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 10: Classification Evaluation, m-fold Cross Validation, Naïve Bayes Classifier

Exercise 10-1 Evaluation of classifiers

Given a data set $D = \{o_1, \ldots, o_n\}$ with known class labels $C(o_i) \in \mathcal{C} = \{A, B, C\}$ of the objects. In order to evaluate the quality of a classifier K, each object $o_i \in D$ is additionally classified using K, yielding class label $K(o_i)$. The results are given in the table below.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$C(o_i)$	A	B	A	C	C	B	A	A	A	B	B	C	C	C	B
$K(o_i)$	A	A	C	C	B	B	A	A	A	C	A	A	C	C	B

(a) Setup the confusion matrix.

Although the confusion matrix is only the middle part of the below-standing table, towards the subsequent exercises we already compute row and column sums, as well as the total sum.

	A	B	C	C_i
A	4	0	1	5
B	2	2	1	5
C	1	1	3	5
K_i	7	3	5	15

For brevity, we will denote with CM_{ij} the entry of the confusion matrix at position (i, j), where $i \in \mathcal{C}$ corresponds to the true class, and $j \in \mathcal{C}$ corresponds to the predicted class.

(b) Compute the accuracy / classification error.

Accuracy

	A	B	C	C_i
A	**4**	0	1	5
B	2	**2**	1	5
C	1	1	**3**	5
K_i	7	3	5	**15**

Recall

	A	B	C	C_i
A	**4**	0	1	**5**
B	2	2	1	5
C	1	1	3	5
K_i	7	3	5	15

Precision

	A	B	C	C_i
A	**4**	0	1	5
B	2	2	1	5
C	1	1	3	5
K_i	**7**	3	5	15

Figure: Summary of evaluation measures directly obtainable from confusion matrix. Red entries go to the nominator, blue ones to the denominator. If there are several red ones, their sum is taken.

The accuracy is given by

$$\text{Accuracy}_D(K) = \frac{|\{o \in D \mid K(o) = C(o)\}|}{|D|}$$

The nominator of the above mentioned fraction can be obtained as the sum of diagonal entries of the confusion matrix, i.e.

$$|\{o \in D \mid K(o) = C(o)\}| = \sum_{i \in \mathcal{C}} CM_{ii} = 4 + 2 + 3 = 9$$

Hence, we obtain $\text{Accuracy}_D(K) = \frac{9}{15} = \frac{3}{5} = 60\%$. The classification error $\text{Error}_D(K)$ can be obtained exploiting the relationship $\text{Error}_D(K) = 1 - \text{Accuracy}_D(K) = 40\%$.

(c) For each class $i \in \mathcal{C}$ compute precision and recall.

The recall is defined as

$$\text{Recall}_D(K, i) = \frac{|\{o \in C_i \mid K(o) = C(o)\}|}{|C_i|}$$

In the confusion matrix, this can be read off by dividing the diagonal entry by the row-sum:

$$\text{Recall}_D(K, i) = \frac{CM_{ii}}{C_i}$$

Similarly, the precision is defined as

$$\text{Precision}_D(K, i) = \frac{|\{o \in C_i \mid K(o) = C(o)\}|}{|K_i|}$$

In the confusion matrix, this can be read off by dividing the diagonal entry by the column-sum:

$$\text{Precision}_D(K, i) = \frac{CM_{ii}}{K_i}$$

Thus, we obtain

class	**recall**	**precision**
A	4/5	4/7
B	2/5	2/3
C	3/5	3/5

(d) To get a complete measure for the quality of the classification with respect to a single class, the F_1-measure (the harmonic mean of precision and recall) is commonly used. It is defined as follows:

$$F_1(K, i) = \frac{2 \cdot \text{Recall}(K, i) \cdot \text{Precision}(K, i)}{\text{Recall}(K, i) + \text{Precision}(K, i)}$$

Compute the F_1-measure for all classes.

Note: "F_1-measure" may refer to the same formula but computed using a different precision and different recall in other applications. It is a specialization of F_β with equal weighting of precision and recall.

