Data Mining - Evaluation of Classifiers

Lecturer: JERZY STEFANOWSKI
Institute of Computing Sciences
Poznan University of Technology
Poznan, Poland

Lecture 4

SE Master Course

2008/2009 revised for 2010

Discovering and evaluating classification knowledge

Creating classifiers is a multi-step approach:

- Generating a classifier from the given learning data set,
- Evaluation on the test examples,
- Using for new examples.

Train and test paradigm!

Evaluation criteria (1)

- *Predictive* (*Classification*) *accuracy*: this refers to the ability of the model to correctly predict the class label of new or previously unseen data:
  - accuracy = % of testing set examples correctly classified by the classifier
- *Speed*: this refers to the computation costs involved in generating and using the model
- *Robustness*: this is the ability of the model to make correct predictions given noisy data or data with missing values

Evaluation criteria (2)

- *Scalability*: this refers to the ability to construct the model efficiently given large amount of data
- *Interpretability*: this refers to the level of understanding and insight that is provided by the model
- *Simplicity*:
  - decision tree size
  - rule compactness
- Domain-dependent quality indicators

Predictive accuracy / error

- General view (statistical learning point of view):
- Lack of generalization – prediction risk:

$$R(f) = E_{xy} L(y, f(x))$$

- where $L(y, \hat{y})$ is a loss or cost of predicting value $\hat{y}$ when the actual value is y and *E* is expected value over the joint distribution of all (*x*,*y*) for data to be predicted.
- Simple classification $\rightarrow$ zero-one loss function

$$L(y, \hat{y}) = \begin{cases} 0 & if\ y = f(y) \\ 1 & if\ y \neq f(y) \end{cases}$$

Evaluating classifiers – more practical …

Predictive (classification) accuracy (0-1 loss function)

- Use testing examples, which do not belong to the learning set
  - N_t – number of testing examples
  - N_c – number of correctly classified testing examples
- Classification accuracy: $$\eta = \frac{N_c}{N_t}$$
- (Misclassification) Error: $$\varepsilon = \frac{N_t - N_c}{N_t}$$
- Other options:
  - analysis of confusion matrix

A confusion matrix

	Predicted		
Original classes	K_1	K_2	K_3
K_1	50	0	0
K_2	0	48	2
K_3	0	4	46

- Various measures could be defined basing on values in a confusion matrix.

Confusion matrix and cost sensitive analysis

	Predicted		
Original classes	K_1	K_2	K_3
K_1	50	0	0
K_2	0	48	2
K_3	0	4	46

$$\mathrm{C}(\varepsilon) = \sum_{i=1}^{r} \sum_{j=1}^{r} n_{ij} \cdot c_{ij}$$

- Costs assigned to different types of errors.
- Costs are unequal
- Many applications:
  loans, medical diagnosis, fault detections,
  spam …
- Cost estimates may be difficult to be acquired from real experts.

Experimental evaluation of classifiers

- How predictive is the model we learned?
- Error on the training data is *not* a good indicator of performance on future data
  - ***Q: Why?***
  - A: Because new data will probably not be **exactly** the same as the training data!
- Overfitting – fitting the training data too precisely - usually leads to poor results on new data.
  - Do not learn too much peculiarities in training data; think about generality abilities!
  - We will come back to it latter during the lecture on *pruning* structures of classifiers.

