Summary on Support Vector Machines with Applications

Liu Huihang

2020.1.6

Support Vector Classifier

Assume our training data consists of N pairs $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ with $x_i \in \mathbb{R}^p$ and $y_i \in \{-1, 1\}$. Define a hyperplane by

$$\{x : f(x) = x^T\beta + \beta_0 = 0\} \tag{1}$$

where β is a unit vector: $\|\beta\| = 1$. A classification rule induced by $f(x)$ is

$$G(x) = \text{sign}\left[x^T\beta + \beta_0\right] \tag{2}$$

We know that $f(x)$ in (1) gives the signed distance from a point x to the hyperplane $f(x) = x^T\beta + \beta_0 = 0$. Since the classes are separable, we can find a function $f(x) = x^T\beta + \beta_0 = 0$ with $y_i f(x_i) > 0$, $\forall i$. Hence we are able to find the hyperplane that creates the biggest margin between the training points for class 1 and -1. The optimization problem

$$\begin{aligned} &\max_{\beta, \beta_0, \|\beta\|=1} M \\ &\text{subject to } y_i\left(x_i^T\beta + \beta_0\right) \geq M, i = 1, \dots, N \end{aligned} \tag{3}$$

captures this concept. The band in the figure is M units away from the hyperplane on either side, and hence $2M$ units wide. It is called the margin.

Computations can be done by solving a quadratic programming problem with Lagrange multipliers method. please see *The Element of Statistical Learning* for detials.

The support vector classifier finds linear boundaries in the input feature space. As with other linear methods, we can make the procedure more flexible by enlarging the feature space using basis expansions such as polynomials or splines. Generally linear boundaries in the enlarged space achieve better training-class separation, and translate to nonlinear boundaries in the original space. Once the basis functions $h_m(x)$, $m = 1, \dots, M$ are selected, the procedure is the same as before. We fit the SV classifier using input features $h(x_i) = (h_1(x_i), h_2(x_i), \dots, h_M(x_i)), i = 1, \dots, N$, and produce the (nonlinear) function $\hat{f}(x) = h(x)^T\hat{\beta} + \hat{\beta}_0$. The classifier is $\hat{G}(x) = sign(\hat{f}(x))$ as before.

The support vector machine classifier is an extension of this idea, where the dimension of the enlarged space is allowed to get very large, infinite in some cases. It might seem that the computations would become prohibitive. It would also seem that with sufficient basis functions, the data would be separable, and overfitting would occur. We first explore how the SVM technology deals with these issues. We then see that in fact the SVM classifier is solving a function-fitting problem using a particular criterion and form of regularization, and is part of a much bigger class of problems that includes the smoothing splines.

Real data: cats

The `cats` data set in the `MASS` package contains three variables, where the variable *Sex* indicates the gender of the cat ('F'or'M'), *Bwt* indicates the cat's weight (unit: kg), and *Hwt* indicates the cat's heart weight (unit: g). Our goal is to use the SVM algorithm to classify cats through two variables, *Bwt* and *Hwt*.

```r
# load package and data
library("MASS")
head(cats)
```

```
##   Sex Bwt Hwt
## 1   F 2.0 7.0
## 2   F 2.0 7.4
## 3   F 2.0 9.5
## 4   F 2.1 7.2
## 5   F 2.1 7.3
## 6   F 2.1 7.6
```

```r
# Select training samples (70%) and test samples (30%)
index <- sample(2,nrow(cats),replace = TRUE,prob=c(0.7,0.3))
traindata <- cats[index==1,]
testdata <- cats[index==2,]

# model
cats_svm_model <- svm(Sex~.,data=traindata)
print(cats_svm_model)
```

```
##
## Call:
## svm(formula = Sex ~ ., data = traindata)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  61
```

```r
# check result on training data
cats_svm_model_pred_1 <- predict(cats_svm_model,traindata[,-1])
cats_table_1 <- table(pred=cats_svm_model_pred_1,true=traindata[,1])
print(cats_table_1)
```

```
##     true
## pred  F  M
##    F 23 11
##    M  9 54
```

```r
accuracy_1 <- sum(diag(cats_table_1))/sum(cats_table_1)
print(accuracy_1)
```

```
## [1] 0.7938144
```

