Statistics 452: Statistical Learning and Prediction

Chapter 9, Part 2: More on Support Vector Machines

Brad McNeney

2018-11-07

SVMs with More than Two Classes

- No natural extension of the separating hyperplanes idea to multiple classes.
- Instead, use one-versus-one or one-versus-all classification.

One-versus-One Classification

- If there are K classes, there are $K(K-1)/2$ pairwise comparisons.
- Can train $K(K-1)/2$ SVMs.
- For a test observation, get $K(K-1)/2$ predictions.
  - Each prediction is like a game in a tournament.
  - The final classification for the test observation is the winner of the tournament; i.e., the class that the test observation was most frequently assigned to.

One-verus-All Classification

- For each class, train a SVM to classify that class versus all others pooled together.
  - End up with K classifiers.
- The classification of a test observation is class with highest confidence; i.e., on the right side of the decision boundary, and farthest from the boundary.

Example: Gene Expression Data

- Four tumor types ($K = 4$) measured on 63 training and 20 test tumors.
- For each tumor, there are 2308 measurements of "gene expression".
  - With this many features relative to the number of observations, non-linear kernels may provide **too much** flexibility – use linear.
- Classify tumor type based on expression measurements.
- `svm()` uses one-vs-one classification

```
library(ISLR)
data(Khan)
# help(Khan)
dat <- data.frame(y=as.factor(Khan$ytrain),
                  Khan$xtrain)
head(names(dat))
```

```
## [1] "y"  "X1" "X2" "X3" "X4" "X5"
```

```r
library(e1071)
fit <- svm(y~., data=dat,kernel="linear",cost=10)
pp <- predict(fit,newdata=data.frame(Khan$xtest))
table(pp,Khan$ytest)
```

```
##
## pp   1 2 3 4
##    1 3 0 0 0
##    2 0 6 2 0
##    3 0 0 4 0
##    4 0 0 0 5
```

```r
2/20
```

```
## [1] 0.1
```

The SVM and Logistic Regression

Loss + Penalty Formulation of the Support Vector Classifier

- Let $\mathbf{X}$ and $\mathbf{y}$ denote the matrix of X's and vector of y's, respectively and

  $$f(X;\beta) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

  be the decision boundary.
- One can show that the criterion function that the support vector classifier minimizes to estimate f is of the form

  $$\min_{\beta}\{L(\mathbf{X},\mathbf{y},\beta) + \lambda P(\beta)\}$$

  where $L()$ is the so-called hinge loss function

  $$L(\mathbf{X},\mathbf{y},\beta) = \sum_{i=1}^{n} \max[0, 1 - y_i f(x_i;\beta)],$$

  $P(\beta) = \sum_{j=1}^{p} \beta_j^2$ is the ℓ_2 penalty function and λ is a tuning parameter.

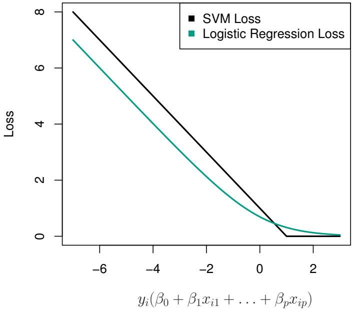
Hinge *versus* Logistic Loss

- The hinge loss function is similar to the logistic loss function

$$\sum_{i=1}^{n} \log(1 + e^{-y_i f(x_i;\beta)}),$$

used in logistic regression.

  - Recall that logistic regression is fit by maximum likelihood.
  - ML amounts to minimizing a negative-log-likelihood loss.
  - Loss, as written above, is for outcomes coded as $-1/1$.



Support Vector Classifier/Machine *versus* Logistic Regression

- Conclude that SV Classifier is similar to logistic regression penalized with an ℓ_2 penalty.
- Can further argue that the SV Machine is similar to ℓ_2-penalized logistic regression with non-linear functions of the predictors.