Machine Learning Pattern Recognition: Lab 2

Nicholas Piël
npiel@science.uva.nl
cknr: 0104612

2007-09-18

1 Hagelslag (introduction)

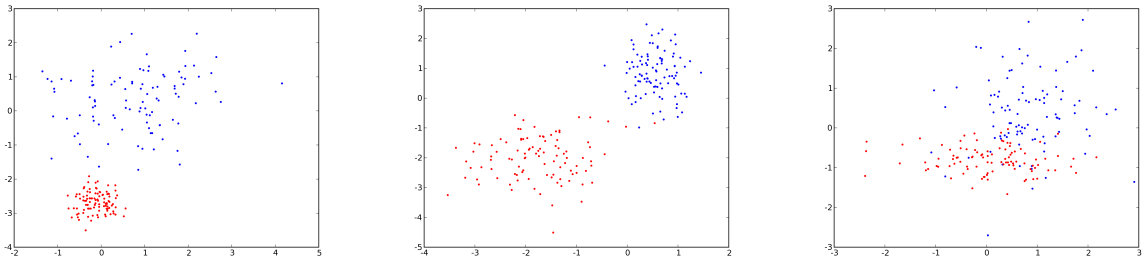
We can see the classification problem as some kind of kitchen accident. Let's say you accidentally dropped 2 packs of hagelslag on to the kitchen floor, one low quality brand of EuroShopper hagelslag which tastes a bit rancid and one delicious pack of the best quality chocolate sprinkles you can find. While the difference in taste is extreme they look very much alike. Now how do you separate the mess on the floor without actually eating all the hagelslag?

In this lab exercise we try to tackle this problem by creating a Bayesian Classifier which can classify data by assuming a Gaussian distribution. This classifier had to be implemented in a combination of matrix manipulations and while the recommended toolkit to implement this was MatLab, I used Python with NumPy for this. But this is not of any importance for the rest of report.

2 Data Generation

First we need to generate some data, before we can actually cluster over it. We do this by randomly sprinkling points. This is done by choosing an arbitrary point where we drop our 'Hagelslag' and a random covariance matrix which specifies how scattered the data will be.

Some examples of the data we are coping with can be found in figure 1



(a) Cleanly separated (b) Connecting (c) Miked Up

Figure 1: Various Distributions

3 Bayes Classifier

In order to be able to classify over this data we separate it in a training and testing set. From this training set we create an estimate of the distribution by taking the mean μ and the variance σ from these we are able to create a Gaussian Distribution.

We can then use this distribution as our probability density function. Then for a certain point x given a certain class C_1 from which we sample μ and σ the following holds:

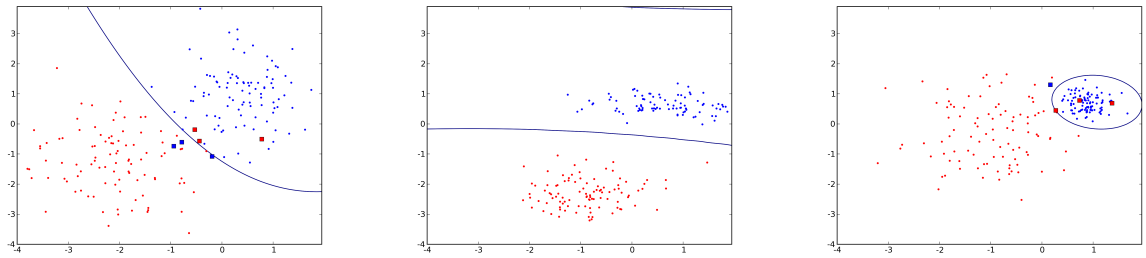
$$P(C_1|x) = gauss(\mu_{c_1}, \sigma_{c_1}, x) \tag{1}$$

Since we know that the priors of both classes are equal (the packs of hagelslag contained the same amount of sprinkles) we can therefor easily assign labels to the data by simply comapring the class likelyhood. The pseudo code for the label assigner has been depicted below.

Listing 1: cap

```
labels <- []
for datapoint in datapoints:
        prob_c1 = gauss(mu_a, sigma_a, datapoint)
        prob_c2 = gauss(mu_b, sigma_b, datapoint)
        if prob_c1 > prob_c2:
                label <- class1
        else:
                label <- class2
        labels.append(label)
return labels
```

When we plot the result of various distributions with its boundary and in such a way that misclassified points are depicted with a square the result can be seen in figure 2



(a) Three datapoints of each class are misclassified

(b) Clean seperation no misclassfications

(c) 3 misclassification in one set and one in the other

Figure 2: Classification of various distributions