Machine Learning 2015/2016, Assignment 1.1:

Classification and Concentration Bounds

Christian Igel, Yevgeny Seldin
Department of Computer Science, University of Copenhagen

The deadline for this assignment is **12:00 pm (noon, not midnight) 23/11/2015**. You must submit your individual solution electronically via the Absalon home page.

A solution consists of:

- A PDF file with detailed answers to the questions, which may include graphs and tables if needed. Do *not* include your source code in this PDF file.
- 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). Source code must be submitted in the original file format - NOT in PDF.
- 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 list of all relevant libraries needed for compiling or using your code.

1 Classification

Please download the data sets `IrisTrainML.dt` and `IrisTestML.dt` from the course homepage. These data sets have been generated from the famous Iris flower data set [2], 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 is measured in millimeters and the width in centimeters. The last column encodes the species.

Figure 1: An example of an Iris versicolor taken from Wikipedia.

You may find some of the algorithms you are supposed to implement (nearest neighbor classification, cross-validation, data normalization) useful later on in the course. Thus, a reliable, general implementation is recommended.

1.1 Nearest neighbor

Nearest neighbor classification is the fundamental non-linear, non-parametric method for classification. Implement a k-nearest neighbor classifier (k-NN). Use the data `IrisTrainML.dt` as training data and report the accuracy (one minus the 0-1 loss) of the corresponding classifier on both the training data and the test

data in `IrisTestML.dt` for $k = 1, 3, 5$.

Deliverables: Source code of your nearest neighbor classifier; training and test results of your k-NN classifier for $k = 1, 3, 5$; short discussion of the results

1.2 Hyperparameter selection using cross-validation

The performance of the nearest neighbor classifier depends on the choice of the parameter k determining the number of neighbors. Such a parameter of the learning *algorithm* (i.e., not a parameter of a resulting model) is called a *hyperparameter*.

Cross-validation is useful to determine proper hyperparameters. You are supposed to find a good value for k from $\{1, 3, 5, \ldots, 25\}$. For every choice of k, estimate the performance of the k-NN classifier using 5-fold cross validation. Pick the k with the lowest average 0-1 loss (classification error), which we will call k_{best} in the following. Only use the training data `IrisTrainML.dt` in the cross validation process to generate the folds.

After you have determined k_{best}, you can estimate the performance of the classifier using the test data `IrisTestML.dt`.

> As a general rule, you must not use the test data in the model building process at all (neither for training, data normalization, nor hyperparameter selection), because otherwise you may get a biased estimate of the generalization performance of the model (see [1, Example 5.3]).

To estimate the generalization performance, build a k_{best}-NN classifier using the complete training data set `IrisTrainML.dt` and evaluate it on the independent test set `IrisTestML.dt`.

Deliverables: Source code; a short description of how you proceeded (e.g., did the cross-validation); number of neighbors as suggested by hyperparameter selection; training and test error of k_{best}-NN

1.3 Data normalization

Data normalization is an important preprocessing step. A basic normalization is to generate zero-mean, unit variance input data.

Consider the *training* data in `IrisTrainML.dt`. Compute the mean and the variance of every input feature (i.e., of every component of the input vector). Find the affine linear mapping $f_{\text{norm}} : \mathbb{R}^2 \to \mathbb{R}^2$ that transforms the training data such that the mean and the variance of every feature in the transformed data are 0 and 1, respectively (verify by computing these values).

Use the same function f_{norm} to also encode the test data. Compute the mean and the variance of every feature in the transformed test data.

The normalization is part of the model building process. Thus, you may only use the training data for determining f_{norm} (always remember that you are supposed to not know the test data).

Now repeat exercise 1.2 with the normalized data instead of the raw data. Perform cross-validation and report the training and test error using the value of k found by the cross-validation procedure. Can you explain the differences compared to the results achieved using the raw (not normalized) data?

Deliverables: Mean and variance of the training data; mean and variance of the transformed test data; number of neighbors k_{best} as suggested by hyperparameter selection; training and test error of k_{best}-NN considering the normalized data; short discussion of the differences in performance when using normalized and raw data

2 Probability theory refreshment

An urn contains five red, three orange, and one blue ball. Two balls are randomly selected (without replacement).