Experimental estimation of classification accuracy

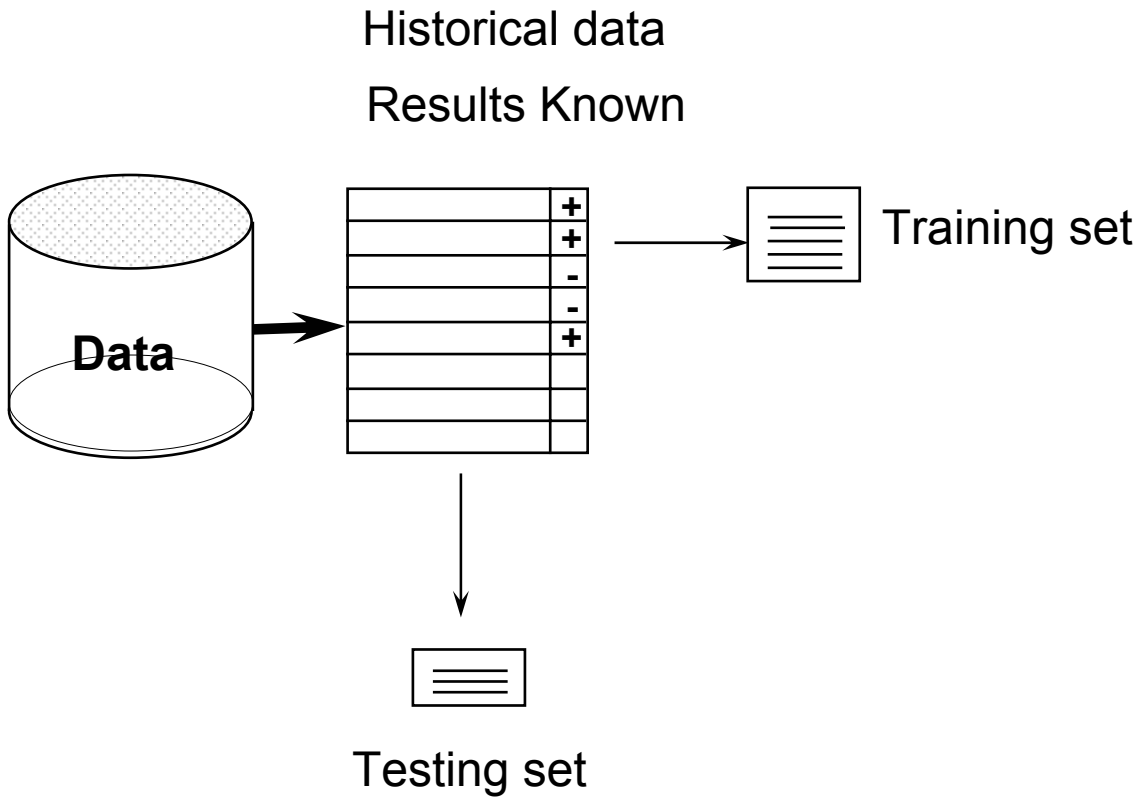
Random partition into **train and test** parts:

- Hold-out
  - use two independent data sets, e.g., training set (2/3), test set(1/3); random sampling
  - repeated hold-out
- *k*-fold cross-validation
  - randomly divide the data set into *k* subsamples
  - use *k-1* subsamples as training data and one sub-sample as test data --- repeat *k* times
- Leave-one-out for small size data

