Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Thomas Seidl
Janina Sontheim, Maximilian Hünemörder

Knowledge Discovery in Databases

WS 2019/20

Exercise 5: Decision Trees, Nearest Neighbor Classifier

Exercise 5-1 Decision Trees

Predict the risk class of a car driver based on the following attributes:

Attribute	Description	Values
time	time since obtaining a drivers license in years	{1-2, 2-7, >7}
gender	gender	{male, female}
area	residential area	{urban, rural}
risk	the risk class	{low, high}

For your analysis you have the following manually classified training examples:

ID	time	gender	area	risk
1	1-2	m	urban	low
2	2-7	m	rural	high
3	>7	f	rural	low
4	1-2	f	rural	high
5	>7	m	rural	high
6	1-2	m	rural	high
7	2-7	f	urban	low
8	2-7	m	urban	low

(a) Construct a decision tree based on this training data. For splitting, use information gain as measure for impurity. Build a separate branch for each attribute. The decision tree shall stop when all instances in the branch have the same class, you do not need to apply a pruning algorithm.

Reminder: When splitting T by attribute A into partitions $T_1, \ldots, T_m$, we have

$$entropy(T) = -\sum_{i=1}^{k} p_i \cdot \log p_i$$

$$IG(T, A) = entropy(T) - \sum_{i=1}^{m} \frac{|T_i|}{|T|} entropy(T_i)$$

As $entropy(T)$ is fixed for a given T, independent of the splitting attribute A, maximising $IG(T, A)$ is equivalent to minimising

$$S = \sum_{i=1}^{m} \frac{|T_i|}{|T|} entropy(T_i)$$

Splits

ID	**time**	**risk**	**gender**	**risk**	**area**	**risk**
1	a1-2	alow	am	alow	aurban	alow
2	b2-7	bhigh	am	ahigh	brural	bhigh
3	c>7	clow	bf	blow	brural	blow
4	a1-2	ahigh	bf	bhigh	brural	bhigh
5	c>7	chigh	am	ahigh	brural	bhigh
6	a1-2	ahigh	am	ahigh	brural	bhigh
7	b2-7	blow	bf	blow	aurban	alow
8	b2-7	blow	am	alow	aurban	alow

Time

time	$\lvert T_i \rvert$	risk	p_i	$\approx entropy(T_i)$
1-2	3	low high	1/3 2/3	0.918
2-7	3	low high	2/3 1/3	0.918
>7	2	low high	1/2 1/2	1

$$S \approx \frac{3}{8} \cdot 0.918 + \frac{3}{8} \cdot 0.918 + \frac{2}{8} \cdot 1 \approx 0.94$$

Gender

gender	$\lvert T_i \rvert$	risk	p_i	$\approx entropy(T_i)$
m	5	low high	2/5 3/5	0.971
f	3	low high	2/3 1/3	0.918

$$S \approx \frac{5}{8} \cdot 0.971 + \frac{3}{8} \cdot 0.918 \approx 0.95$$

Area

area	$\lvert T_i \rvert$	risk	p_i	$\approx entropy(T_i)$
rural	5	low high	1/5 4/5	0.722
urban	3	low high	3/3 0/3	0

$$S \approx \frac{5}{8} \cdot 0.722 + \frac{3}{8} \cdot 0 \approx 0.45$$

Decision As area yields the lowest S and hence, the highest information gain, it is chosen for split. The branch for $area = urban$ is already pure, and hence not further processed.

Splits The second branch contains the following data

ID	**time**	**risk**	**gender**	**risk**
2	b2-7	bhigh	am	ahigh
3	c>7	clow	bf	blow
4	a1-2	ahigh	bf	bhigh
5	c>7	chigh	am	ahigh
6	a1-2	ahigh	am	ahigh

Time

time	$\lvert T_i \rvert$	risk	p_i	$\approx entropy(T_i)$
1-2	2	low high	0/2 2/2	0
2-7	1	low high	0/1 1/1	0
>7	2	low high	1/2 1/2	1

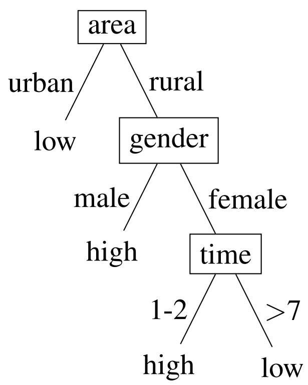
$$S \approx \frac{2}{5} \cdot 0 + \frac{1}{5} \cdot 0 + \frac{2}{5} \cdot 1 = 0.4$$

Gender

gender	$\lvert T_i \rvert$	risk	p_i	$\approx entropy(T_i)$
m	3	low high	0/3 3/3	0
f	2	low high	1/2 1/2	1

$$S \approx \frac{3}{5} \cdot 0 + \frac{2}{5} \cdot 1 = 0.4$$

Decision Choose arbitrary, here gender. There remains only a single non-pure branch, `female`, which can be split using `time`. The final tree is given by



(b) Apply the decision tree to the following drivers:

ID	time	gender	area
A	1-2	f	rural
B	2-7	m	urban
C	1-2	f	urban

The following table shows the classification, and highlights attributes that contributed to the decision.

