General remarks

The goal is to help you understand the basic underlying principles of classification techniques commonly used in machine learning, and give you practical experience to expand on this knowledge. You will use some existing implementations of classification algorithms and implement some yourself to gain a deeper understanding of how to put classification into practice. You will get to know the following classification techniques:

- nearest neighbour
- naive bayes
- decision tree
- support vector machine

We delay the discussion of another common classifier - namely neural networks - to later in this course.
All classifiers should work at least with the **IRIS data** set:
http://archive.ics.uci.edu/ml/datasets/Iris.
Note: Scikit-learn already contains the IRIS data set in the required format.

Literature

When you use commonly available literature for the solution of your problem (i.e. books, research papers, etc.), you have to provide the references. If the literature is not commonly available (i.e. special handout scripts, etc.), you are required to provide a copy of this literature.

The following literature is recommended:

- Artificial Intelligence - A modern Approach, Stuart Russel and Peter Norvig, Prentice Hall, 2003
- Machine Learning and Pattern Recognition, Christopher M. Bishop, Springer, 2006
- Machine Learning, Tom Mitchell, McGraw Hill, 1997
- Information Theory, Inference,and Learning Algorithms, David J.C. MacKay, Cambridge University Press, 2003

Problem 5.1 (Nearest Neighbour Classifier) **(28 P.)**

For this task download the file *classification.py* from StudIP and complete the function stubs where marked.

a) Implement a k-nearest neighbour classifier yourself (don’t use the implementation from sklearn) that allows to apply various distance measures and various neighbourhoods. Your classifier should support **normalized euclidean distance** and **two additional distance measures** of your choice. Report which distance measures you used. (10 P.)

b) Perform training and test on the IRIS data set using 10-fold cross validation. Try to find the optimal neighbourhood size using normalized euclidean distance. **Illustrate** the classification accuracy of your classifier for all tested neighbourhood sizes ($1 \ldots 100$). **Explain** the reason for the shape of the plotted curve. (14 P.)

c) Repeat classification for your best neighbourhood size with your additional distance measures. Report the results. (4 P.)

Problem 5.2 (Bayesian classification - Optimality and minimum risk) **(12 P.)**
Assume two classes y_1 and y_2 and a single continuous feature x. The probability density functions are given by Gaussians with variance $\sigma^2 = 0.5$ for both classes. Mean values are 0 and 1 respectively:

$$\text{pdf class } y_1: \quad p(x|y_1) = \frac{1}{\sqrt{\pi}} \exp(-x^2)$$

$$\text{pdf class } y_2: \quad p(x|y_2) = \frac{1}{\sqrt{\pi}} \exp(-(x-1)^2)$$

If $P(y_1) = P(y_2) = 0.5$:

a) Find the threshold (decision boundary) value x_0 that minimizes the classification error. (2 P.)

b) Compared to (a), where would a threshold move, that minimizes the risk of misclassification given the loss matrix:

$$\begin{pmatrix} 0 & 5 \\ 2 & 0 \end{pmatrix}$$

No calcuatlion needed (see (c)). **Explain and show your reasoning!** (3 P.)

c) Calculate the threshold asked about in (b). (5 P.)

d) Given a loss matrix with L_{kj}, the expected risk is minimized if, for each $\mathbf{x}$, we choose the class that minimizes

$$\sum_k L_{kj} p(y_k, \mathbf{x})$$

Verify that, when the loss matrix is given by $L_{kj} = 1 - I_{kj}$, where I_{kj} are the elements of the identity matrix, this reduces to the criterion of choosing the class having the largest posterior probability. What is the interpretation of this form of loss matrix? (2 P.)

Note: Please provide the complete outline of your solution and provide short(!) comments for each important step towards your solution. Document any assumption you make.

Problem 5.3 (Bayesian classification - Naive bayes) **(20 P.)**
For this task download the file *classification.py* from StudIP and complete the function stubs. Program a Gaussian Bayesian classifier. To find the posterior of each feature x_n, i.e. $P(y|x_n)$ use a Gaussian normal distribution to represents the pdf (probability density function) of the likelihood $p(x_n|y)$, i.e.

$$p(x_n|y) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{(x_n - \mu_n)^2}{2\sigma_n^2}\right)$$

where M = number of samples (per class) $\quad \mu_n = \frac{\sum_i x_n^{(i)}}{M} \quad \sigma_n^2 = \frac{\sum_i (x_n^{(i)} - \mu_n)^2}{M}$

a) Report the mean and variance for each feature (per class) and the prior for each class using the IRIS dataset. (5 P.)

b) Use a 10-fold cross validation. Report the corresponding accuracy of your classifier for the IRIS dataset. (15 P.)

On the hand-in date, **07.12.2016**, you must hand-in the following: [1]

a) a text file stating how much time you (all together) used to complete this exercise sheet
b) your solutions / answers / code

for problem 5.1 and 5.2 and 5.3.

[1] upload via StudIP (if there are problems with the upload contact me **beforehand**: krell@uni-bremen.de)