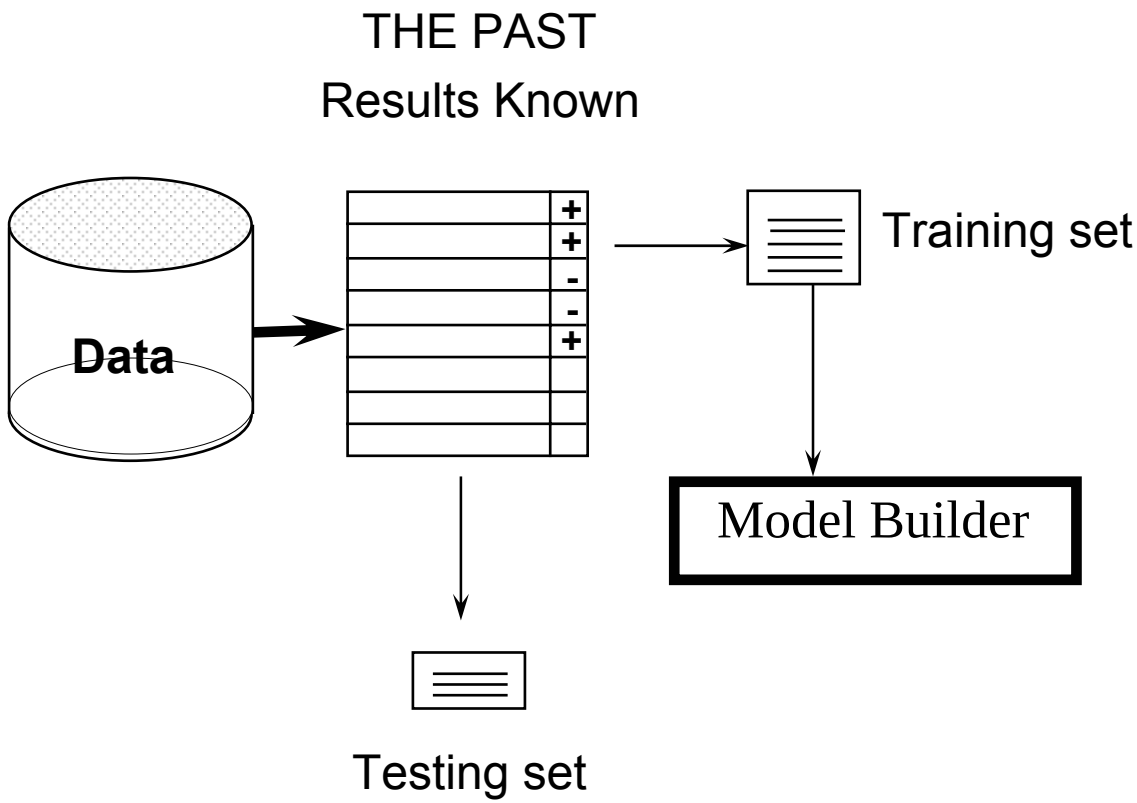
Evaluation on “LARGE” data, hold-out

- A simple evaluation is sufficient
  - Randomly split data into training and test sets (usually 2/3 for train, 1/3 for test)
- Build a classifier using the *train* set and evaluate it using the *test* set.

