Statistical Methods for Machine Learning 2015, Assignment II:

Basic Supervised Learning Algorithms

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

The goal of this assignment is to get familiar with basic supervised machine learning algorithms.

You have to pass this and the other 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 **Tues. 03/03/2015**. 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 consists of:

- Your solution source code (Matlab / R / Python scripts or C / 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, etc.
- Your code should also include a README text file describing how to compile and run your program, as well as a 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 Classification

We consider the data sets `IrisTrain2014.dt` and `IrisTest2014.dt` from the previous assignment. These data sets have been generated from the famous Iris flower data set [1, 3]. However, instead of the original four input features only two are considered. Furthermore, one feature has been rescaled and some examples have been removed.

II.1.1 Linear discriminant analysis

In the previous assignment, we applied nearest neighbor classification, a non-linear, non-parametric method. Now we consider a linear, parametric model for classification.

Apply linear discriminant analysis (LDA) as described on the lecture slides to the training data set 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: Implementation of LDA or description of tool used for LDA; training and test error of LDA.

II.1.2 LDA and normalization

What happens if you normalize the Iris data to have zero mean and unit variance as in the previous assignment before you use LDA? How do training and test error change? Why?

Note that under the assumption that the variances are non-zero, the normalization is an affine linear mapping and a bijection.

Deliverables: Training and test error on the transformed data, discussion of the results and explanation of the effect of the transformation.

II.1.3 Bayes optimal classification and probabilistic classification

Let us consider a toy binary classification problem with a single-element input space $X = \{0\}$ and output space $Y = \{0, 1\}$. The label 1 shows up in three-fourths of all trials, and we are given a training set $S = \{(0, 0), (0, 1), (0, 1), (0, 1)\}$ reflecting this.

What is the Bayes optimal classifier? What is its (Bayes optimal) risk?

Now, someone comes up with the idea of using a probabilistic classifier. To this end, let's extend the concept of a hypothesis, which is a deterministic function according to our definition in the lecture, and consider a probabilistic classifier that predicts 0 with a probability of 0.25 and 1 with 0.75. That is, the output of the classifier given an input is a random variable.

What is the risk of this classifier? Note that to compute the risk, you have to compute the expectation over the probabilistic output of the classifier. Please write down this computation.

Deliverables: Description of Bayes optimal classifier, Bayes risk; risk of probabilistic classifier, explanation of how you computed it.

II.2 Regression: Sunspot Prediction

In the regression part of this assignment, we will study regression with linear models (as explained by Bishop [2] Sec. 3.1).

Background

The goal of the exercise is to predict the average number of sunspots. We consider data based on the yearly sunspot number data provided by the Sunspot Index Data Center (SIDC), see `http://sidc.oma.be`. On `http://www.icsu-fags.org/ps11sidc.htm` we find the following short introduction to the sunspot data:

> Sunspots are extended regions on the Sun with a strong magnetic field. They have a lower temperature (3500-4500 K) than the surrounding photosphere (5800 K). The sunspots radiate less energy than the undisturbed photosphere of the Sun and are therefore visible as dark spots on the surface of the Sun. Sunspots are observed with some regularity since 1700 and on a strict daily basis since 1849; the relative [...] number (defined as ten times the number of groups + the number of spots) shows an 11 year cycle detected by Schwabe in 1843. The sunspot number reflects the magnetic activity of the Sun,

which has a large impact on the magnetosphere of the Earth and is responsible for e.g. magnetic storms and polar lights.

In this assignment, you are going to predict the average number of sunspots in a year t based on the average numbers in the years $t-1$, $t-2$, $t-4$, $t-8$, and $t-16$.

The training data can be found in the file `sunspotsTrainStatML.dt` and the test data in the file `sunspotsTestStatML.dt`. Each row corresponds to a 5-dimensional observation $\mathbf{x}$ and, in the last column, the target variable denoted t in [2]. The training data consist of the years 1716–1915, and the test data consist of the years 1916–2011.

The goal of our modeling is to find a mapping $f\colon \mathbb{R}^5 \to \mathbb{R}$ for predicting the number of sunspots based on previous observations.

In the following we will consider the following three selections of observation variables:

Selection 1: Let $\mathbf{x}$ consist of the data in columns 3-4, hence the dimensionality of this subset is $D = 2$.

Selection 2: Let $\mathbf{x}$ consist of the data in column 5, hence the dimensionality of this subset is $D = 1$.

Selection 3: Let $\mathbf{x}$ consist of the data in column 1-5, hence the dimensionality of this subset is $D = 5$.

II.2.1 Maximum likelihood solution

We will start by trying to model the data set with linear regression as defined in (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 - 3 by constructing their corresponding design matrices as defined in (3.16). Train each of these models on the training set by finding the maximum likelihood (ML) estimate by using (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`. Inverting matrices with Shark is discussed in the LinAlg tutorial. In Python, `numpy.linalg.pinv` should do the trick.)

For the variable selection 2, plot the x and y variables of your training set together, along with the real and predicted target variables on the test set together.

Apply each model to the test set using (3.3) with the ML parameter estimate and compute and report the root mean square (RMS) error

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

To further validate the quality of your prediction, plot years versus predicted sunspot numbers for each model on the test set, along with the actual sun spot number.

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

II.2.2 Maximum a posteriori solution

Next let us try to learn the three models using Bayesian learning and maximum a posteriori (MAP) estimation. Let us fix the prior distribution on the parameters to the zero mean isotropic Gaussian as given in (3.52) in [2]. Set the noise precision parameter to $\beta = 1$. We may then obtain the posterior distribution as given in (3.49), by computing the estimates of the mean $\mathbf{m}_N$ and covariance $\mathbf{S}_N$ using (3.53) and (3.54). The posterior mean is the MAP estimate.

Deliverables: Using the MAP estimate and (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 three models seem to provide the best prediction? How do 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.2.1?

II.2.3 Weighted sum-of-squares (based on CB Ex. 3.3)

Consider a data set in which each data point t_n is associated with a weighing factor $r_n > 0$, so that the sum-of-squares error function becomes

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N} r_n\{t_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\}^2.$$

Deliverables: Find an expression for the solution $\mathbf{w}^*$ that minimizes this error function. Give two alternative interpretations of the weighted sum-of-squares error function in terms of (i) data dependent noise variance and (ii) replicated data points.

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.