P9120 HW3 answer

Guojing Wu / UNI: gw2383

10/27/2019

1. Suppose $X \in \mathbb{R}^p$ and $Y \in -1, 1$. For any real-valued function f on $\mathbb{R}^p$, let L(Y,f(X)) denote the loss function for measuring errors between Y and f(X). Let $f^* = argmin_f EL(Y, f(X))$, where the expectation is taken over the joint distribution of X and Y. Show that:

(a) (Logistic Regression) If $L(y, f(x)) = log[1 + exp(-yf(x))]$, then $f^*(x) = log\frac{Pr(Y=1|X=x)}{Pr(Y=-1|X=x)}$.

In general, the goal of this learning problem is to minimize the expected risk: $EL(Y, f(X)) = \int_{X\times Y} L(y, f(x))Pr(x, y)dxdy$, where $Pr(x, y) = Pr(y|x)Pr(x)$ and $L(y, f(x)) = \phi(yf(x))$, so we can rewrite the expected risk as:

$$\begin{aligned} EL(Y, f(X)) &= \int_{X\times Y} L(y, f(x))Pr(x, y)dxdy \\ &= \int_X \int_Y \phi(yf(x))Pr(y|x)Pr(x)dxdy \\ &= \int_X [\phi(f(x))Pr(1|x) + \phi(-f(x))Pr(-1|x)]Pr(x)dx \\ &= \int_X [\phi(f(x))Pr(1|x) + \phi(-f(x))(1 - Pr(1|x))]Pr(x)dx \end{aligned}$$

Here we set $\eta = Pr(1|x)$, and by taking the functional derivative of the term $[\phi(f(x))Pr(1|x) + \phi(-f(x))(1 - Pr(1|x))]$ with respect to f and setting the derivative equal to 0 we get:

$$\frac{\partial\phi(f)}{\partial f}\eta + \frac{\partial\phi(-f)}{\partial f}(1 - \eta) = 0 \tag{1}$$

For question (a), the $\phi(yf(x)) = log[1 + exp(-yf(x))]$, hence $\phi(f) = log[1 + exp(-f)]$, solving equation (1) we get:

$$\begin{aligned} \frac{-e^{-f^*}}{1 + e^{-f^*}}\eta + \frac{e^{f^*}}{1 + e^{f^*}}(1 - \eta) &= 0 \\ \frac{e^{f^*}}{1 + e^{f^*}} &= \eta \\ f^* &= log\frac{\eta}{1 - \eta} \end{aligned}$$

So the minimizer is $f^*(x) = log\frac{Pr(Y=1|X=x)}{1-Pr(Y=1|X=x)} = log\frac{Pr(Y=1|X=x)}{Pr(Y=-1|X=x)}$.

(b) (SVM) If $L(y, f(x)) = [1 - yf(x)]_+$, then $f^*(x) = sign[Pr(Y = 1|X = x) - \frac{1}{2}]$.

Take the derivative of $\phi(f) = [1 - f]_+$ and $\phi(-f) = [1 + f]_+$ we get:

$$\phi'(f) = \begin{cases} -1, & f < 1 \\ 0, & f \geq 1 \end{cases}$$

$$\phi'(-f) = \begin{cases} 1, & f > -1 \\ 0, & f \leq -1 \end{cases}$$

We try to solve equation (1) under several cases:

- when $-1 < f^* < 1$, we have $-\eta + 1 - \eta = 0$, so $\eta = \frac{1}{2}$, but the solution is undefined
- when $f^* = -1$, we have $-\eta + 0 = 0$, so $\eta = 0$
- when $f^* = 1$, we have $0 + 1 - \eta = 0$, so $\eta = 1$

Combining all the cases above, we get $f^*(x) = sign[Pr(Y = 1|X = x) - \frac{1}{2}]$

(c) (Regression) If $L(y, f(x)) = [y - f(x)]^2$, then $f^*(x) = 2Pr(Y = 1|X = x) - 1$.

From text book, the squared error $[y - f(x)]^2 = [1 - yf(x)]^2$, so $\phi(f) = [1 - f]^2$. Solving equation (1) we get:

$$\begin{aligned} -2(1 - f^*)\eta + 2(1 + f^*)(1 - \eta) &= 0 \\ f^* &= 2\eta - 1 \end{aligned}$$

So the minimizer is $f^*(x) = 2Pr(Y = 1|X = x) - 1$.

(d) (AdaBoost) If $L(y, f(x)) = exp[-yf(x)]$, then $f^*(x) = \frac{1}{2} log \frac{Pr(Y=1|X=x)}{Pr(Y=-1|X=x)}$.