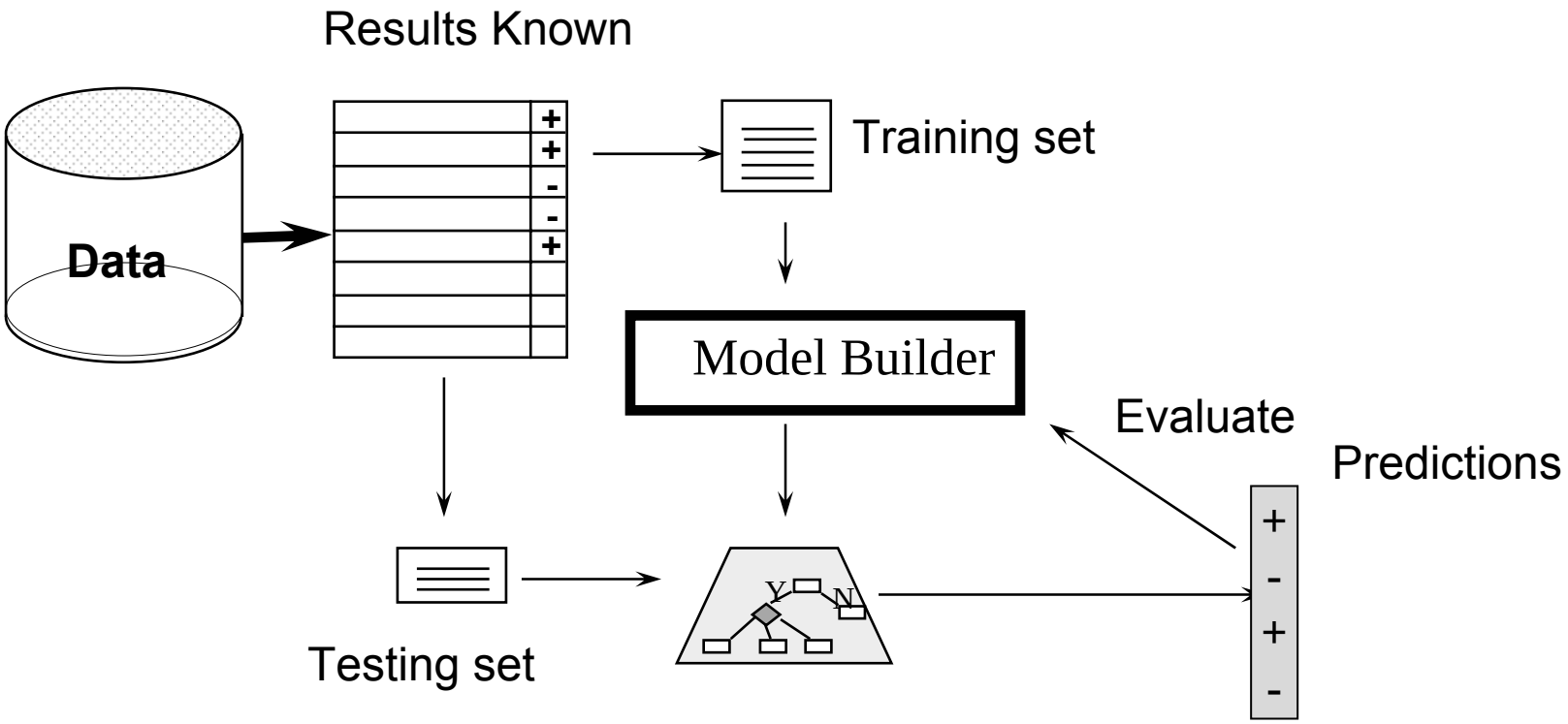
Step 1: Split data into train and test sets



Step 2: Build a model on a training set



Step 3: Evaluate on test set



Remarks on hold-out

- It is important that the test data is not used *in any way* to create the classifier!
- One random split is used for really large data
- For medium sized $\rightarrow$ **repeated hold-out**
- Holdout estimate can be made more reliable by repeating the process with different subsamples
  - In each iteration, a certain proportion is randomly selected for training (possibly with stratification)
  - The error rates (classification accuracies) on the different iterations are averaged to yield an overall error rate
  - Calculate also a standard deviation!

Repeated holdout method, 2

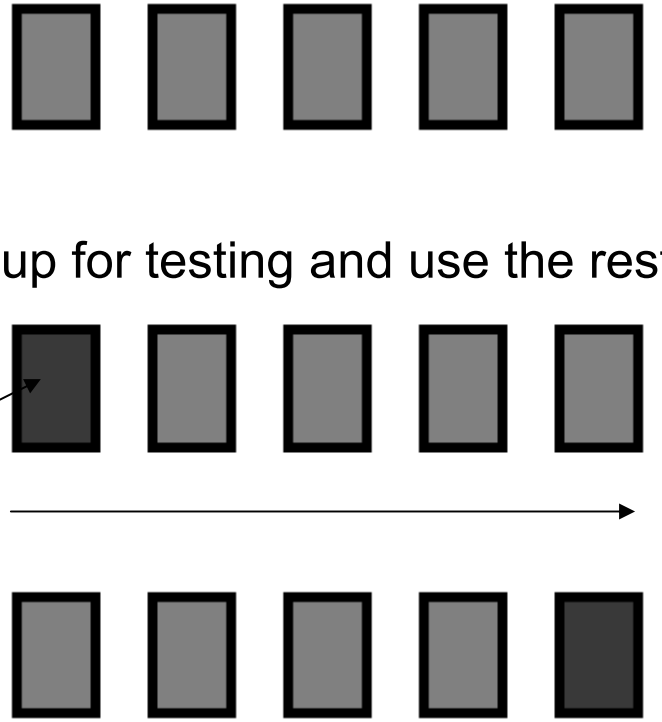
- Still not optimum: the different test sets usually overlap (difficulties from statistical point of view).
- Can we prevent overlapping?

