# Exercise 7

**Deadline: 12.01.2015**

## 1 Proof of the covariance penalty formula (6 points)

We want you to show that the in sample generalization error on the test set $\mathrm{Err}_{in}$ for squared loss is given by:

$$\mathrm{Err}_{in} = \mathrm{err} + \frac{2}{N}\sum_i \mathrm{cov}\left(y_i, \hat{y}_i\right) \tag{1}$$

Here err is the error on the training set $S = \{(x_i, y_i)\}_{i=1}^N$. Note that we create the test set $S'$ by redrawing the $y'$ for the $x$ in the training set: $S' = \{(x_i, y_i')\}_{i=1}^N$. We assume that the measured values are obscured by Gaussian noise:

$$y_i = \mu_i + \epsilon_i \tag{2}$$
$$y_i' = \mu_i + \epsilon_i' \tag{3}$$
$$\epsilon_i, \epsilon' \sim \mathcal{N}(0, \sigma^2)\ . \tag{4}$$

Because of eq. (4) the mean values of the training and the test set $\mu_i = \mathbb{E}(y_i)$ and $\mu_i' = \mathbb{E}(y_i')$ are equal.

Prove the claim by showing that for each $i$

$$\mathbb{E}_{y_i}(\mathrm{Err}_i - \mathrm{err}_i) = 2\mathbb{E}_{y_i}(y_i - \mu_i)(\hat{y}_i - \mu_i)\ . \tag{5}$$

Note that $\mathbb{E}_{y_i}(y_i - \mu_i)\mu_i = \mathbb{E}_{y_i}(y_i - \mu_i)\hat{\mu}_i = 0$ and therefore the right hand side of eq. (5) corresponds to a proper covariance. The test and training errors are given by:

$$\mathrm{Err}_i = \mathbb{E}_{y_i'}(y_i' - \hat{y}_i)^2 \tag{6}$$
$$\mathrm{err}_i = (y_i - \hat{y}_i)^2\ . \tag{7}$$

**Hint**: At one point you might want to insert $\mu_i - \mu_i$.

## 2 Model selection

For this exercise we go back to the digits dataset provided by sklearn. We want to compare different criteria that are all supposed to give an estimate of the optimism of the model. The basic experiment is simple. Its goal is to **distinguish threes from eights**. Therefore define a training set, train a classifier, determine the optimism via a criterion and compare with the performance on a test set. Your task is to write a nice framework in which you can test different settings. If you have a function with equal interface for all interchangeable parts you will save yourself some work. As basic classifiers we want you to take

- **Nearest Neighbor**
- **LDA**
- **QDA**
- **Naive Bayes**

(you are encouraged to use your own implementations from older exercises but if you have missed one exercise you can always fall back to the implementations from sklearn: `http://scikit-learn.org/stable/supervised_learning.html#supervised-learning`).

### 2.1 Error on test-set (2 points)

To get the best estimate possible for the optimism of the used algorithms you are supposed to use repeated 2-fold cross validation on the whole data for each algorithm. These test errors will be the reference values for the experiments in 2.2.

### 2.2 Modle selection criteria (8 points)

You should start by dividing your data into a test and a training set $(1:1)$. All experiments in 2.2 should be done on the training-set only. The criteria that you are supposed to test are:

- **k-Fold Cross Validation**; For $k = 2$ and $k = 10$.
- **Repeated k-Fold Cross Validation**; For $k = 2$ and $k = 10$.
- **Corrected k-Fold Cross Validation**; For $k = 2$ and $k = 10$.
- **Reverse 10-Fold Cross Validation**; Train on $\frac{1}{10}$ of the data and predict on the remaining parts.
- **Bootstrap Sampling**; The difference to cross validation is that the subset of data points for training is drawn with replacement – it is possible that the same sample contributes multiple times.
- **Covariance Penalty with Rademacher Sampling**
- **Akaike information criterion**;

  $AIC_c = N * \log(\text{err}) + 2(d_f + 1)\frac{N}{N - d_f - 2}$ ($N$ - number of samples; $d_f$ degrees of freedom; err - error on training set)
- **BIC**; the Bayesian information criterion is closely related to the Akaike criterion. Therefore a slight change in your Akaike function should be sufficient to compute the BIC:

  $BIC = N * \log(\text{err}) + \log(N)d_f$

Present your finding in one table per algorithm. In each table show the the real optimism (from 2.1) as well as all the estimated ones against the size of the training set (Compare the whole training set vs $\frac{1}{5}$ of it). Comment on your results. Is there a criterion that constantly gives more reliable results than the others? Which of the models has the best trade-off between complexity and performance?

Note that the nearest neighbor classifier is somehow special since its training error is always zero. The Akaike criterion and the covariance penalty with Rademacher sampling will both have their problems here.

### 2.3 Reduced featurespace (4 points)

Repeat 2.1 and 2.2 for a reduced, two dimensional feature-space (use your feature reduction method from the old exercise). Add your results to the respective tables from 2.2.

## Regulations

Please hand in the python code, figures and explanations (describing clearly which belongs to which). Non-trivial sections of your code should be explained with short comments, and variables should have self-explanatory names. Plots should have informative axis labels, legends and captions. Please enclose all results into a single .pdf document and hand in the .py files that created the results. Please email the solutions to `niko.krasowski@iwr.uni-heidelberg.de` before the deadline. You may hand in the exercises in teams of maximally three people, which must be clearly named on the solution sheet (one email is sufficient). Discussions between different teams about the exercises are encouraged, but the code must not be copied verbatim (the same holds for any implementations which may be available on the WWW). Please respect particularly this rule, otherwise we cannot

give you a passing grade. Solutions are due by email at the beginning of the next exercise. For each exercise there will be maximally 20 points assigned. If you have 50% or more points in the end of the semester you will be allowed to take the exam.