Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Thomas Seidl
Max Berrendorf, Julian Busch

Knowledge Discovery and Data Mining I

WS 2018/19

Exercise 12: Decision Trees, Nearest Neighbor Classifier, Regression Trees

QA Session

The latter part of the last lecture slot on February 5th will be dedicated to a QA-session which is intended to give you an opportunity to ask questions about the lecture and exercise contents and also benefit from the discussions of other students' questions. Note that we cannot answer any specific questions regarding exam contents. During the session, we will discuss the questions, which you send to us via e-mail in advance (`berrendorf@dbs.ifi.lmu.de`). Please hand in your questions before February 4th, 12:00, so we have some time to prepare them. Note that in contrast to the lecture, the QA session will not be recorded.

Exercise 12-1 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

$$\begin{aligned} \mathit{entropy}(T) &= -\sum_{i=1}^{k} p_i \cdot \log p_i \\ IG(T, A) &= \mathit{entropy}(T) - \sum_{i=1}^{m} \frac{|T_i|}{|T|} \mathit{entropy}(T_i) \end{aligned}$$

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

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

Splits

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

Time

time	$\|T_i\|$	risk	p_i	$\approx \mathit{entropy}(T_i)$
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	$\|T_i\|$	risk	p_i	$\approx \mathit{entropy}(T_i)$
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	$\|T_i\|$	risk	p_i	$\approx entropy(T_i)$
rural	5	low	1/5	0.722
		high	4/5	
urban	3	low	3/3	0
		high	0/3	

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

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	2-7	high	m	high
3	>7	low	f	low
4	1-2	high	f	high
5	>7	high	m	high
6	1-2	high	m	high

Time

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

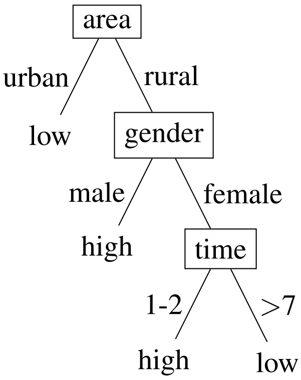
$$S \approx \frac{2}{5} \cdot 0 + \frac{1}{5} \cdot 0 + \frac{2}{5} \cdot 1 = 0.4$$

Gender

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

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

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	1-2	f	rural	high
B	2-7	m	urban	low
C	1-2	f	urban	low

Exercise 12-2 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 12-3 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) $k = 4$

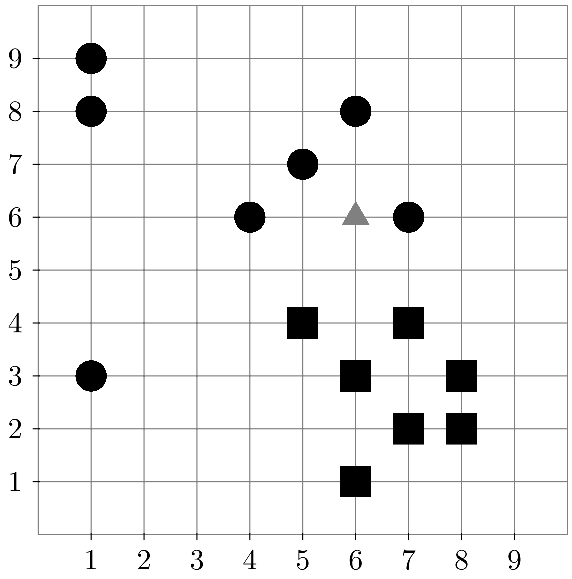
The 4 nearest neighbors are all circles, such that the object would also be classified as a circle. This seems intuitive, since the object is located within the circle cluster.

(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.



Exercise 12-4 Regression Trees

Consider the following data samples of the form (x, y), where the input value is $x \in R$ and the output value is $y \in R$:

$$p_1 = (-3, -1), p_2 = (-2, 0), p_3 = (-1, 1), p_4 = (1, 1), p_5 = (2, 0), p_6 = (3, -1)$$

Search for the first best split. If the decision is obvious, you don't have to compute all possible splits. Then decide whether the split is significant or not by using the impurity ratio with $\tau_0 = 0.5$.

