Model evaluation

- qualitative – following the definition of data mining (Piatetski-Shapiro, Fayaad, 90th):
  how new, interesting, useful and understandable the model is

  (not) corresponding to expectations (common sense), to knowledge of an expert
- quantitative

Model evaluation

- qualitative – following the definition of data mining (Piatetski-Shapiro, Fayaad, 90th):

  how new, interesting, useful and understandable the model is

  (not) corresponding to expectations (common sense), to knowledge of an expert
- quantitative

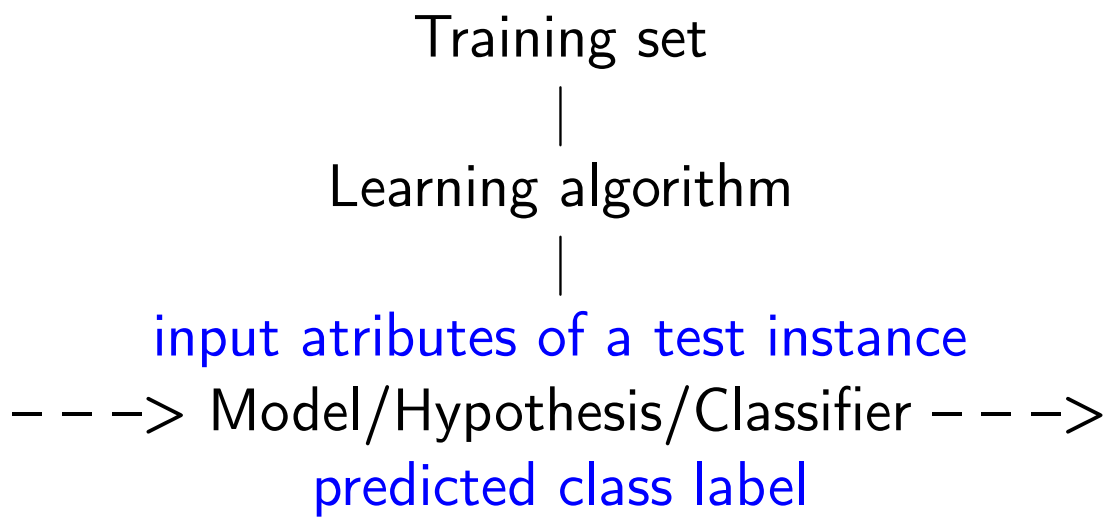
Evaluation for different machne learning task

- clustering – is the number of clusters and the structure appropriate
- associations – which rule is interesting
- outlier detection – top N outliers
- classification and regression

Evaluation for different machne learning task

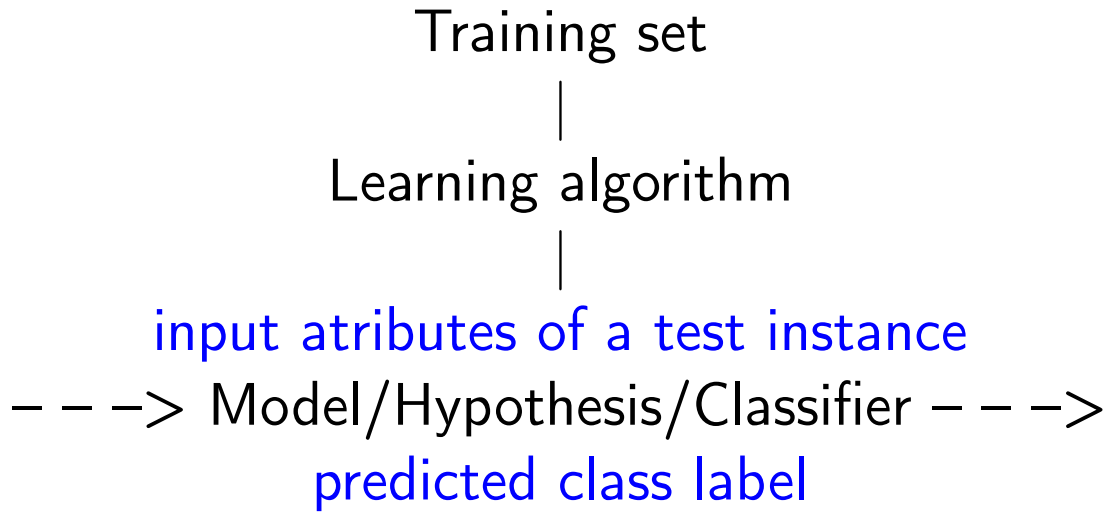
- clustering – is the number of clusters and the structure appropriate
- associations – which rule is interesting
- outlier detection – top N outliers
- classification and regression

Classification



- accuracy [celková správnost] – how often returns the correct class label
- speed – learning, testing
- robustness – to make correct predictions given noisy data or data with missing values
- scalability – efficient for large amounts of data
- comprehensibility – how is the model explainable

Classification



- accuracy [celková správnost] – how often returns correct class label
- speed – learning, testing
- robustness – to make correct predictions given noisy data or data with missing values
- scalability – efficient for large amounts of data

Classification

main criterion – how succesful Model is on data

a principal decision – what data to use for the most accurate prediction of model accuracy

Most common (but correct?)

