|t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Entropy = – 0 $\log_2$ 0 – 1 $\log_2$ 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

THE UNIVERSITY OF MELBOURNE

Question: What is entropy of this node?

$$H(t) = -\sum_j p(j|t)\log_2 p(j|t)$$

C1	**13**
C2	**20**

P(C1) = P(C2) =

Entropy =

THE UNIVERSITY OF MELBOURNE

Question: What is entropy of this node?

$$H(t) = -\sum_j p(j|t)\log_2 p(j|t)$$

C1	**13**
C2	**20**

P(C1) = $\frac{13}{33}$ P(C2) = $\frac{20}{33}$

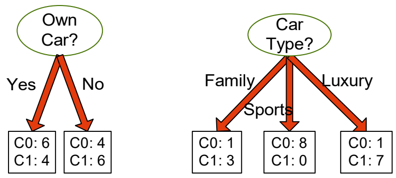
Entropy = $-(\frac{13}{33} log_2 \frac{13}{33} + \frac{20}{33} log_2 \frac{20}{33})$

How good is a Split?

- Compare the impurity (entropy) of parent node (before splitting)
- With the impurity (entropy) of the children nodes (after splitting)

$$
\begin{aligned}
Gain &= H(Parent) - H(Parent|Child) \\
&= H(Parent) - \sum_{j=1}^{k} \frac{N(v_j)}{N} H(v_j)
\end{aligned}
$$

- $H(v_j)$: impurity measure of node v_j
- j: children node index
- $N(v_j)$: number of data points in child node v_j
- N: number of data points in parent node
- The larger the gain, the better



How good is a Split?

$$
\begin{aligned}
Gain &= H(Parent) - H(Parent|Child) \\
&= H(Parent) - \sum_{j=1}^{k} \frac{N(v_j)}{N} H(v_j)
\end{aligned}
$$

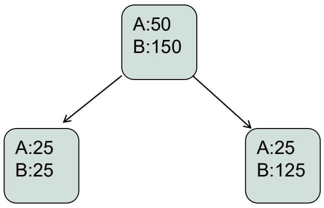
- Note: the information gain is equivalent to the mutual information between the class feature and the feature being split on
- Thus splitting using the information gain is to choose the feature with highest information shared with the class variable

Question 2c) from 2016 exam

Given a dataset with two classes, A and B, suppose the root node of a decision tree has 50 instances of class A and 150 instances of class B. Consider a candidate split of this root node into two children, the first with (25 class A and 25 class B), the second with (25 class A and 125 class B). Write a formula to measure the utility of this split using the entropy criterion. Explain how this formula helps measure split utility

Question 2c) from 2016 exam

Given a dataset with two classes, A and B, suppose the root node of a decision tree has 50 instances of class A and 150 instances of class B. Consider a candidate split of this root node into two children, the first with (25 class A and 25 class B), the second with (25 class A and 125 class B). Write a formula to measure the utility of this split using the entropy criterion. Explain how this formula helps measure split utility

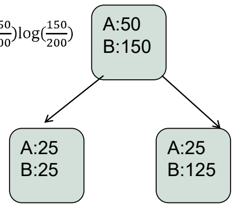


THE UNIVERSITY OF MELBOURNE

Question 2c) from 2016 exam

Given a dataset with two classes, A and B, suppose the root node of a decision tree has 50 instances of class A and 150 instances of class B. Consider a candidate split of this root node into two children, the first with (25 class A and 25 class B), the second with (25 class A and 125 class B). Write a formula to measure the utility of this split using the entropy criterion. Explain how this formula helps measure split utility

Entropy(root)= $-(\frac{50}{200})\log(\frac{50}{200}) - (\frac{150}{200})\log(\frac{150}{200})$



Entropy(left child)= $-\left(\frac{25}{50}\right)\log\left(\frac{25}{50}\right) - \left(\frac{25}{50}\right)\log\left(\frac{25}{50}\right)$