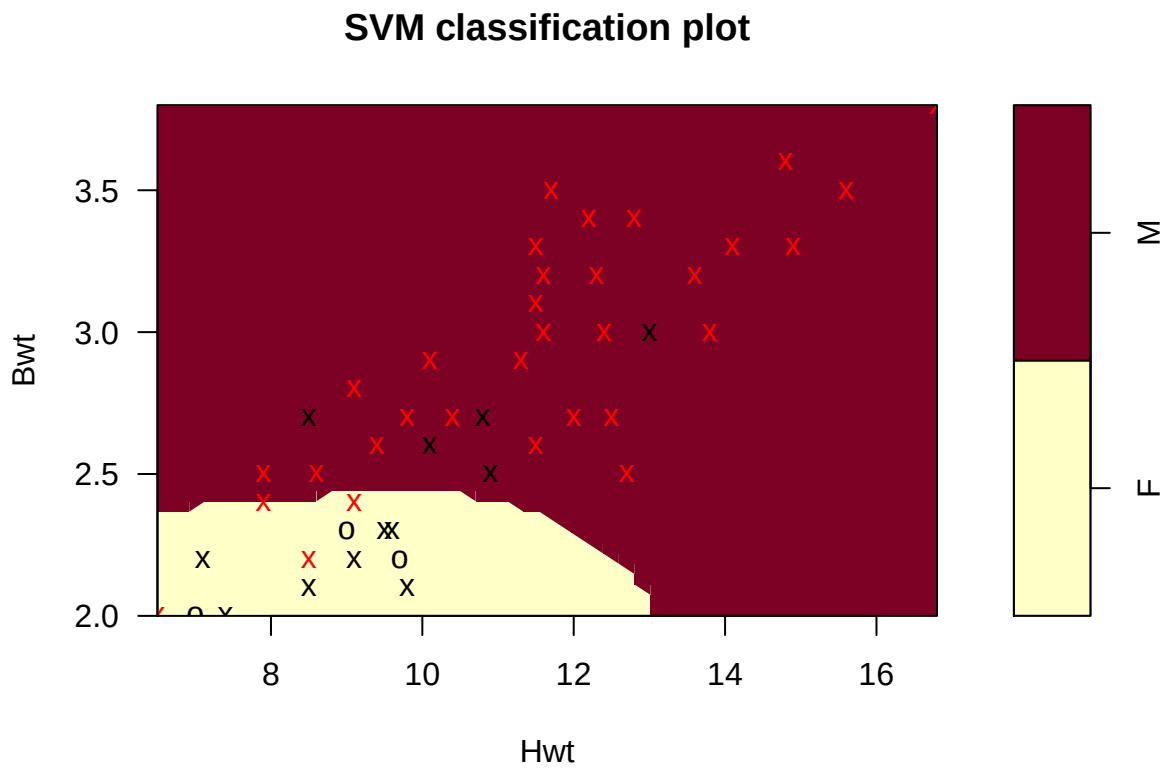
```r
# check on the test data
cats_svm_model_pred_2 <- predict(cats_svm_model,testdata[,-1])
cats_table_2 <- table(pred=cats_svm_model_pred_2,true=testdata[,1])
cats_table_2
```

```
##     true
## pred  F  M
##    F 10  3
##    M  5 29
```

```r
accuracy_2 <- sum(diag(cats_table_2))/sum(cats_table_2)
accuracy_2
```

```
## [1] 0.8297872
```

```r
# plot the result
plot(cats_svm_model,testdata)
```



Real data: iris

First of all, for the classification problem, the 'type' parameter in the `svm()` han function has three options: *C-classification*, *nu-classification*, and *one-classification*. And the 'kernel' parameter has four options: *linear*, *polynomial*, *radial* and *sigmoid*.

In order to comprehensively compare the model effects of these *12* combinations, build a self-written function

```r
svm_test <- function(x,y){
  type <- c('C-classification','nu-classification','one-classification')
  kernel <- c('linear','polynomial','radial','sigmoid')
  pred <- array(0, dim=c(nrow(x),3,4))
  errors <- matrix(0,3,4)
  dimnames(errors) <- list(type, kernel)
  for(i in 1:3){
    for(j in 1:4){
      pred[,i,j] <- predict(object = svm(x, y, type = type[i], kernel = kernel[j]), newdata = x)
      if(i > 2) errors[i,j] <- sum(pred[,i,j] != 1)
      else errors[i,j] <- sum(pred[,i,j] != as.integer(y))
```

```
      }
    }
  return(errors)
}

iris_x <- iris[,1:4]
iris_y <- iris[,5]
svm_test(x=iris_x,y=iris_y)
```

```
##                     linear polynomial radial sigmoid
## C-classification         5          7      4      17
## nu-classification        5         14      5      12
## one-classification     102         75     76      75
```

According to the above results, type = 'C-classification' and kernel = 'radial' return the smallest amount of false positives, so this combination can be considered to classify IRIS.

```
iris_model <- svm(x=iris[,1:4],y=iris[,5],type = 'C-classification', kernel = 'radial')
iris_pred <- predict(object = iris_model, newdata = iris[,1:4])
iris_Freq <- table(iris[,5], iris_pred)
print(iris_Freq)
```

```
##             iris_pred
##              setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         2
##   virginica       0          2        48
```

In this way, we saw that the correct classification rate reached 97.3%, and 4 samples were misjudged. In order to further improve the classification effect, we can use the `tune.svm()` function to find the optimal parameters of the `svm()` function.