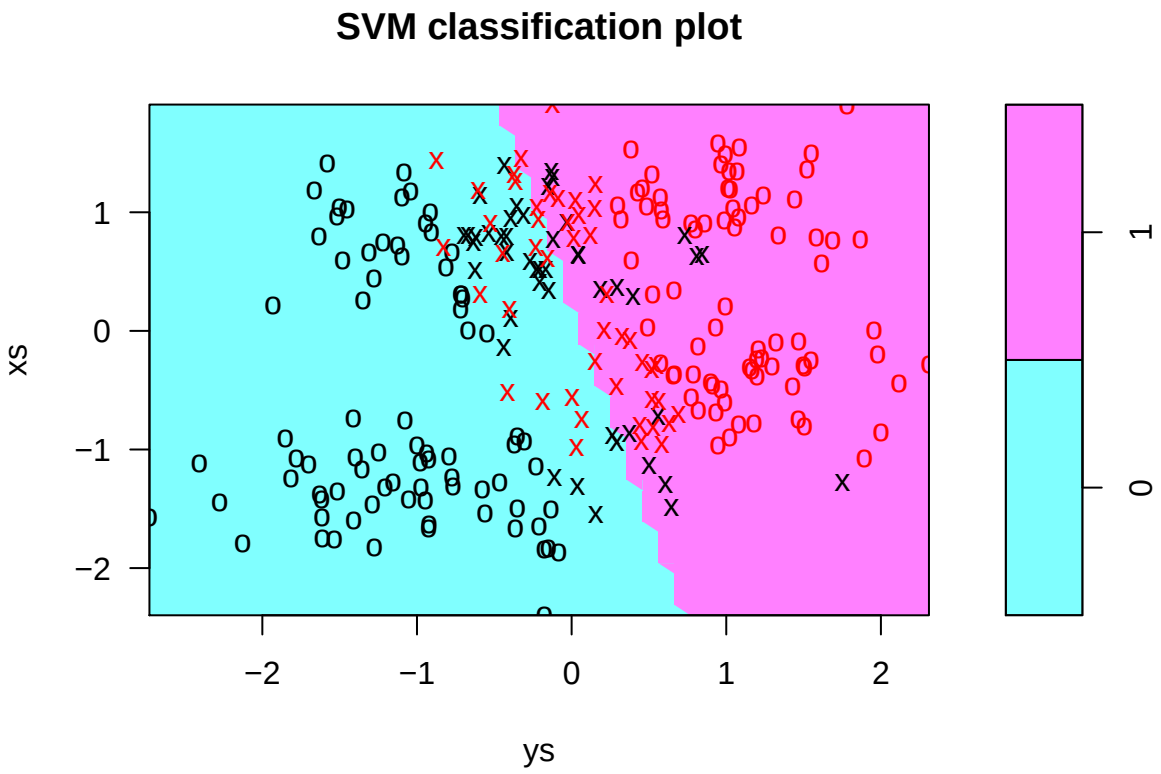
$\phi(f) = e^{-f}$. Solving equation (1) we get:

$$\begin{aligned} -e^{-f^*}\eta + e^{f^*}(1 - \eta) &= 0 \\ e^{2f^*}(1 - \eta) &= \eta \\ f^* &= \frac{1}{2} log \frac{\eta}{1 - \eta} \end{aligned}$$

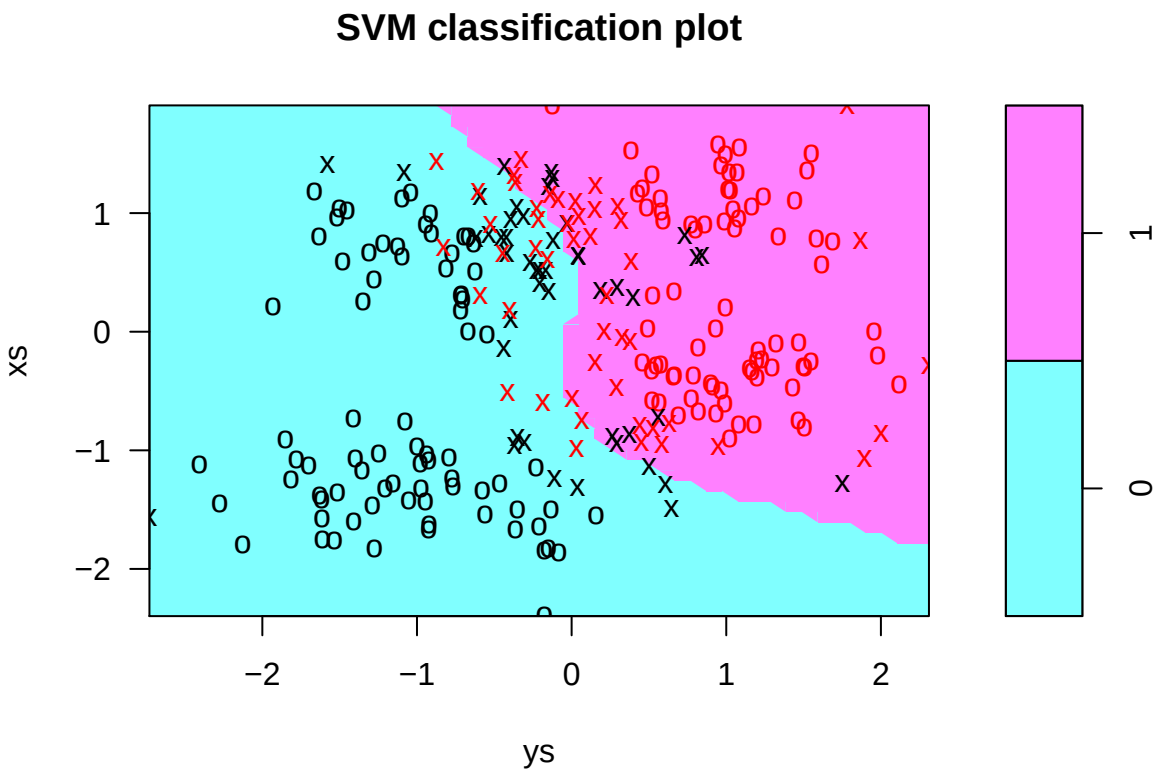
So the minimizer is $f^*(x) = \frac{1}{2} log \frac{Pr(Y=1|X=x)}{1-Pr(Y=1|X=x)} = \frac{1}{2} log \frac{Pr(Y=1|X=x)}{Pr(Y=-1|X=x)}$.