Initially, the root of the regression tree contains all points in the dataset, i.e., $T = \{p_1, \dots, p_6\}$. In order to determine whether a split is significant, we need to compute the impurity of T. For the second sub-taks, we need to fit an optimal regression line which minimizes the SSE loss. The closed form solution of this line is given as

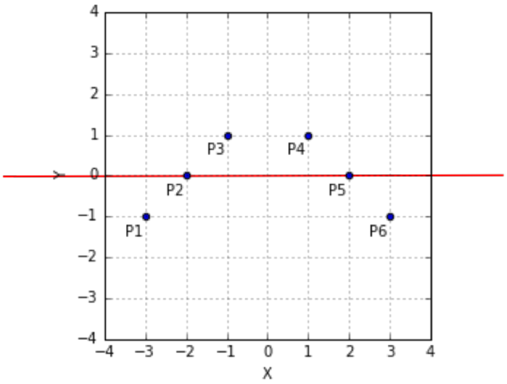
$$\beta_1 = \frac{Cov(x, y)}{Var(x)}, \qquad \beta_0 = \bar{y} - \beta_1 \bar{x}.$$

From the plot it is directly obvious that $\bar{x} = \bar{y} = 0$ and thus

$$\begin{aligned} Cov(x, y) &= \sum_{(x,y) \in T} (x - \bar{x})(y - \bar{y}) \\ &= \sum_{(x,y) \in T} xy \\ &= 3 + 0 - 1 + 1 + 0 - 3 \\ &= 0 \end{aligned}$$

The optimal regression line coefficients can then be computed as $\beta_1 = \beta_0 = 0$. Since the optimal regression line is constant, we get the same impurity for the optimal constant regression line in the first sub-task:

$$imp(T) = \frac{1}{6}(1+0+1+1+0+1) = \frac{2}{3}$$

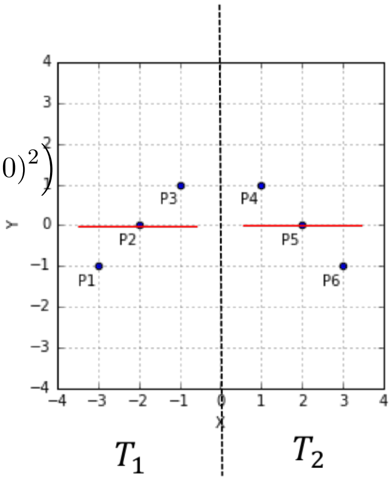


(a) Fit constant functions and use the variance of the residuals as impurity measure. Note that an optimal constant regression function always predicts the mean output value over all training samples, such that in this case, the variance of the residuals corresponds to the variance of the outputs.

While trying all possible splits and determining the split with the smallest summed impurity, we will rely on symmetry. In total, three different cases need to be considered based on how many points are split from the rest of the dataset:

Split in half. We get $\bar{y_{T_1}} = \bar{y_{T_2}} = 0$ and thus

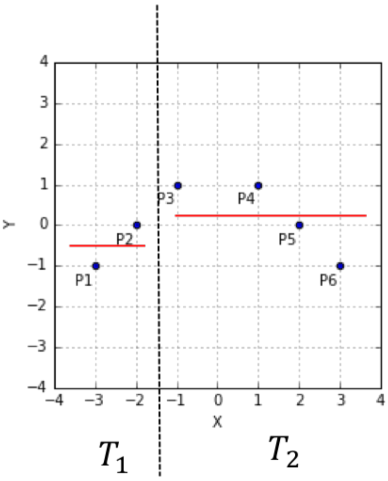
$$\begin{aligned} imp(T_1) = imp(T_2) &= \frac{1}{3}\left((1-0)^2 + (0-0)^2 + (1-0)^2\right) \\ &= \frac{2}{3} \\ \implies imp(T_1) + imp(T_2) &= \frac{4}{3} \end{aligned}$$



