



Prof. Dr. Steffen Wagner
Angewandte Statistik
Fachbereich II
Berliner Hochschule für Technik

Exercise 08: Classification with LDA/QDA

Machine Learning I – SoSe 24

1 Preparations	2
1.1 RStudio Project	2
1.2 Required Packages	2
1.3 Required Data	2
2 Bayes Classifier	3
2.1 Bayes Classifier by hand	3
2.2 Posterior function in R	4
2.3 LDA and QDA with the Diabetes Data	6



This workshop covers hierarchical clustering and soft clustering. At the end of the worksheet there are a couple of written exercises for you to do at home, which should be good practice for the exam.

1 Preparations

1.1 RStudio Project

1. Open your Machine Learning 1 RStudio Project
2. Create an R Script file to perform this exercise

1.2 Required Packages

For this exercise you require the following additional R packages. Please make sure that you have installed them on your computer before coming to the workshop session for the case that Eduroam is not working.

```
# check if packages can be loaded, i.e. they are already installed
library(ggplot2)           # for visualisation
library(MASS)              # for LDA and QDA
library(pROC)
```

If you get an error at this stage, you need to install the packages.

1.3 Required Data

In this Worksheet we will use again the data set `Diabetes.Rda` that is available via Moodle.



2/ Bayes Classifier

2.1 Bayes Classifier by hand

Let Y be a random variable, which takes the values 0 or 1, dependent on a predictor variable x . Assume that, if $Y=0$ then $X|Y=0$ is $N(4, 1)$ distributed, and if $Y=1$ then $X|Y=1$ is $N(5, 1)$ distributed. The prior probabilities, when x is unknown, are $P(Y=0) = P(Y=1) = 0.5$

Tasks:

- a) Write down the formula for $\phi_0(x)$, the density of $X|Y=0$ and for $\phi_1(x)$, the density of $X|Y=1$. Hint: The general formula for a normal distribution can be found on Slide 8 of Lecture 4. [formula of the normal dist with \(4,1\) & \(5,1\)](#)
- b) Write down the expression for the posterior probability $\pi_1(x) = P(Y=1|x)$ and simplify as much as possible.
- c) Check that the Bayes classifier corresponds to: classify Y equal to one if and only if $P(Y=1|x) > P(Y=0|x)$.
- d) Use your answer from part (b) to write $P(Y=1|x) > P(Y=0|x)$ as an inequality in terms of x . Simplify to obtain the inequality

$$\exp\left\{-\frac{1}{2}(x-5)^2\right\} > \exp\left\{-\frac{1}{2}(x-4)^2\right\}.$$

- e) Taking the logarithm of this inequality, show that the Bayes Classifier simplifies to: classify Y equal to one if and only if $x > 4.5$.



2.2 Posterior function in R

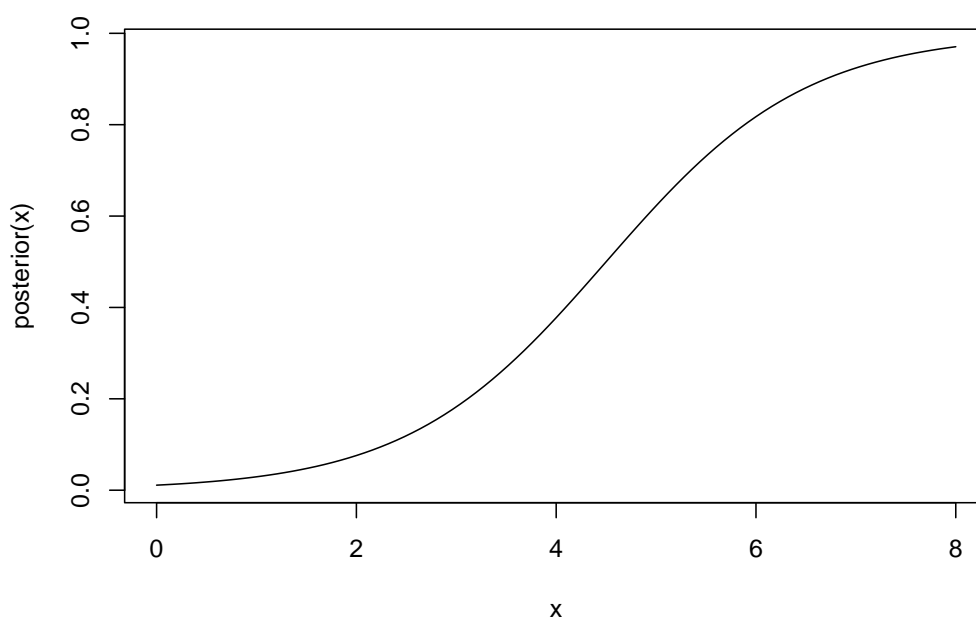
Use your answer from Exercise 1 Part (b) to write an R function called `posterior` to compute the posterior probability of $P(Y=1|x)$.