Cross-validation

- *Cross-validation* avoids overlapping test sets
  - First step: data is split into *k* subsets of equal size
  - Second step: each subset in turn is used for testing and the remainder for training
- This is called *k-fold cross-validation*
- Often the subsets are stratified before the cross-validation is performed
- The error estimates are averaged to yield an overall error estimate

Cross-validation example:

- Break up data into groups of the same size
-
-
- Hold aside one group for testing and use the rest to build model
-
- Repeat



More on 10 fold cross-validation

- Standard method for evaluation: stratified ten-fold cross-validation
- Why ten? Extensive experiments have shown that this is the best choice to get an accurate estimate
  (since CART book by Breiman, Friedman, Stone, Olsen 1994)
  However, other splits – e.g. 5 cv – are also popular.
- Also the standard deviation is essential for comparing learning algorithms.
- Stratification reduces the estimate's variance!
- Even better: repeated stratified cross-validation
  - E.g. ten-fold cross-validation is repeated more times and results are averaged (reduces the variance)!

Leave-One-Out cross-validation

- Leave-One-Out:
  a particular form of cross-validation:
  - Set number of folds to number of training instances
  - i.e., for *n* training instances, build classifier *n* times but from *n* -1 training examples …
- Makes best use of the data.
- Involves no random sub-sampling.
- Quite computationally expensive!

