Assignment II:
Basic Learning Algorithms

Christian Igel, Kim Steenstrup Pedersen, Sami Brandt
Department of Computer Science, University of Copenhagen

The goal of this assignment is to get familiar with basic supervised learning methods. The first half of the assignment is based on parts of chapters 1-4 from C. M. Bishop *Pattern Recognition and Machine Learning* [2].

You have to pass this and the following mandatory assignments in order to be eligible for the exam of this course. There are in total 3 mandatory pass/fail assignments on this course, which can be solved individually or in groups of no more than 3 participants. The course will end with a larger written exam assignment which must be solved individually and is graded (7-point scale).

The deadline for this assignment is **March 5, 2013**. You must submit your solution electronically via the Absalon home page. Go to the assignments list and choose this assignment and upload your solution prior to the deadline. If you choose to work in groups on this assignment you should only upload one solution, but remember to include the names of all participants both in the solution as well as in Absalon when you submit the solution. If you do not pass the assignment, having made a *serious* attempt, you may get a second chance of submitting a new solution.

A solution consist of:

- Your solution source code (Matlab / R / Python scripts / C or C++ / Java code) with comments about the major steps involved in each Question (see below).
- Your code should be structured such that there is one main file that we can run to reproduce all the results presented in your report. This main file can, if you like, call other files with functions / classes.
- Your code should also include a README text file describing how to compile and run your program, as well as list of all relevant libraries needed for compiling or using your code. If we cannot make your code run we will consider your submission incomplete and you may be asked to resubmit.

- A PDF file with notes detailing your answers to the questions, which may include graphs and tables if needed (**max. 10 pages** text including figures and tables). Do *not* include your source code in this PDF file.

II.1 Regression

In the regression part of this assignment, we will study regression with linear models (as explained in Bishop Sec. 3.1) and we will experiment with different basis functions. The data set we will consider is a real-world data set which contains a set of estimates of the percentage of body fat determined by underwater weighing and various body circumference measurements from 252 men. We will try to make predictions of the percentage of body fat based on circumference measurements using linear regression models.

You can load the data set with the function `readbodyfat` (available for Matlab and R). Since the data set seems to be ordered somewhat with respect to the age of the test subjects, we should pick the members of the training and test sets at random. After loading the dataset the function performs a random split of the data set into a training and test set (in roughly 80% training and 20% test).

The file `bodyfat.txt` contains a detailed explanation of the variables in the dataset and also the dataset itself in text format if you need it. The short description is that we would like to predict percentage body fat given in the 2nd column from a subset of the variables given in columns 4 to 15. We will not use the values found in column 1 (column 2 is actually a function of column 1) and column 3 (the age of the subject). We will think of column 2 as what is referred to in [2] as the target variable t and a selection of columns 4 to 15 as the observations $\mathbf{x}$. We will in the following consider the following two selections of observation variables:

Selection 1: Let $\mathbf{x}$ consist of the data in columns 4, 7, 8, and 9, hence the dimensionality of this subset is $D = 4$.

Selection 2: Let $\mathbf{x}$ consist of the data in column 8, hence the dimensionality of this subset is $D = 1$.

II.1.1 Maximum likelihood solution

We will start by trying to model the data set with linear regression as defined in eq. (3.1) in [2] and learn the parameters using the maximum likelihood approach. That is, we will use the linear model

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_D x_D \ .$$

Deliverables: Implement this model for variable selections 1 and 2 by constructing their corresponding design matrices as defined in eq. (3.16). Train each of these models on the training set by finding the maximum likelihood (ML) estimate by using eq. (3.15) (hint: In Matlab you can compute the pseudo inverse of the design matrix with the function `pinv` and in R by using `ginv` or `pseudoinverse`). Apply each model to the test set using eq. (3.3) with the ML parameter estimate and compute and report the root mean square (RMS) error

$$\mathrm{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(t_n - y(\mathbf{x}_n, \mathbf{w}))^2} \ .$$

Which of the two models seem to provide the best prediction?

TODO next year: Make them do cross validation or at least take averages over different partitioning of the data set.

II.1.2 Maximum a posteriori solution

Next let us try to learn the two models using Bayesian learning and maximum a posteriori (MAP) estimation. Lets fix the prior distribution on the parameters to the zero mean isotropic Gaussian as given in eq. (3.52). Set the noise precision parameter to $\beta = 1$. We may then obtain the posterior distribution as given in eq. (3.49), by computing the estimates of the mean $\mathbf{m}_N$ and covariance $\mathbf{S}_N$ using eq. (3.53) and (3.54). The posterior mean is the MAP estimate.