Using the computation from the confusion matrix of recall and precision, we obtain

$$\begin{aligned} F_1(K,i) &= \frac{2 \cdot \text{Recall}(K,i) \cdot \text{Precision}(K,i)}{\text{Recall}(K,i) + \text{Precision}(K,i)} \\ &= \frac{2 \cdot \frac{CM_{ii}}{C_i} \cdot \frac{CM_{ii}}{K_i}}{\frac{CM_{ii}}{C_i} + \frac{CM_{ii}}{K_i}} \\ &= \frac{\frac{2CM_{ii}^2}{C_iK_i}}{\frac{K_iCM_{ii}+C_iCM_{ii}}{C_iK_i}} \\ &= \frac{2CM_{ii}^2}{K_iCM_{ii} + C_iCM_{ii}} \\ &= \frac{2CM_{ii}^2}{(K_i + C_i)CM_{ii}} \\ &= \frac{2CM_{ii}}{K_i + C_i} \end{aligned}$$

and hence,

$$\begin{aligned} F_1(K,A) &= \frac{2 \cdot 4}{5+7} = \frac{8}{12} = \frac{2}{3} \\ F_1(K,B) &= \frac{2 \cdot 2}{5+3} = \frac{4}{8} = \frac{1}{2} \\ F_1(K,C) &= \frac{2 \cdot 3}{5+5} = \frac{6}{10} = \frac{3}{5} \end{aligned}$$

(e) So far, the F_1-measure is only defined for classes and not yet useful to get an overview of the overall performance of the classifiers. For this, one commonly takes the average over all classes using one of the following two approaches:

(i) **Micro Average F_1-Measure**: The values of TP, FP and FN are added up over all classes. Then precision, recall and F_1-measure are computed using these sums.

(ii) **Macro Average F_1-Measure**: Precision and recall are computed for each class individually, afterwards the average precision and average recall are used to compute the F_1-measure.

Compute the Micro- and Macro-Average F_1-measures for the example above. What do you observe?

(i) **Micro Average F_1:**

- $|TP| = 4 + 2 + 3 = 9$
- $|FP| = 3 + 1 + 2 = 6$
- $|FN| = 1 + 3 + 2 = 6$

Precision: $9/15$, Recall: $9/15$, **Micro Average F_1:** $9/15 = 3/5 = 60\%$.

Observation: The recall for TP and FN over all classes is not a particular sensible value, and the same as precision (since FP in columns without the diagonal and FN in rows without diagonal add up to the same):

$$\begin{aligned} \text{micro precisionrecall}(K) &= \frac{|\{o \in D \mid K(o) = C(o)\}|}{|D|} \\ &= \frac{\sum_i |TP_i|}{|D|} \\ &= \text{Accuracy}(K) \end{aligned}$$

Micro F_1 is in terms of information retrieval the same as accuracy in machine learning:

$$\begin{aligned} \text{micro}F_1 &= \frac{\frac{2\cdot\sum_i |TP_i|\cdot\sum_i |TP_i|}{|D|\cdot|D|}}{2\cdot\frac{\sum_i |TP_i|}{|D|}} \\ &= \frac{2\cdot\sum_i |TP_i|\cdot\sum_i |TP_i|\cdot|D|}{2\cdot\sum_i |TP_i|\cdot|D|\cdot|D|} \\ &= \frac{\sum_i |TP_i|}{|D|} \end{aligned}$$

Should be obvious: the harmonic mean of accuracy and accuracy is accuracy.

(ii) **Macro Average F_1:**

- average precision: $1/3(4/7 + 2/3 + 3/5) \approx 0.613$
- average recall: $1/3(4/5 + 2/5 + 3/5) = 0.6$

Macro Average $F_1 \approx \frac{2\cdot 0.6\cdot 0.613}{0.6+0.613} = 0.606$.
The key difference between Micro- and Macro-Average is that in Macro Average all classes have the same weight. In Micro-Average, the classes are weighted by their size, so classes with a small size play a very small role. With Macro-Average, small classes are as important as large classes, which is often a desired property.

Example: an illness affects only 1% of the population. Always predicting "healthy" is correct in 99% of the cases, but the remaining 1% of error is maybe worse than errors in the healthy 99%.

