Tutorial - Exercise sheet 1

Pattern and Speech Recognition

Introduction

The goal of the first tutorial is to introduce you to basic probability concepts and linear classification. In order to do so, we will manipulate the `iris` dataset. You can find a full description of this dataset on the following wikipedia webpage : `http://en.wikipedia.org/wiki/Iris_flower_data_set`

Ex. 1 — Retrieve the `iris` dataset from this URL:
`http://ftp.ics.uci.edu/pub/machine-learning-databases/iris/iris.data`

1 Sets and probability

First of all, we consider the probability space Ω as the set of instances from the dataset and only these ones.

Ex. 2 — How many instances labeled `Iris-setosa` are present in the dataset? Same question considering the label `Iris-versicolor` and the label `Iris-virginica`. From the achieved results, can you deduce $P(label = \text{Iris-setosa})$? (Same question for the other labels)

Ex. 3 — We are now considering the *sepal width* (the second column of the dataset). How many instances have their sepal width greater than 4 ? Deduce the probability $P(\text{sepal width} > 4)$.

Ex. 4 — Same questions considering that the *sepal length* (the first column) has to be less than 6.

Ex. 5 — We are now considering the 3 following events A, B and C:

- A which represents `label=Iris-setosa`,
- B which represents `sepal width>4`,
- C which represents `sepal length<6`.

Represent graphically the following events :

- $\bar{A}$ = instances whose label is not `Iris-setosa`,
- $B \cup C$ = instances whose sepal width is greater than 4 and the sepal length less than 6,
- $B \cap C$ = instances whose sepal width is greater than 4 or the sepal length less than 6,
- $B \cup \bar{A}$ = instances whose sepal width is greater than 4 but whose label is not `Iris-setosa`.

From these, deduce a way to compute $P(\bar{A})$, $P(B \cup C)$, $P(B \cap C)$ and $P(B \cup \bar{A})$

Ex. 6 — **Bonus question** - How can you proceed to compute the probability of the event C when you know that the label of the instances is `label=Iris-setosa` ? What is the name of this kind of probability ?

2 Linear classification

The goal of this part is to introduce basic linear classification. At the end of this part, you should be able to understand:

- what is the objective of the classification task,
- how linear classification is working,
- determinate if you should use a linear classification or not.

2.1 An introduction to linear classification

The goal of the exercise is to use a linear classifier to predict the label of a given flower. In order to simplify the problem, we are going to consider a subset of instances qualified by a subset of properties. Therefore we consider two possible labels: `Iris-setosa` and `Iris-virginica`. We also consider two properties to qualify an Iris: the *petal length* (column 3) and the *petal width* (column 4).

Data preparation

Ex. 7 — We don't want to consider `Iris-versicolor` instances. Filter them from the dataset

Ex. 8 — We qualify an instance only using the *petal length* and the *petal width*. Filter the other properties from the dataset. **You should keep the label (last column)**

Ex. 9 — Plot the result dataset in a 2-D plot using points. (By convention, we are considering the *petal length* as the x-axis and the *petal width as the y-axis*)

Ex. 10 — Improve the plots by rendering in red the instances labeled `Iris-setosa` and blue the ones labeled `Iris-virginica`.

Ex. 11 — What can we conclude from this plot?

From now, we called the obtained dataset Ω.

Using the classifier require two datasets: a training corpus (L) to train the classifier, a test corpus (T) to estimate the quality of the classifier. These corpora respect the following constraints:

- the instances should be randomized,

- you should follow the same distribution than the full corpus Ω. In our case, each class has the same probability (see first section results). Therefore, your training corpus should as many instances of `Iris-setosa` as instances of `Iris-virginica`,
- the test corpus size should be significantly smaller than the training corpus. In our case, we consider the size of the test corpus of 10% of the size of the full corpus (= 10 instances).

Ex. 12 — From Ω, generate the training corpus and the test corpus.

Training the classifier

Now that we have prepared the data, we want to train a classifier. To achieve this part, we need to introduce some definitions.

Definition 1. We are considering a couple of values (x, y). x represents the *petal length* and y the *petal width.*

Definition 2. We define the *linear decision boundary* by the following equation:

$$ax + b = y \tag{1}$$

where a and b are the parameters of the linear classification.

Definition 3. We define by *criterion* an equation which leads to take a decision.

In our case, the criterion is expressed as the following :

$$ax + b \begin{cases} < y & \text{labeled as virginica} \\ > y & \text{labeled as setosa} \end{cases} \tag{2}$$

The goal of the linear classifier training part is to estimate the most accurate values for (a, b)

Ex. 13 — Explain why the criterion is working? What can you propose to deal with the case $ax + b = y$? Is it useful in the current dataset ?

In order to compute the parameters, we will use a *brute force* method called *grid search.* We set an interval for a (e.g $[min_a, max_a]$) and another one for b (e.g $[min_b, max_b]$). Then we search, within these intervals, the parameters that give the minimum number of misclassified instances.

Create a for loop for a with small increments (step is noted i_a) and another for loop **within** the first loop for b (step is noted i_b). For every combination of a and b, compute the misclassification error over all training instances.

After finishing with the iterations, the parameters you found with the minimum error so far should be your classifier parameters.

Ex. 14 — Implement the algorithm and save the trace into a text file. The trace should contains the value for (a, b) and corresponding misclassification error value. For now, we consider the following values

- $min_a = min_b =$ -3
- $max_a = max_b = 3$

- $i_a = i_b = 0.7$

Ex. 15 — Plot the resulting linear decision boundary and the instances on the *same figure*. Analyze the results

Ex. 16 — Try different combination of values for $\{i_y, min_y, max_y\}$ where $y \in \{a, b\}$. Report 3 different cases which are significant and explain how these values can influence the classification.

Prediction and validation

Now that we have a valid classifier, it is time to use it.

Ex. 17 — Evaluate your classifier with the test dataset. Plot the test instances and the decision boundary, and report the error as well.

Ex. 18 — Considering the following list of instances, use your classifier to find the label for each one of them:

Petal length	Petal width
0	2
6.0	2.5
0	2
6.0	-3
6.0	-2
0	4
3.0	-3

Table 1: Some unlabeled instances

2.2 Bonus exercise: is linear classification always applicable ?

Ex. 19 — Retrieve the *obfuscated* data set from this url:
http://www.coli.uni-saarland.de/~slemaguer/teaching/obfuscated.data

Ex. 20 — Apply the procedure previously described. Is linear classification adapted? Why? Represent graphically how you would classify (You can use gimp to draw your solution on top of the plot of the instances)?