Deliverables: Using the MAP estimate and eq. (3.3) apply each model to the test set and compute and plot the root mean square (RMS) error for different values of the prior precision parameter α. Which of the two models seem to provide the best prediction? How does the results compare with the maximum likelihood results? For what value of the prior precision parameter α does the RMS error go below the RMS for the maximum likelihood solution from Question II.1.1?

II.1.3 Theory behind the maximum a posterori solution (based on CB Ex. 3.7)

By using the technique of "completing the square", verify the result eq. (3.49) for the posterior distribution of the parameters $\mathbf{w}$ in the linear basis function model in which $\mathbf{m}_N$ and $\mathbf{S}_N$ are defined by eqs. (3.50) and (3.51) respectively.

Deliverables: The proof.

II.2 Classification

Please download the data sets `irisTrain.dt` and `irisTest.dt` from the course homepage. These data sets have been generated from the famous Iris flower data set [1], which has been used as an example for classification algorithms since the work of Ronald Fisher [3]. However, instead of the original four input features only two are considered. Furthermore, a feature has been rescaled and some examples have been removed.

The data set describes three different species of Iris, namely Iris setosa, Iris virginica and Iris versicolor. That is, we have a three class classification problem. Each line in the data files corresponds to one flower. The first two columns of our version of the data are the lengths and the widths of the sepals. Sepals are modified leaves that are part of the calyx of a flower. In our modified version of the data set, the length are measure in millimeters and the width in centimeters.

II.2.1 Linear discriminant analysis

Visualize the training data sets in a 2D scatter plot. One axis should correspond to the length, the other to the width of the sepals. Please indicate the different classes by using three different colors for the points in the scatter plot.

Now we consider building a linear model for classification. Apply linear discriminant analysis (LDA) to the training data sets and report the accuracies of the classifier on the training set as well as on the test set.

It is highly recommended that you implement the LDA algorithm by yourself. However, it is acceptable to use a software tool. If you do not implement the algorithm yourself, give arguments for why you choose a specific LDA implementation and how to use it.

Deliverables: scatter plot of the training data; implementation of LDA or description of tool used for LDA; training and test error of LDA

II.2.2 Nearest neighbor with Euclidean metric

Linear discriminant analysis is a linear technique using a parametric model. Now we apply a non-linear, non-parametric method to the same data, namely nearest neighbor classification.

Implement a k-nearest neighbor classifier (k-NN).

Train it on `irisTrain.dt` and report its accuracy on the training set and the test set for $k = 1, 3, 5, 7$.

Deliverables: source code of your nearest neighbor classifier; training and test

results of your k-NN classifier for$k = 1, 3, 5, \ldots, 7$, short discussion of the results

II.2.3 Changing the metric

The function

$$d(\boldsymbol{x}, \boldsymbol{z}) = \|\boldsymbol{M}\boldsymbol{x} - \boldsymbol{M}\boldsymbol{z}\| \tag{II.2.1}$$

with

$$\boldsymbol{M} = \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix} \tag{II.2.2}$$

and $\|\cdot\|$ denoting the standard L_2-norm is a metric on $\mathbb{R}^2$. Proof this statement.

Deliverables: proof that d is a metric

II.2.4 Nearest neighbor with non-standard metric

Use the metric d, defined by eqs. (II.2.1) and (II.2.2), as a metric in your k-NN classifier and apply the resulting classifier to the Iris data as in Assignment II.2.2. Measure training and test error for$k = 1, 3, 5, \ldots, 7$,

Compare your results with the results achieved in Assignment II.2.2. Discuss the results in detail. Explain the performance differences and describe the effect of applying $\boldsymbol{M}$.

Deliverables: training and test results of your k-NN classifier for$k = 1, 3, 5, \ldots, 7$ using the metric d; discussion of the results, reasons for performance differences, explanation of the effect of the transformation by $\boldsymbol{M}$ on the input data

II.2.5 LDA with rescaled data

What happens if you apply $\boldsymbol{M}$ to the Iris data before you use LDA? That is, every training and test pattern $(\boldsymbol{x}, y) \in \mathbb{R}^2 \times \{0, 1, 3\}$ is replaced by $(\boldsymbol{M}\boldsymbol{x}, y)$ and LDA is applied to the transformed data. How do training and test error change? Why?

Deliverables: training and test error on the transformed data, discussion of the results and explanation of the effect of the transformation by $\boldsymbol{M}$

References

[1] E. Anderson. The species problem in iris. *Annals of the Missouri Botanical Garden*, 23(3):457–509, 1936.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[3] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.