EPFL, Autumn 2024
Yoav Zemel — Ho Yun
Guillaume Obozinski — Shivang Sachar

Statistical Machine Learning

Exercise sheet 6

Exercise 6.1 (Multiclass Logistic Regression) In this exercise we derive a multiclass generalization of logistic regression. We shall assume that the input variable X is a vector in $\mathbb{R}^p$ as before, but the output variable Y is a vector of the form $y = (y_1, \ldots, y_K) \in \{0,1\}^K$ with $\sum_{k=1}^K y_k = 1$. Thus, the vector y such that $y_k = 1$ for $k = m$ and 0 otherwise, corresponds to the class m. It follows that the probability of X being in class m given $X = x$ is $\mathbb{P}(Y_m = 1|X = x)$, where $Y = (Y_1, \ldots, Y_K)$.

(a) Let $w_1, \ldots, w_K \in \mathbb{R}^p$ be K vectors of parameters each associated with the corresponding class. Construct a conditional model for $Y = y|X = x$ such that $\mathbb{P}(Y_k = 1|X = x) \propto \exp(w_k^\top x)$. In particular, find $\mathbb{P}(Y_k = 1|X = x)$.

(b) Show that when $K = 2$, the proposed model is equivalent to logistic regression, except that the model is over-parameterized, and therefore w_1 and w_2 are not identifiable. Is this a problem?

(c) Show that the model is still overparametrized if $K > 2$ and that one can impose the constraint $\sum_{k=1}^K w_k = 0$.

(d) Express $\mathbb{P}(Y_k = 1|Y_k + Y_j = 1, X = x)$, or alternatively, derive the log-odds between two classes. What is the shape of $\{x \mid \mathbb{P}(Y_k = 1|X = x) = \mathbb{P}(Y_j = 1|X = x)\}$? Deduce that the region of space where class k is most likely is a polyhedron.

(e) Assume that we have a sample $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$ with $(x^{(i)} \in \mathbb{R}^p$ and $y^{(i)}$ an indicator vector. Write the negative (conditional) log-likelihood of the sample and show that it can be interpreted as the empirical risk associated with a loss function that you will specify $\ell : \mathbb{R}^K \times \{0,1\}^K \to \mathbb{R}$ applied to a predictor $f(x)$ of the form $f(x) = (f_1(x), \ldots, f_K(x))$ with $f_k(x) = w_k^\top x$.

(f) How would you apply Tikhonov regularization to the corresponding empirical risk?

(g) Since the model is over-parameterized, instead of using the strategy proposed in (c), one could propose to just set $w_K = 0$. Prove that this would yield an equivalent model.

(h) Now, if we use Tikhonov regularization, why is the option to set $w_K = 0$ not such a good idea?

(i) Why is the option proposed in (c) better? If we regularize with Tikhonov regularization and don't enforce the constraint $\sum_{k=1}^K w_k = 0$, what happens?

Practical exercises

Exercise 6.2 (Implementation of the LDA and QDA algorithms and comparison with logistic regression) The files `classificationA.train`, `classificationB.train` and `classificationC.train` contain samples of data (x_i, y_i) where $x_i \in \mathbb{R}^2$ and $y_i \in \{0, 1\}$ (each line of each file contains the 2 components of x_i then y_i.). The goal of this exercise is to implement linear classification methods and to test them on the three data sets.

(a) For each data set (A,B,C) represent graphically the training data as a point cloud in $\mathbb{R}^2$ using different markers for the two classes using `ggplot`,

```
#Input
inp <- scan("classificationA.train", list(x1=0,x2=0,y=0))
inp <- data.frame(x1=inp$x1,x2=inp$x2,y=inp$y)
#Base Plot
G <- ggplot(data = inp, mapping = aes(x = x1, y = x2,
     color = as.factor(y)))+ geom_point()
```

(b) Apply LDA and compute the MLE estimates for all the parameters. Plot the classification boundary for LDA for each data set by completing the following `R` code by entering correct values for `b` and `S` which are determined by the fact that the boundary is given by $\mathtt{w}[1]\mathtt{x1} + \mathtt{w}[2]\mathtt{x2} + \mathtt{b} = 0$.

```
#LDA Boundary Parameters
b <- << ENTER b HERE >>
w <- << ENTER w HERE >>

#LDA Boundary Function: x2 = (- x1*w[1] + b)/w[2]
LDAcurve <- function(x) {
    (- x*w[1] + b)/w[2]
}
#Plot with LDA boundary
Graph_LDA <- G
    + stat_function(fun = LDAcurve, color = "black")
```

(c) On a separate figure, plot the classification boundary for QDA, on top of the data for each data set by completing the following code. Because we do not have a handy expression for the graph of the boundary, say $\mathtt{f}(\mathtt{x1}, \mathtt{x2}) = 0$, we shall draw it as a contour of $\mathtt{f}(\mathtt{x1}, \mathtt{x2}) = \mathtt{z}$. Enter the expression from $\mathtt{f}(\mathtt{x1}, \mathtt{x2})$ below.

```
#QDA Contour
cont_QDA <- curve3d(<< ENTER FUNCTION f(x1,x2) HERE >>,
from = c(-6,-6), to = c(6,6), n=c(100,100),
sys3d="none")
dimnames(cont_QDA$z) <- list(cont_QDA$x,cont_QDA$y)
M_QDA <- reshape2::melt(cont_QDA$z)
#Plot with QDA boundary
Graph_QDA <- G
```

```
+ geom_contour(data=M_QDA,
aes(x=Var1,y=Var2,z=value),
breaks=0,linejoin = "round",colour="black")
```

(d) Run logistic regression using the `glm` function in `R`, and make again a similar plot with the data and the decision boundary using `stat_function` in `ggplot`.

```
#Logistic Regression
logres <- glm(y ~ x1 + x2, data = inp, family = binomial)
summary(logres)$coef

#Logistic Regression Coefficients
m1 <- summary(logres)$coef[[2,1]] #Coefficient of x1
m2 <- summary(logres)$coef[[3,1]] #Coefficient of x2
mc <- summary(logres)$coef[[1,1]] #Constant term

#Logistic Regression Boundary Function: f(x) =  -(mc + m1 * x1)/m2
logcurve <- function(x) {
<< ENTER CODE HERE >>
}
```

Are the coefficients very large? If so, why?

(e) Do the same visualizations on the three testing data sets.

(f) Compute the misclassification error of all three methods on all the three training sets and their corresponding testing sets. Which method performs better and why?