Summary on A Selective Overview of Variable Selection in High Dimensional Feature Space

Liu Huihang

2020.1.6

Review of model selection method

High dimensional statistical problems arise from diverse fields of scientific research and technological development. Variable selection plays a pivotal role in contemporary statistical learning and scientific discoveries.

Consider the linear regression model

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon \tag{1}$$

The first method we learn is Lasso Tibshirani [1996],

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - x_i^T\beta\right)^2 + \|\beta\|_1 \tag{2}$$

It is easy to understand and the speed of computation is faster than solving a $\|\cdot\|_0$ penalty function which is named as best sebset selection method

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - x_i^T\beta\right)^2 + \|\beta\|_0 \tag{3}$$

But this method takes a very long time to solve at that time. Thirty years later, we have the latest methods to solve this problem in an acceptable time Wen et al. [2017].

Later many a methods were developed to imporve the result of selection. They used the same framework to constrain the problem as following

$$\min_{\beta\in\mathbf{R}^p}\left\{\frac{1}{2n}\|\mathbf{y}-\mathbf{X}\beta\|^2 + \sum_{j=1}^{p} p_\lambda\left(|\beta_j|\right)\right\} \tag{4}$$

For example, SCAD introduced by Fan and Li [2001]

$$p'_\lambda(t) = \lambda\left\{I(t\le\lambda) + \frac{(a\lambda - t)_+}{(a-1)\lambda}I(t>\lambda)\right\} \quad \text{for some } a>2 \tag{5}$$

where $p_\lambda(0)=0$ and, often, $a=3.7$ is used (suggested by a Bayesian argument).

A penalty of similar spirit is the minimax concave penalty (MCP) Zhang and Li [2011], whose derivative is given by

$$p'_\lambda(t) = \frac{(a\lambda - t)_+}{a} \tag{6}$$

A family of concave penalties that bridge the L_0 and L_1 penalties was studied by Lv et al. [2009] And Zheng et al. [2014] focus on the hard thresholding penalty

$$p_{\mathrm{H},\lambda}(t) = \frac{1}{2}\left\{\lambda^2 - (\lambda - t)_+^2\right\}, \quad t\geqslant 0 \tag{7}$$

Generally speaking, two classes of penalty functions have been proposed in the literature: convex ones and concave ones. The L_1 penalty tends to yield a larger model than the true one for optimizing predictions, and many of the selected variables may be insignificant, showing that the resulting method may not be ideal for variable selection. The relatively large model size also reduces the interpretability of the selected model. When concave penalties is used, it is generally difficult to study the properties of the global optimizer for concave regularization methods. Fan and Lv [2013] characterize theoretically the global optimizer of the regularization method with the combined L1 and concave penalty, in the setting of the high-dimensional linear model by studying the resulting regularization problem

$$\min_{\beta \in \mathbf{R}^p} \left\{ \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_0 \|\beta\|_1 + \|p_\lambda(\beta)\|_1 \right\} \tag{8}$$

where $\lambda_0 = c\{(logp)/n\}^{1/2}$ for some positive constant c.

Since then, the classical high dimensonal selection problems have been studied well and not many scholars have been studying general high-dimensional problems in the world.

Scholars are paying more attention to some faster calculation methods and deeper statistical problems such as statistical inference.

References

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

Yingying Fan and Jinchi Lv. Asymptotic properties for combined L 1 and concave regularization. *Biometrika*, 101(1):57–70, 2013.

Jinchi Lv, Yingying Fan, and Others. A unified approach to model selection and sparse recovery using regularized least squares. *The Annals of Statistics*, 37(6A):3498–3528, 2009.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Canhong Wen, Aijun Zhang, Shijie Quan, and Xueqin Wang. BeSS: An R Package for Best Subset Selection in Linear, Logistic and CoxPH Models. *arXiv preprint arXiv:1709.06254*, 2017.

Yiyun Zhang and Runze Li. Iterative conditional maximization algorithm for nonconcave penalized likelihood. In *Nonparametric Statistics and Mixture Models: A Festschrift in Honor of Thomas P Hettmansperger*, pages 336–351. World Scientific, 2011.

Zemin Zheng, Yingying Fan, and Jinchi Lv. High dimensional thresholded regression and shrinkage effect. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(3):627–649, 2014.