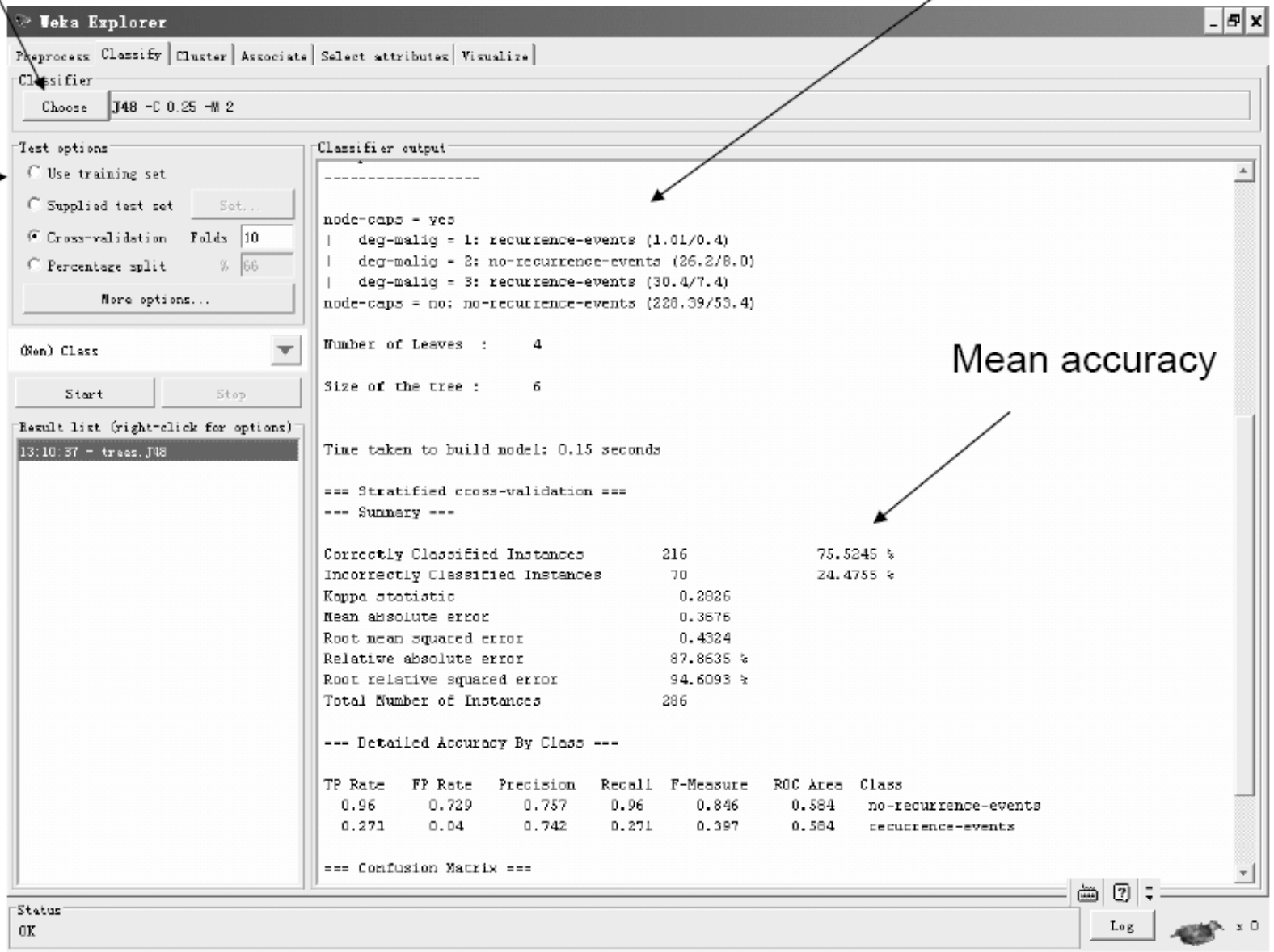
WEKA Explorer

Classifier

Testing data

Decision Trees

Mean accuracy



Comparing data mining algorithms

- Frequent situation: we want to know which one of two learning schemes performs better.
- Note: this is domain dependent!
- Obvious way: compare 10-fold CV estimates.
- Problem: variance in estimate.
- Variance can be reduced using repeated CV.
- However, we still don’t know whether the results are reliable.
  - There will be a long explanation on this topic in future lectures