Entropy(right child)= $-\left(\frac{25}{150}\right)\log\left(\frac{25}{150}\right) - \left(\frac{125}{150}\right)\log\left(\frac{125}{150}\right)$

THE UNIVERSITY OF MELBOURNE

Question 2c) from 2016 exam

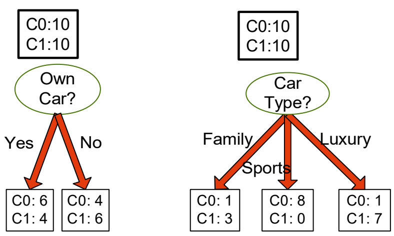
Given a dataset with two classes, A and B, suppose the root node of a decision tree has 50 instances of class A and 150 instances of class B. Consider a candidate split of this root node into two children, the first with (25 class A and 25 class B), the second with (25 class A and 125 class B). Write a formula to measure the utility of this split using the entropy criterion. Explain how this formula helps measure split utility

$$Split\ utility = Information\ Gain$$
$$= Entropy(root) - Entropy(root|split)$$
$$= Entropy(root) - [(\frac{50}{200}) * Entropy(left\ child)$$
$$+ \left(\frac{150}{200}\right) * Entropy(right\ child)]$$

THE UNIVERSITY OF MELBOURNE

How to determine the Best Split?

Before Splitting: 10 records of class 0,
10 records of class 1



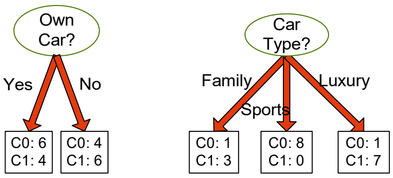
Which test condition is the best?

- Compute the gain of all splits
- Choose the one with largest gain

THE UNIVERSITY OF MELBOURNE

How to determine the Best Split?

Before Splitting: 10 records of class 0,
10 records of class 1



Own Car: Information gain=0.029

Car type: Information gain=0.62

We should choose Car type as the best split???!!!

THE UNIVERSITY OF MELBOURNE

Creating a decision tree

- Calc information gain [Left Child],[Right Child] for each of the following
  - Refund [Yes], [No]
  - Marital status [Single],[Married],[Divorced]
  - Taxable income
    - [60,60], (60,220]
    - [60,70], (70,220]
    - [60,75],(75,220]
    - [60,85],(85,220]
    - [60,90],(90,220]
    - [60,95],(95,220]
    - [60,100],(100,220]
    - [60,120],(120,220]
    - [60,125],(125,220]
- Choose feature+split with the highest information gain and use this as the root node and its split
- Do recursively, terminating when a node consists of only Cheat=No or Cheat=Yes.

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

THE UNIVERSITY OF MELBOURNE

Decision tree: advantages and disadvantages

- Advantages
  - Easy to interpret
  - Relatively efficient to construct
  - Fast for making a decision about a test instance
- Disadvantages
  - A simple greedy construction strategy, producing a set of ``If ..then'' rules. Sometimes this is too simple for data with complex structure:
    - ``Everything should be as simple as possible, but no simpler''
  - For complex datasets, the tree might grow very big and not be easy to understand
  - May behave strangely for some types of features (E.g. student ID feature from earlier slide)

THE UNIVERSITY OF MELBOURNE

Decision tree classifier: training and testing

- Divide training data into:
  - Training set (e.g. 2/3)
  - Test set (e.g. 1/3)
- Learn decision tree using the training set
- Evaluate performance of decision tree on the test set

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

Training Set

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

Test Set

Metrics for Performance Evaluation

- Can be summarized in a Confusion Matrix (contingency table)
  - Actual class: {yes, no, yes, yes, …}
  - Predicted class: {no, yes, yes, no…}

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

a: TP (true positive)
b: FN (false negative)
c: FP (false positive)
d: TN (true negative)

Metrics for Performance Evaluation

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

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

- Actual class: {yes, no, yes, yes, no, yes, no, no}
- Predicted: {no, yes, yes, no, yes, no, no, yes}