- learning data
- test set
- cross-validation
- leave-one-out

Is there any other possibility, maybe better? bootstraping, splitting data into disjunctive parts, ...

Classification

main criterion – how succesful Model is on data

a principal decision – what data to use for the most accurate prediction of model accuracy

Most common (but correct?)

- learning data
- test set
- cross-validation
- leave-one-out

Is there any other possibility, maybe better? bootstraping, splitting data into disjunctive parts, ...

Classification

main criterion – how succesful Model is on data.

a principal decision – what data to use for the most accurate prediction of model accuracy

Most common (but correct?)

- learning data
- test set
- cross-validation
- leave-one-out

Is there any other possibility, maybe better? bootstraping, splitting data into disjunctive parts, ...

Classification

main criterion – how succesful Model is on data.

a principal decision – what data to use for the most accurate prediction of model accuracy

Most common (but correct?)

- learning data
- test set
- cross-validation
- leave-one-out

Is there any other possibility, maybe better? bootstraping, splitting data into disjunctive parts, ...

Classification

main criterion – how succesful Model is on data.

a principal decision – what data to use for the most accurate prediction of model accuracy

Most common (but correct?)

- learning data
- test set
- cross-validation
- leave-one-out

Is there any other possibility, maybe better? bootstraping, splitting data into disjunctive parts, ...

Classification

main criterion – how succesful Model is on data.

a principal decision – what data to use for the most accurate prediction of model accuracy

Most common (but correct?)

- learning data
- test set
- cross-validation
- leave-one-out

Is there any other possibility, maybe better? bootstraping, splitting data into disjunctive parts, ...

Confusion matrix

		Predicted class: *yes*	Predicted class: *no*	Total
Actual class	*yes*	TP	FN	P
	no	FP	TN	N
	Total	P'	N'	$P + N$

TP, TN, FP, FN ... the number of true positive, true negative, false positive, false negative

P, N ... cardinality of positive and negative samples

Evaluation measures

(overall) accuracy [celková správnost]

$$Acc = \frac{TP+TN}{TP+TN+FP+FN}$$

error rate, (misclassification rate) [chyba]

$$Err = 1 - Acc = \frac{w_{FP}*FP+w_{FN}*FN}{TP+TN+FP+FN}$$

w_{FP}, w_{FN} ... weight of FP and FN errors

default $w_{FP}, w_{FN} = 1$

precision

$$\frac{TP}{TP+FP}$$

sensitivity, true positive rate, recall

$$\frac{TP}{TP+FN}$$

specificity, true negative rate

$$\frac{TN}{TN+FP}$$

Evaluation measures

(overall) accuracy [celková správnost]

$$Acc = \frac{TP+TN}{TP+TN+FP+FN}$$

error rate, (misclassification rate) [chyba]

$$Err = 1 - Acc = \frac{w_{FP}*FP+w_{FN}*FN}{TP+TN+FP+FN}$$

w_{FP}, w_{FN} ... weight of FP and FN errors

default $w_{FP}, w_{FN} = 1$

precision

$$\frac{TP}{TP+FP}$$

sensitivity, true positive rate, recall

$$\frac{TP}{TP+FN}$$

specificity, true negative rate

$$\frac{TN}{TN+FP}$$

Evaluation measures

Accuracy for a class P, N

F-measures combines precision and recall

F, F1, F-score = hramonic mean of precision and recall

$$F_1 = \frac{2 * precision * recall}{precision + recall}$$

$$F_\beta = \frac{(1+\beta^2) precision * recall}{\beta^2 * precision + recall}$$

β ... a non-negative real number

Evaluation measures for regression trees

Table 5.8 Performance measures for numeric prediction*.

