Introduction to Machine Learning
NPFL 054

http://ufal.mff.cuni.cz/course/npfl054

Barbora Hladká
hladka@ufal.mff.cuni.cz

Martin Holub
holub@ufal.mff.cuni.cz

Charles University,
Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics

Outline

- **Basics of classifier evaluation**
  - why we need evaluation
  - working with data
  - sample error and generalization error

- **Overfitting**

Fundamentals of classifier evaluation

You need thorough evaluation to

1. **get a reliable estimate of the classifier performance**
   – i.e. how it will perform on new – so far unseen – data instances
   – possibly even in the future

2. **compare your different classifiers** that you have developed
   – to decide which one is "the best"

= Model assessment and selection

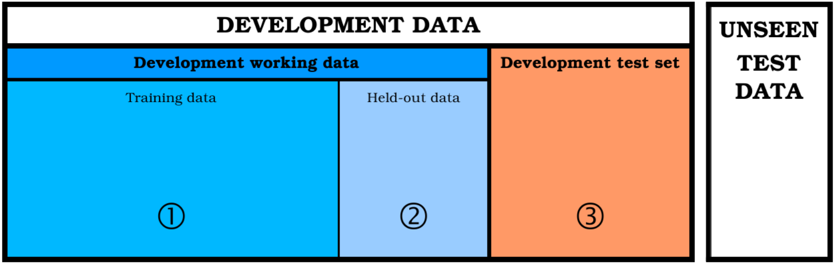
Model robustness

You need *good* performance

not only on *your* data,

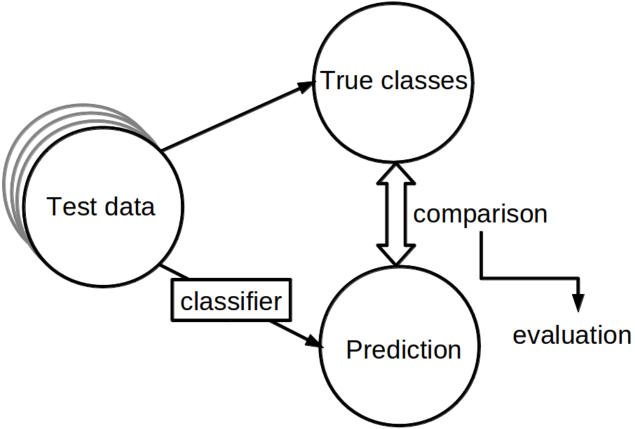
but also on any data that can be *expected*!

Working with data
Development data and its division



All subsets should be selected randomly in order to represent the characteristic distribution of both feature values and target values in the available set of examples.

Evaluation – basic scheme



Development data – the working portion

Development working data

Is used both for training your classifier and for evaluation when you tune the learning parameters.

- **Training data**
  is used for **training** your classifier with a particular learning parameter settings when you tune your classifier

- **Held-out data**
  is used for **evaluating** your classifier with a particular learning parameter settings when you tune your classifier

Development data – the test portion

Development test set

- the purpose is to simulate the "real" test data
- should be used only for your final development evaluation when your classifier has already been tuned and your learning parameters are finally set
- using it you get an estimate of your classifier's performance at the end of the development
- is also used for model selection

Sample accuracy and sample error rate

To measure the performance of classification tasks we often use (sample) *accuracy* and (sample) *error rate*

Sample accuracy is the number of correctly predicted examples divided by the number of all examples in the predicted set

Sample error rate is equal to **1 - accuracy**

Training error rate is the sample error rate measured on the training data set

Test error rate is the sample error rate measured on the test data set

Sample error and generalization error

Sample error of a hypothesis h with respect to a data sample S of the size n is usually measured as follows

- for **regression**: **mean squared error** $\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$
- for **classification**: **classification error** $= \frac{1}{n}\sum_{i=1}^{n}\mathrm{I}(\hat{y}_i \neq y_i)$

Generalization error (aka "true error" or "expected error") measures how well a hypothesis h generalizes beyond the used training data set, to unseen data with distribution $\mathcal{D}$. Usually it is defined as follows

- for **regression**: $\mathrm{error}_{\mathcal{D}}(h) = \mathsf{E}\,(\hat{y}_i - y_i)^2$
- for **classification**: $\mathrm{error}_{\mathcal{D}}(h) = \Pr\,(\hat{y}_i \neq y_i)$

Minimizing generalization error vs. overfitting

Finding a model that minimizes generalization error
. . . is one of central goals of the machine learning process