You will start by assuming the same model as in Ex 1, and then generalise it to general π_0 , μ_0 , μ_1 and σ .

- a) You can use the function `dnorm(x, mean = , sd =)` to compute the density of a normal distribution. `x` should be an argument to the function `posterior` so your function should use the following template:

```
posterior <- function(x){  
  ?????  
}
```

- b) Plot the function using the R function `curve()` for `x` values from 0 to 8 so that you obtain



In Exercise 1 you showed that the most-likely-outcome changes at the point $x=4.5$.

- c) Use `posterior(4.5)` to find the posterior probability at $x=4.5$. Why is this result “obvious”?
- d) Now adapt your function `posterior` to accept the following *function arguments* with the given default values.
- `pi0` is the prior probability $P(Y=1)$ with default value 0.5
 - `mu0` and `mu1` are the respective means for class 0 and class 1 with default values 4 and 5.
 - `sigma` the variance (in both classes) with default value 1.
- e) Check that your function gives sensible results by plotting the function with different



argument values. If you want to use `ggplot2::ggplot()` instead of base R graphics use the following code:

```
library(ggplot2)
ggplot() +
  geom_function(fun = posterior,
               args = list(pi0 = ???,
                           mu0 = ???,
                           mu1 = ???,
                           sigma = ???)) +
  scale_x_continuous(limits = c(0, 8))

# with base R graphics this can be achieved via
curve(posterior(x, pi0 = ???, mu0 = ???, mu1 = ???, sigma = ???),
      from = 0, to = 8)
```



2.3 LDA and QDA with the Diabetes Data

In this exercise we will work again with the `Diabetes` data set applied to classification by logistic regression. This week you will use linear and quadratic discriminant analysis. You will use the functions `lda` and `qda` from the `MASS` package.

a) Preparations:

- Download the `Diabetes` dataset from Moodle and the R code template `Classification_Diabetes.R` you used last week.
- Use the template to split the data into the *exactly same* training and test data sets as last time.
- Delete all the code beginning with section # 03b: `model training ----` and save the R file with a new file name, e.g. `Classification_Diabetes_LDA_QDA.R`.

b) Instead of applying logistic regression to classify the data you will use LDA and QDA. For using an LDA model with only one variable, e.g. `Age` use the following code:

```
library(MASS)
lda.fit1 <- lda(YN ~ Age, data = train)
```

To assess the classification quality use

```
library(pROC)
prtest <- predict(lda.fit1, newdata=test)
roc.obj1 <- roc(test$YN, prtest$posterior[, 2])
ggroc(roc.obj1)
auc(roc.obj1)
```

- c) Adapt the code to fit the following discriminant models, each time plotting the ROC curve and the obtaining the AUC.
- LDA model using `BMI`
 - LDA model using `Age` and `BMI`
 - QDA model using `Age` and `BMI`
- d) Which model gives the best AUC on the test data? Compare the LDA/QDA model results also with last weeks logistic regression model using `Age` and `BMI`.

The given code in this exercise should be enough for you to fit the LDA and QDA models, but further help can be found in Labs 4.7.3 & 4.7.4 in James et. al.