Performance measure	Formula
mean-squared error	$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}$
root mean-squared error	$\sqrt{\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}}$
mean absolute error	$\frac{\lvert p_1-a_1\rvert+\ldots+\lvert p_n-a_n\rvert}{n}$
relative squared error	$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\ldots+(a_n-\bar{a})^2}$, where $\bar{a}=\frac{1}{n}\sum_i a_i$
root relative squared error	$\sqrt{\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\ldots+(a_n-\bar{a})^2}}$
relative absolute error	$\frac{\lvert p_1-a_1\rvert+\ldots+\lvert p_n-a_n\rvert}{\lvert a_1-\bar{a}\rvert+\ldots+\lvert a_n-\bar{a}\rvert}$
correlation coefficient	$\frac{S_{PA}}{\sqrt{S_P S_A}}$, where $S_{PA}=\frac{\sum_i (p_i-\bar{p})(a_i-\bar{a})}{n-1}$, $S_p=\frac{\sum_i (p_i-\bar{p})^2}{n-1}$, and $S_A=\frac{\sum_i (a_i-\bar{a})^2}{n-1}$

* p are predicted values and a are actual values.

Comparing different settings of classifiers

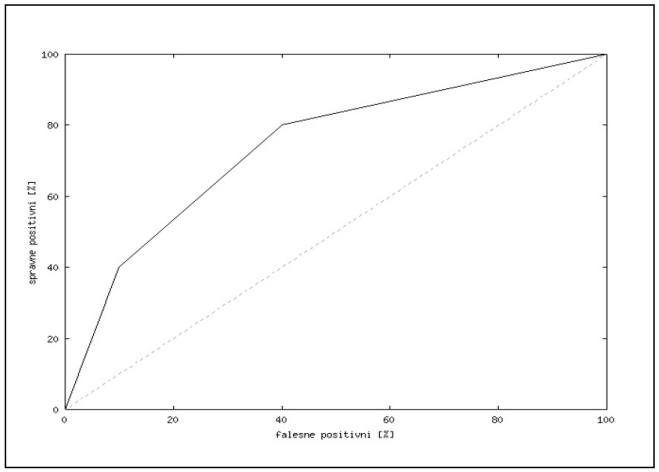
most common : Learning curve, ROC, Recall-Precision curve

Learning curve

- Performance as a function of number of iterations
- e.g. X= a number of learning instances, Y = accuracy

,

ROC curve



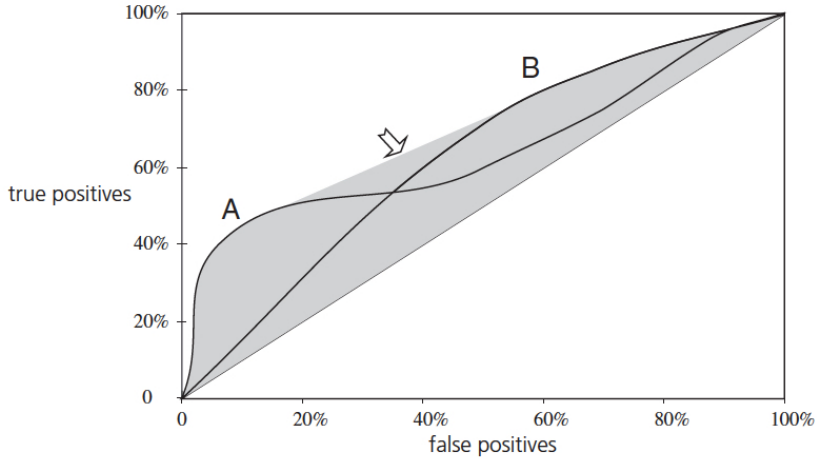


Figure 5.3 ROC curves for two learning methods.

- relation between *TP* and *FP*
- for models that returns in addition to a prediction, also weight (probability)
- Declare x_n to be a positive if $P(y = 1 \mid x_n) > \Theta$
- otherwise declare it to be negative (y=0)
- Number of TPs and FPs depends on threshold Θ. As we change Θ, we get different (TPR, FPR) points.

Comparing different classifiers

Cross-validation

The following table gives a possible result of evaluating three learning algorithms on a data set with 10-fold cross-validation:

t	*Fold*	*Naive Bayes*	*Decision tree*	*Nearest neighbour*
	1	0.6809	0.7524	0.7164
	2	0.7017	0.8964	0.8883
	3	0.7012	0.6803	0.8410
	4	0.6913	0.9102	0.6825
	5	0.6333	0.7758	0.7599
	6	0.6415	0.8154	0.8479
	7	0.7216	0.6224	0.7012
	8	0.7214	0.7585	0.4959
	9	0.6578	0.9380	0.9279
	10	0.7865	0.7524	0.7455
	avg	**0.6937**	**0.7902**	**0.7606**
	stdev	0.0448	0.1014	0.1248

The last two lines give the average and standard deviation over all ten folds. Clearly the decision tree achieves the best result, but should we completely discard nearest neighbour?