2. Get the "Ripleydataset" (synth.tr) from the website http://www.stats.ox.ac.uk/pub/PRNN/. The dataset contains two predictors and a binary outcome.

(a) Construct a linear support vector classifier.



(b) Construct a support vector classifier with Radial kernel.



(c) Construct a classifier using AdaBoost algorithm (with 50 boosting iterations) with decision stumps as weak learners.

Select the tuning parameter involved in SVM models appropriately. For each method, compute the test error and its standard error on the test set (synth.te). Provide a simple graphical visualization of the produced classification models (i.e. something similar to Figure 2.2 in the textbook [ESL]) and discuss your results.

Since this is a classification problem, we use misclassification rate as the test error.

	svm.linear	svm.raidal	adaboost
misclassification rate	0.1010000	0.09300	0.1280000
standard error	0.3179624	0.30511	0.3547137

The result shows that for the `synth` data, support vector classifier with Radial kernel has the best performance, while adaboost has the worst.

Appendix

```
knitr::opts_chunk$set(echo = FALSE, message = FALSE, warning = FALSE, comment = "")
library(tidyverse)
```

```r
library(e1071)  # for SVM
library(JOUSBoost)  # for adaboost
library(ada)
# load train and test data
train <- read.table("synth.tr", header = T) %>%
  as.tibble() %>%
  mutate(yc = as.factor(yc),
         xs = scale(xs),
         ys = scale(ys))

test <- read.table("synth.te", header = T) %>%
  as.tibble() %>%
  mutate(yc = as.factor(yc),
         xs = scale(xs),
         ys = scale(ys))

# linear SVM
svm.linear = svm(yc ~ ., data = train, type = 'C-classification', kernel = 'linear')
pred.svm.linear = predict(svm.linear, test)

err.svm.linear = sum((as.numeric(pred.svm.linear) - as.numeric(test$yc)) != 0) / length(test$yc)
stderr.svm.linear = sqrt(var(as.numeric(pred.svm.linear) - as.numeric(test$yc)))
plot(svm.linear, train)

# radial SVM
svm.radial = svm(yc ~ ., data = train, type = 'C-classification', kernel = 'radial')
pred.svm.radial = predict(svm.radial, test)

err.svm.radial = sum((as.numeric(pred.svm.radial) - as.numeric(test$yc)) != 0) / length(test$yc)
stderr.svm.radial = sqrt(var(as.numeric(pred.svm.radial) - as.numeric(test$yc)))
plot(svm.radial, train)

# adaboost
train.ada = train %>%
  mutate(yc = ifelse(yc == 1, 1, -1)) %>%
  as.matrix()
test.ada = test %>%
  mutate(yc = ifelse(yc == 1, 1, -1)) %>%
  as.matrix()

ada = adaboost(train.ada[,c(1,2)], train.ada[,3], tree_depth = 1, n_rounds = 50)
pred.ada = predict(ada, test.ada[,c(1,2)])

err.ada = sum((as.numeric(pred.ada) - as.numeric(test.ada[,3])) != 0) / length(test.ada[,3])
stderr.ada = sqrt(var((as.numeric(pred.ada) - as.numeric(test.ada[,3]))/2))

# output test error and standard error
err.output <- tibble(
  svm.linear = c(err.svm.linear, stderr.svm.linear),
  svm.raidal = c(err.svm.radial, stderr.svm.radial),
  adaboost = c(err.ada, stderr.ada)
)
rownames(err.output) = c("misclassification rate", "standard error")
```

```
err.output %>% knitr::kable()
```