Comparing two classifiers on the same data

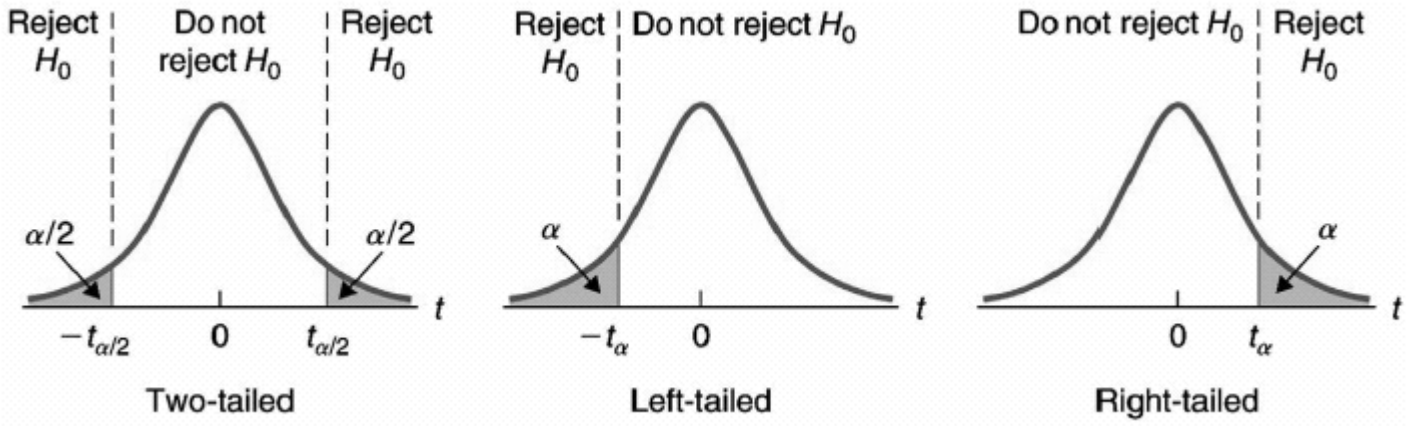
- Summary of results in separate folds

Podział	Kl_1	Kl_2
1	87,45	88,4
2	86,5	88,1
3	86,4	87,2
4	86,8	86
5	87,8	87,6
6	86,6	86,4
7	87,3	87
8	87,2	87,4
9	88	89
10	85,8	87,2
Srednia	**86,98**	**87,43**
Odchylenie	**0,65**	**0,85**

The general question: given two classifiers K1 and K2 produced by feeding a training dataset D to two algorithms A1 and A2,
which classifier will be more accurate in classifying new examples?

Paired t-test

- The null hypothesis H0: the average performance of classifiers on the data D is =
- H1: usually $\neq$
- Test statistics and the decision based on α



- Remark: assumption $\rightarrow$ the paired difference variable should be normally distributed!

Summary

- What is the classification task?
- Discovering classifiers is a muti-step approach.
  - Train and test paradigm.
- How could you evaluate the classification knowledge:
  - Evaluation measures – predictive ability.
- Empirical approaches – use independent test examples.
  - Hold-out vs. cross validation.
  - Repeated 10 fold stratified cross validation.
- More advances issues (e.g. more about comparing many algorithms and ROC analysis will be presented during future lectures)

Klasyfikacja binarna (chory vs. zdrowy)

- Jedna z klas posiada szczególne znaczenie, np. diagnozowanie poważnej choroby

	Przewidywane klasy decyzyjne	
Oryginalne klasy	Pozytywna	Negatywna
Pozytywna	**TP**	FN
Negatywna	**FP**	**TN**

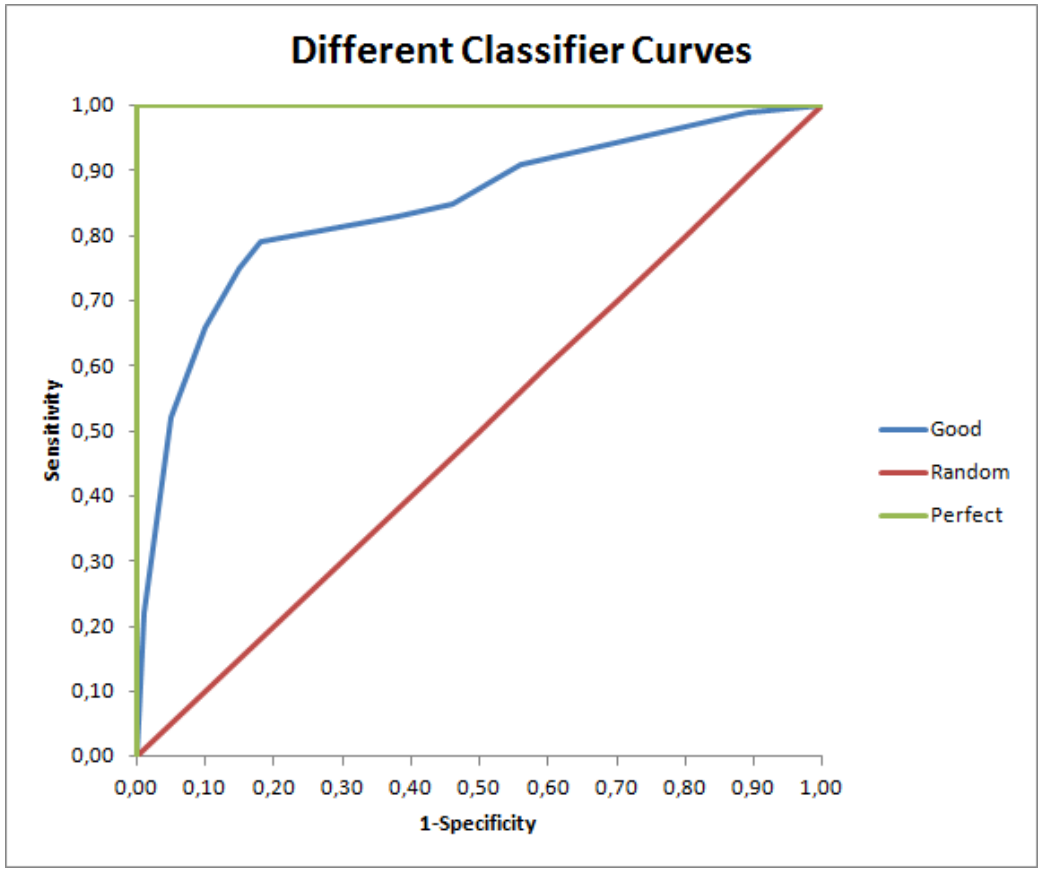
- Nazewnictwo (inspirowane medycznie):
  - TP (*true positive*) – liczba poprawnie sklasyfikowanych przykładów z wybranej klasy (*hit*),
  - FN (false negative) – liczba błędnie sklasyfikowanych przykładów z tej klasy (*miss*),
  - TN (*true negative*) – liczba przykładów poprawnie nie przydzielonych do wybranej klasy (*correct rejection*),
  - FP (*false positive*) – liczba przykładów błędnie przydzielonych do wybranej klasy, podczas gdy w rzeczywistości do niej nie należą (*false alarm*)