Significance testing in cross-validation: the paired t-test

- For a pair of algorithms we calculate the difference in accuracy on each fold; this difference is normally distributed if the two accuracies are. Null hypothesis: the true difference is 0, so that any differences in performance are attributed to chance. We calculate a p-value using the normal distribution, and reject the null hypothesis if the p-value is below our significance level α.
- We don't have access to the true standard deviation in the differences, which therefore needs to be estimated.Use distribution is referred to as the t-distribution.
- The extent to which the t-distribution is more heavy-tailed than the normal distribution is regulated by the number of degree of freedom: in our case this is equal to 1 less than the number of folds (since the final fold is completely determined by the other ones).

Paired t-test

The numbers show pairwise differences in each fold. The null hypothesis in each case is that the differences come from a normal distribution with mean 0 and unknown standard deviation.

t

Fold	*NB−DT*	*NB−NN*	*DT−NN*
1	-0.0715	-0.0355	0.0361
2	-0.1947	-0.1866	0.0081
3	0.0209	-0.1398	-0.1607
4	-0.2189	0.0088	0.2277
5	-0.1424	-0.1265	0.0159
6	-0.1739	-0.2065	-0.0325
7	0.0992	0.0204	-0.0788
8	-0.0371	0.2255	0.2626
9	-0.2802	-0.2700	0.0102
10	0.0341	0.0410	0.0069
avg	-0.0965	-0.0669	0.0295
stdev	0.1246	0.1473	0.1278
p-value	0.0369	0.1848	0.4833

The p-value in the last line of the table is calculated by means of the t-distribution with $k - 1 = 9$ degrees of freedom, and only the difference between the naive Bayes and decision tree algorithms is found significant at $\alpha = 0.05$.

Interpretation of results over multiple data sets

The t-test is not appropriate for multiple data sets because performance measures cannot be compared across data sets. In order to compare two learning algorithms over multiple data sets we need to use a test specifically designed for that purpose such as Wilcoxon's signed-rank test.

- The idea is to rank the performance differences in absolute value, from smallest (rank 1) to largest (rank n).
- We then calculate the sum of ranks for positive and negative differences separately, and take the smaller of these sums as our test statistic.
- For a large number of data sets (at least 25) this statistic can be converted to one which is approximately normally distributed, but for smaller numbers the critical value (the value of the statistic at which the p-value equals α) can be found in a statistical table.

Interpretation of results over multiple data sets II

- The Wilcoxon signed-rank test assumes that larger performance differences are better than smaller ones, performance differences are treated as ordinals rather than real-valued.
- Furthermore, there is no normality assumption regarding the distribution of these differences which means, among other things, that the test is less sensitive to outliers.
- In statistical terminology the test is 'non-parametric' as opposed to a parametric test such as the t-test which assumes a particular distribution. Parametric tests are generally more powerful when that assumed distribution is appropriate but can be misleading when it is not.

Wilcoxon's signed-rank test

t	*Data set*	*NB−DT*	*Rank*
	1	-0.0715	4
	2	-0.1947	8
	3	0.0209	1
	4	-0.2189	9
	5	-0.1424	6
	6	-0.1739	7
	7	0.0992	5
	8	-0.0371	3
	9	-0.2802	10
	10	0.0341	2

The sum of ranks for positive differences is $1 + 5 + 2 = 8$ and for negative differences $4 + 8 + 9 + 6 + 7 + 3 + 10 = 47$. The critical value for 10 data sets at the $\alpha = 0.05$ level is 8, which means that if the smallest of the two sums of ranks is less than or equal to 8 the null hypothesis that the ranks are distributed the same for positive and negative differences can be rejected. This applies in this case.

Multiple algorithms over multiple data sets

- Friedman test tells us whether the average ranks as a whole display significant differences
- further analysis is needed on a pairwise level. This is achieved by applying a post-hoc test once the Friedman test gives significance - Nemenyi test
- A variant of the Nemenyi test called the Bonferroni–Dunn test can be applied when we perform pairwise tests only against a control algorithm.

Sampling

- holdout – split data randomly to learning and test data, e.g. 2/3 vs. 1/3
  stratified sampling – preserve relative frequency of classes in samples
- Random (sub)sampling – holdout method is repeated k times
  The overall accuracy estimate is taken as the average of the accuracies obtained from each iteration.
- bootstraping
- undersampling/oversampling of a class – for processing imbalanced data