Exercise 10-2 m-fold Cross Validation

Suppose, you have a 2-dimensional dataset consisting of 5 classes with 90 objects each, arranged as follows

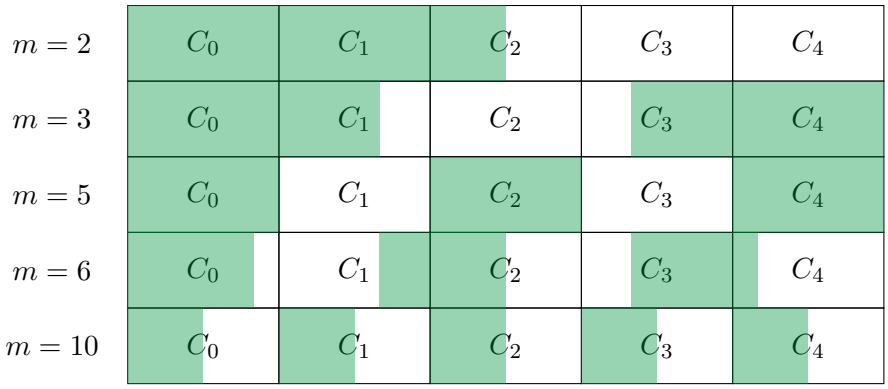
$$\overbrace{x_0,\ldots,x_{89}}^{C(x)=0},\overbrace{x_{90},\ldots,x_{179}}^{C(x)=1},\ldots,\overbrace{x_{360},\ldots,x_{449}}^{C(x)=4},$$

and that the classes are linearly separable (i.e. can be separated using a hyperplane). Suppose further, that someone has produced a poor implementation of the m-fold cross validation procedure and applied it in combination with a multi-class linear classifier to obtain the following results:

m	2	3	5	6	10
accuracy	20%	40%	0%	100%	100%

What is the problem with the implementation of the m-fold cross validation? Describe and explain the result for each value of m in short and precise sentences. How could the implementation be improved?

Observations The classes are linearly separable. If we have enough samples from every class in the training set, we can, in principle, train a multi-class linear classifier with no error. Thus, we could expect (almost) perfect accuracy. On the other hand, if for one class no samples are in the training set, we cannot classify any object of that class correctly.



Problem with the Implementation The folds are constructed by simply cutting the data into consecutive blocks. This is problematic, since the data is sorted, as we will see in the following. The following visualisation shows the coverage of the classes by the m folds.

Case $m = 2$: Suppose, we use the first fold for training. Then, the last two classes are not represented in the training data. Thus, at least $4/5$ of the test samples are misclassified. On the other hand, half of the samples of class C_3 are in the training set If we assume, that all test samples of class C_3 are classified correctly, we arrive at the observed accuracy of $1/5 = 20\%$. By symmetry we obtain the same results, if we use the second fold for training.

Case $m = 3$: Each fold consists of 5/3 blocks. Suppose, we use the first two folds for training By the same reasoning as for $m = 2$: 3/5 of the test sample are misclassified; $2/5 = 40\%$ of the test samples can be classified correctly. Again by symmetry, we obtain the same results if we use any of the other folds for testing.

Case $m = 5$: Now each fold corresponds to exactly one class. The class that is used for testing is not represented in the training data. Thus, all test samples are misclassified and we get an accuracy of 0%.

Case $m = 6, 10$: Now m is large enough, such that a fold can never contain all samples from a certain class. Thus, all classes are represented in the training set and we can observe an accuracy of 100%.

Improved Implementation At least, all classes that appear in the dataset should always be represented in the training data.

It is further reasonable, to construct training and test sets, such that the class distributions in both sets represent the class distribution in the whole dataset. This can be achieved by performing *stratified sampling*:

(a) Divide each class (*stratum*) separately into m chunks, either deterministically or by random sampling.

(b) Construct a fold for the m-fold cross-validation by taking a chunk from each class and combining them.

Exercise 10-3 Naive Bayes

The skiing season is open. To reliably decide when to go skiing and when not, you could use a classifier such as Naive Bayes. The classifier will be trained with your observations from the last year. Your notes include the following attributes:

- The weather: The attribute `weather` can have the following three values: sunny, rainy and snow.

- The snow level: The attribute `snow level` can have the following two values: ≥50 (There are at least 50 cm of snow) and <50 (There are less than 50 cm of snow).