ID	time	gender	area	risk
A	a1-2	af	arural	high
B	2-7	m	aurban	low
C	1-2	f	aurban	low

Exercise 5-2 Information gain

In this exercise, we want to look more closely at the information gain measure.

Let T be a set of n training objects with the attributes $A_1, \ldots, A_a$ and the k classes c_1 to c_k.

Let $\{T_i^A \,|\, i \in \{1, \ldots, m_A\}\}$ be the disjoint, complete partitioning of T produced by a split on attribute A (where m_A is the number of disjoint values of A).

(a) *Uniform distribution*
Compute *entropy*(T), *entropy*(T_i^A) for $i \in \{1 \ldots m_A\}$ as well as *information-gain*(T, A) given the assumption that the class membership of T is uniformly distributed and independent of the values of A. Interpret your result!

independent uniform distribution:

$$\begin{aligned}
p_i &= \frac{1}{k} \forall 1 \leq i \leq k \\
|T_i^A| &= \frac{1}{m_A} \cdot |T| \\
\textit{entropy}(T) &= -\sum_{i=1}^{k} p_i \log p_i \\
&= -k \cdot \frac{1}{k} \cdot \log \cdot \frac{1}{k} \\
&= -\log \frac{1}{k} \\
&= \log k \\
\textit{entropy}(T_i^A) &= \log k \textit{ (analogously)} \\
\textit{information-gain}(T, A) &= \textit{entropy}(T) - \sum_{i=1}^{m_A} \frac{|T_i^A|}{|T|} \cdot \textit{entropy}(T_i^A) \\
&= \log k - m_A \cdot \frac{1}{m_A} \cdot \log k \\
&= 0
\end{aligned}$$

Interpretation: The split leads to no gain of information. This result is intuitive, a split on such an attribute provides no benefit.

(b) *Attributes with many values*

Let A be an attribute with random values, not correlated to the class of the objects. Furthermore, let A have enough values, such than no two instances of the training set share the same value of A. What happens in this situation when building the decision tree? What is problematic with this situation?

In this case, a split on A leads to maximally pure child nodes (i.e., $p_i = 1$ for a single i and$p_j = 0$ for all $j \neq i$), since each node contains only a single sample. As a result, each node will have zero entropy such that

$$\textit{information-gain}(T, A) = \textit{entropy}(T) - 0$$

is maximal. Thus, A will be chosen as split attribute at the root and the tree is completed.

Problem: The tree achieves (optimal) zero training error but grotesquely overfits. In fact, it is useless since no generalization occurred and the tree simply memorized the training data. A large error can be expected if the tree is applied to new test data unseen during training.

Such a situation might occur if the sample size considered for a split is very small, for instance when dealing with a very small training dataset or when splitting a node deep within a tree. A possible solution for the latter case might be to perform pre-pruning, e.g. by requiring a minimum number of samples for a split.

Exercise 5-3 Nearest neighbor classification

The 2D feature vectors in the figure below belong to two different classes (circles and rectangles). Classify the object at $(6, 6)$ — in the image represented using a triangle — using k nearest neighbor classification. Use Manhattan distance (L_1 norm) as distance function, and use the non-weighted class counts in the k-nearest-neighbor set, i.e. the object is assigned to the majority class within the k nearest neighbors. Perform kNN classification for the following values of k and compare the results with your own "intuitive" result.

(a) $k = 4$

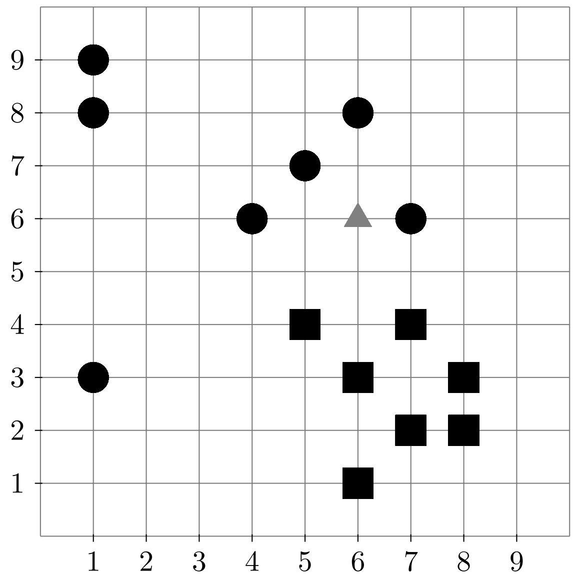
The 4 nearest neighbors are all circles, such that the object would also be classified as a circle. This seems intuitive, since the object is located within the circle cluster.

(b) $k = 7$

The 7 nearest neighbors additionally contain 3 rectangles in addition to the 4 circles. Since the circles are still in the majority, the object would still be classified as a circle. However, the decision is less confident than before.

(c) $k = 10$

The 10 nearest neighbors consist of 4 circles and 6 rectangles. Now the majority vote decides for the rectangle class. The reason is that the algorithm observes a larger neighborhood and that the rectangle class within that neighborhood is larger. In some applications it makes sense to search for patterns on a larger scale, since smaller classes might also be regarded as noise.


9
8
7
6
5
4
3
2
1
1 2 3 4 5 6 7 8 9