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a= 1 (TP)	b=3 (FN)
	Class=No	c=3 (FP)	d=1 (TN)

Question

- For an accurate decision tree classifier, we want to minimise both:
  - False positives (saying yes when we should say no)
  - False negatives (saying no when we should say yes)
- Describe a real scenario where it is
  - More important to minimise the false positives
  - More important to minimise the false negatives

THE UNIVERSITY OF MELBOURNE

Limitations of accuracy

- Consider a 2-class problem
  – Number of Class 0 examples = 9990
  – Number of Class 1 examples = 10

- • If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %
  – Accuracy is misleading here because model does not detect any class 1 example
  – Other metrics can be used instead of accuracy, that address this problem (but we won't cover these)

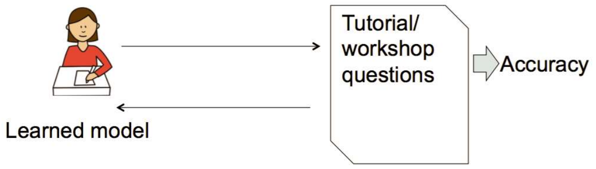
THE UNIVERSITY OF MELBOURNE

2017 exam question 3f

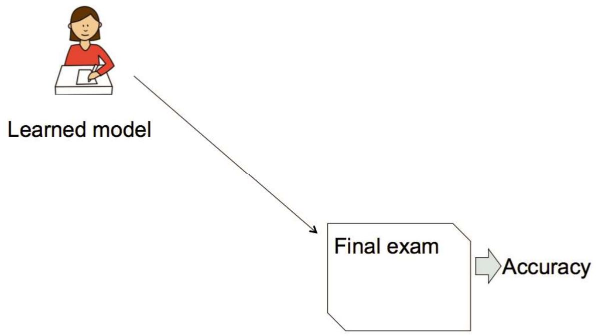
- (2 marks) What is the purpose of separating a dataset into training and test sets, when evaluating the performance of a classifier?

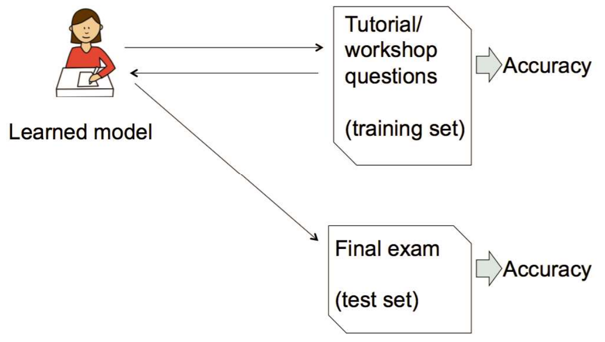
THE UNIVERSITY OF MELBOURNE

Why do we split the dataset into training and testing for evaluating accuracy?



THE UNIVERSITY OF MELBOURNE



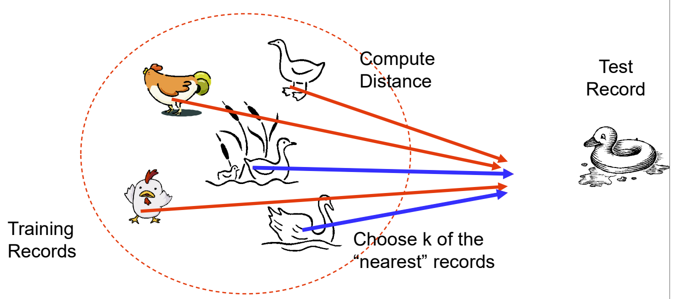


K nearest neighbor classifier

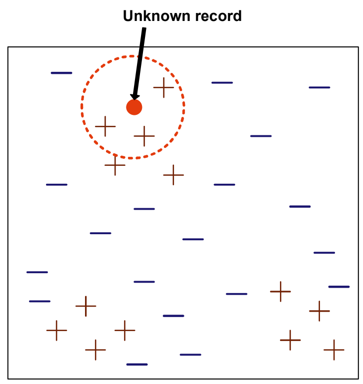
- Another widely used and intuitive algorithm for prediction