Assume you wanted to go skiing 8 times during the previous year. Here is the table with your decisions:

weather	sunny	rainy	rainy	snow	snow	sunny	snow	rainy
snow level	<50	<50	≥50	≥50	<50	≥50	≥50	<50
ski?	no	no	no	yes	no	yes	yes	yes

(a) Compute the *a priori* probabilities for both classes `ski = yes` and `ski = no` (on the training set)!

$$\begin{aligned} P(ski) &= 0.5 \\ P(\neg ski) &= 0.5 \end{aligned}$$

(b) Compute the conditional distributions for the two classes for each attribute.

	weather			**snow**	
	sunny	rainy	snow	< 50	≥ 50
ski	1/4	1/4	2/4	1/4	3/4
$\neg ski$	1/4	2/4	1/4	3/4	1/4

(c) Decide for the following weather and snow conditions, whether to go skiing or not! Use the Naive Bayes classificator for finding the decision.

day	**weather**	**snow level**
A	sunny	≥50
B	rainy	<50
C	snow	<50

Day A

$$\begin{aligned} & P(ski|weather = sunny, snow \geq 50) \\ = & \frac{P(weather = sunny|ski) \cdot P(snow \geq 50|ski) \cdot P(ski)}{P(weather = sunny, snow \geq 50)} \\ = & \frac{\frac{1}{4} \cdot \frac{3}{4} \cdot \frac{1}{2}}{P(weather = sunny, snow \geq 50)} = \frac{\frac{3}{32}}{P(weather = sunny, snow \geq 50)} \end{aligned}$$

$$\begin{aligned} & P(\neg ski|weather = sunny, snow \geq 50) \\ = & \frac{P(weather = sunny|\neg ski) \cdot P(snow \geq 50|\neg ski) \cdot P(\neg ski)}{P(weather = sunny, snow \geq 50)} \\ = & \frac{\frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{2}}{P(weather = sunny, snow \geq 50)} = \frac{\frac{1}{32}}{P(weather = sunny, snow \geq 50)} \end{aligned}$$

$\Rightarrow$ Ski

Day B

$$
\begin{aligned}
&P(ski|weather = rainy, snow < 50) \\
&= \frac{P(weather = rainy|ski) \cdot P(snow < 50|ski) \cdot P(ski)}{P(weather = rainy, snow < 50)} \\
&= \frac{\frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{2}}{P(weather = rainy, snow < 50)} = \frac{\frac{1}{32}}{P(weather = rainy, snow < 50)}
\end{aligned}
$$

$$
\begin{aligned}
&P(\neg ski|weather = rainy, snow < 50) \\
&= \frac{P(weather = rainy|\neg ski) \cdot P(snow < 50|\neg ski) \cdot P(\neg ski)}{P(weather = rainy, snow < 50)} \\
&= \frac{\frac{2}{4} \cdot \frac{3}{4} \cdot \frac{1}{2}}{P(weather = rainy, snow < 50)} = \frac{\frac{6}{32}}{P(weather = rainy, snow < 50)}
\end{aligned}
$$

$\Rightarrow$ do not ski

Day C

$$
\begin{aligned}
&P(ski|weather = snow, snow < 50) \\
&= \frac{P(weather = snow|ski) \cdot P(snow < 50|ski) \cdot P(ski)}{P(weather = snow, snow < 50)} \\
&= \frac{\frac{2}{4} \cdot \frac{1}{4} \cdot \frac{1}{2}}{P(weather = snow, snow < 50)} = \frac{\frac{2}{32}}{P(weather = snow, snow < 50)}
\end{aligned}
$$

$$
\begin{aligned}
&P(\neg ski|weather = snow, snow < 50) \\
&= \frac{P(weather = snow|\neg ski) \cdot P(snow < 50|\neg ski) \cdot P(\neg ski)}{P(weather = snow, snow < 50)} \\
&= \frac{\frac{1}{4} \cdot \frac{3}{4} \cdot \frac{1}{2}}{P(weather = snow, snow < 50)} = \frac{\frac{3}{32}}{P(weather = snow, snow < 50)}
\end{aligned}
$$

$\Rightarrow$ do not ski