Split two points. For the left hand side, we get $\bar{y_{T_1}} = \frac{1}{2}(-1+0) = -\frac{1}{2}$ and

$$imp(T_1) = \frac{1}{2}\left(\left(-1+\frac{1}{2}\right)^2 + \left(0+\frac{1}{2}\right)^2\right) = \frac{1}{4}$$

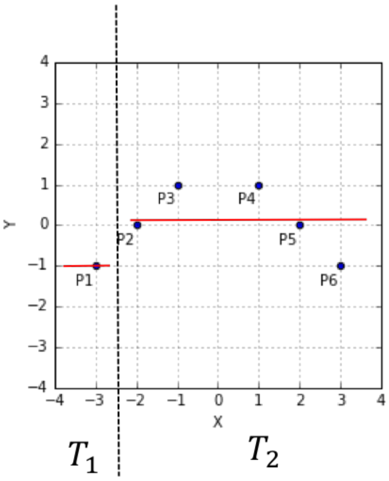
For the right hand side, we get $\bar{y_{T_2}} = \frac{1}{4}(1+1+0-1) = \frac{1}{4}$ and



$$imp(T_2) = \frac{1}{4}\left(\left(1-\frac{1}{4}\right)^2 + \left(1-\frac{1}{4}\right)^2 + \left(0-\frac{1}{4}\right)^2 + \left(-1-\frac{1}{4}\right)^2\right) = \frac{11}{16}$$

Since $imp(T_1) + imp(T_2) = \frac{15}{16} < \frac{4}{3}$, this split is better than the previous one.

Split one point. Since the left child node consists of only a single point, it can be fit perfectly such that $imp(T_1) = 0$. For the right hand side, we get $\bar{y_{T_2}} = \frac{1}{5}(0+1+1+0-1) = \frac{1}{5}$ and



$$imp(T_2) = \frac{1}{5}\left(\left(0-\frac{1}{5}\right)^2 + \left(1-\frac{1}{5}\right)^2 + \left(1-\frac{1}{5}\right)^2 + \left(0-\frac{1}{5}\right)^2 + \left(-1-\frac{1}{5}\right)^2\right) = \frac{14}{25}$$

Since $imp(T_1) + imp(T_2) = \frac{14}{25} < \frac{15}{16}$, this split is even better than the previous one. However, it is not significant since

$$\frac{imp(T_1) + imp(T_2)}{imp(T)} = \frac{14}{25} \cdot \frac{3}{2} = \frac{21}{25} > \tau_0 = \frac{1}{2}$$

In order to avoid overfitting, the split would not be performed, since the accuracy gain is deemed to small to justify the additional specialization of the model.

(b) Fit linear functions and use the variance of the residuals as impurity measure.

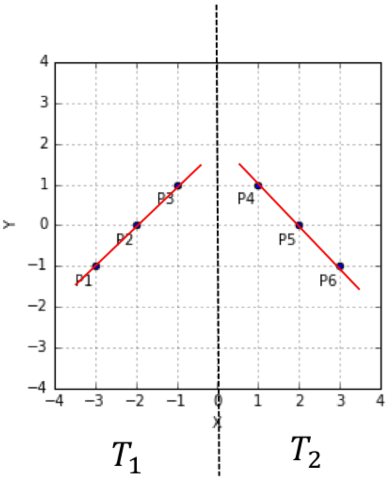
If we simply split the dataset in half, the points in both, T_1 and T_2, are collinear such that we can fit regression lines with no error, i.e.,

$$imp(T_1) + imp(T_2) = 0 + 0 = 0$$

Since the impurity is always non-negative, there cannot exist a better split. The split is further significant since

$$\frac{imp(T_1) + imp(T_2)}{imp(T)} = 0 < \tau_0 = \frac{1}{2}$$

Note that this holds for any positive threshold τ, i.e., such a perfect split would always be performed, irrespective of the chosen threshold.



4
3
2
1
0
−1
−2
−3
−4
Y
−4
−3
−2
−1
0
1
2
3
4
X
P1
P2
P3
P4
P5
P6
T_1
T_2