1. What is the sample space of this experiment?
2. What is the probability of each point in the sample space?
3. Let X represent the number of orange balls selected. What are the possible values of X?
4. Calculate $\mathbb{P}\{X = 0\}$.
5. Calculate $\mathbb{E}[X]$.

3 Probability theory refreshment

Let X and Y be two discrete random variables taking values in $\mathcal{X}$ and $\mathcal{Y}$, respectively. Let p_X be the distribution of X, p_Y the distribution of Y, and p_{XY} the distribution of X and Y. In other words, $\mathbb{P}\{X = x\} = p_X(x)$, $\mathbb{P}\{Y = y\} = p_Y(y)$, and $\mathbb{P}\{(X = x) \text{ AND } (Y = y)\} = p_{XY}(x, y)$. A convenient way to represent a joint probability distribution of two discrete random variables is a table. For example, if X and Y are two fair coins then the joint distribution table looks

like this:

$X\backslash Y$	0	1
0	1/4	1/4
1	1/4	1/4

. And if Z_1 and Z_2 are two fair coins and we define $X = Z_1 + Z_2$ and $Y = Z_1 \times Z_2$ then the joint distribution of X and Y is:

$X\backslash Y$	0	1
0	1/4	0
1	1/2	0
2	0	1/4

.

We remind you the following properties and definitions from the probability theory:

(a) $p_X(x) = \sum_{y \in \mathcal{Y}} p_{XY}(x, y)$.

(b) If X and Y are independent then $p_{XY}(x, y) = p_X(x) p_Y(y)$.

(c) $\mathbb{E}[X] = \sum_{x \in \mathcal{X}} x p_X(x)$.

Starting from the definitions, prove the following identities:

1. $\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$.
2. If X and Y are independent then $\mathbb{E}[XY] = \mathbb{E}[X]\,\mathbb{E}[Y]$. (Mark the step where you are using the independence assumption. Note that this assumption was not required in point 1.)
3. Provide an example of two random variables X and Y for which $\mathbb{E}[XY] \neq \mathbb{E}[X]\,\mathbb{E}[Y]$. (Describe how you define the random variables, provide a joint probability distribution table, and calculate $\mathbb{E}[XY]$ and $\mathbb{E}[X]\,\mathbb{E}[Y]$.)
4. $\mathbb{E}[\mathbb{E}[X]] = \mathbb{E}[X]$.
5. Variance of a random variable is defined as $\mathbb{V}[X] = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right]$. Show that $\mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$.

4 Markov's inequality vs. Hoeffding's inequality vs. binomial bound

Let $X_1, \ldots, X_{10}$ be i.i.d. Bernoulli random variables with bias $\frac{1}{2}$. (I.e., $\mathbb{P}\{X_i = 0\} = \mathbb{P}\{X_i = 1\} = \frac{1}{2}$.)

1. Use Markov's inequality to bound the probability that $\sum_{i=1}^{10} X_i \geq 9$. (Hint: you have to define a new random variable $S = \sum_{i=1}^{10} X_i$ and apply Markov's inequality to S.)

2. Use Hoeffding's inequality to bound the probability of the same event. (Hint: now you do not need to group the variables together and you can exploit their independence.)

3. Calculate the exact probability of the above event. (Hint: S has binomial distribution.)

4. Compare the three results.

5 Hoeffding's inequality

1. An airline knows that any person making a reservation on a certain flight will not show up with a probability of 0.05 (5 percent). They introduce a policy to sell 100 tickets for a flight that can hold only 99 passengers. Bound the probability that the number of people that show up for a flight will be larger than the number of seats.

2. An airline has collected a sample of 10000 flight reservations and figured out that in this sample 5 percent of passengers who made a reservation on a certain flight did not show up. They introduce a policy to sell 100 tickets for a flight that can hold only 99 passengers. Bound the probability that the number of people that show up for a flight will be larger than the number of seats.

References

[1] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin. *Learning from Data*. AMLbook, 2012.

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

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