Miary oceny dla klasyfikacji binarnej

- Dodatkowe miary oceny rozpoznawania wybranej klasy:
  - Wrażliwość / czułość (*sensitivity*) = TP / (TP+FN)
  - Specyficzność (*specificity*) = TN / (FP+TN)
- Inne miary:
  - *False-positive rate* = FP / (FP+TN), czyli 1 – specyficzność.

	Przewidywane klasy decyzyjne	
Oryginalne klasy	Pozytywna	Negatywna
Pozytywna	TP	FN
Negatywna	FP	TN

Krzywa ROC (receiver operating characteristic)



Pole pod krzywą ROC –"całościowa" charakterystyka klasyfikatora

0.9 – 1.0 – *excellent* (A)

0.8 – 0.9 – *good* (B)

0.7 – 0.8 – *fair* (C)

0.6 – 0.7 – *poor* (D)

0.5 – 0.6 – *fail* (E)

http://gim.unmc.edu/dxtests

Krzywa ROC – przykład zastosowania

JAMA | Journals | Enter Search Term

This Issue | Views 153,309 | Citations 276 | Altmetric 884

Download PDF | More | Cite This | Permissions

Original Investigation | Innovations in Health Care Delivery FREE

December 13, 2016

Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs

Varun Gulshan, PhD[1]; Lily Peng, MD, PhD[1]; Marc Coram, PhD[1]; et al

» Author Affiliations | Article Information

JAMA. 2016;316(22):2402-2410. doi:10.1001/jama.2016.17216

Editorial Comment

Key Points

Question How does the performance of an automated deep learning algorithm compare with manual grading by ophthalmologists for identifying diabetic retinopathy in retinal fundus photographs?

Finding In 2 validation sets of 9963 images and 1748 images, at the operating point selected for high specificity, the algorithm had 90.3% and 87.0% sensitivity and 98.1% and 98.5% specificity for detecting referable diabetic retinopathy, defined as moderate or worse diabetic retinopathy or referable macular edema by the majority decision of a panel of at least 7 US board-certified ophthalmologists. At the operating point selected for high sensitivity, the algorithm had 97.5% and 96.1% sensitivity and 93.4% and 93.9% specificity in the 2 validation sets.

Meaning Deep learning algorithms had high sensitivity and specificity for detecting diabetic retinopathy and macular edema in retinal fundus photographs.