Nearest Neighbor Classifiers

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

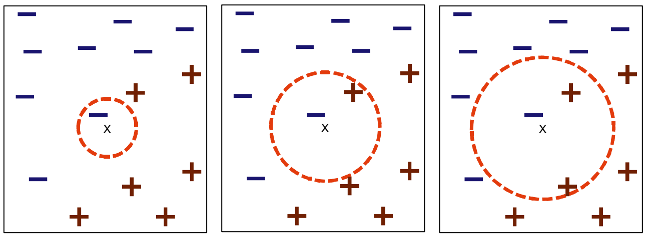


Nearest-Neighbor Classifiers



- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve
- To classify an unknown record:
  1. Compute distance to other training records
  2. Identify *k* nearest neighbors
  3. Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Definition of Nearest Neighbor



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

Distance measure

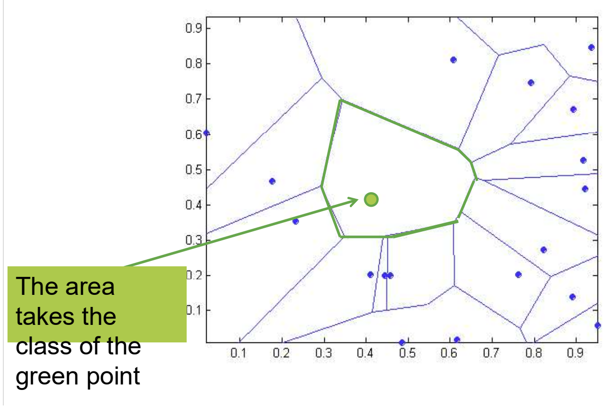
- Compute distance between two points $p=(p_1,p_2,\ldots)$, $q=(q_1,q_2,\ldots)$
  - Euclidean distance

$$d(p,q)=\sqrt{\sum_i (p_i-q_i)^2}$$

  - Can also use Pearson coefficient (similarity measure)
- Determine the class from nearest neighbor list
  - take the majority vote of class labels among the k-nearest neighbors
  - Or weight the vote according to distance
    - weight factor, $w = \frac{1}{d^2}$

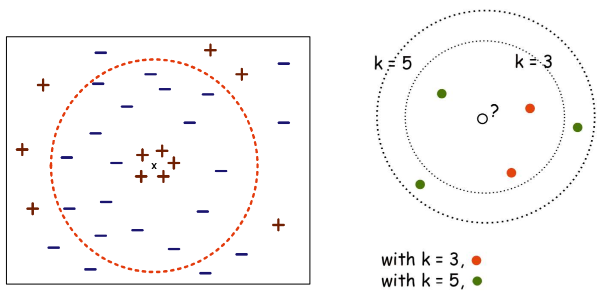
1 nearest-neighbor

Voronoi Diagram defines the classification boundary



K- Nearest Neighbor classifier

- Choosing the value of k:
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include points from other classes



http://2.bp.blogspot.com/-EK4tA2525EM/U-c6Q4jJuwI/AAAAAAAADjw/hdqRXuunpnQ/s1600/knn.png

THE UNIVERSITY OF MELBOURNE

Points to remember from this lecture

- Understand the use of accuracy as a metric for measuring the performance of a classification method.
- Understand how TP,TN,FP and FN are used in the accuracy calculation. The formula for accuracy will be provided on the exam
- understand the operation and rationale of the k nearest neighbor algorithm for classification
- understand the advantages and disadvantages of using k nearest neighbor or decision tree for classification

THE UNIVERSITY OF MELBOURNE

References and Acknowledgement

This lecture was prepared using some material adapted from:

- https://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf
- CS059 - Data Mining -- Slides
- http://www-users.cs.umn.edu/~kumar/dmbook/dmslides/chap4_basic_classification.ppt