SVM-flexible discriminant analysis

Huimin Peng

November 20, 2014

Outline

SVM

Nonlinear

SVM = Penalization method

discriminant analysis

FDA: flexible discriminant analysis

penalized discriminant analysis

mixture discriminant analysis

SVM

Define a hyperplane that separates the observations:

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

The optimization problem is

$$\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{subject to } \xi_i \geq 0, y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i \ \forall i,$$

where $\sum \xi_i \leq C$ and ξ_i is the proportion of wrong predictions. C is the tunning parameter. Large C will reduce positive ξ_i and lead to a wiggly boundary. Small C will lead to a smoother boundary.

Reference: Elements of statistical learning chapter 12

The solution is

$$\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i.$$

The decision function is

$$\hat{G}(x) = \mathsf{sign}[\hat{f}(x)] = \mathsf{sign}[x^T \hat{\beta} + \hat{\beta}_0].$$

Nonlinear

Input features: (transformed feature vectors)

$$h(x_i) = (h_1(x_i), h_2(x_i), \cdots, h_M(x_i)).$$

Similarly, the classifier:

$$\hat{G}(x) = \mathsf{sign}(\hat{f}(x)) = \mathsf{sign}\left(h(x)^T \hat{\beta} + \hat{\beta}_0\right).$$

The solution function is

$$\hat{f}(x) = \sum_{i=1}^{N} \hat{\alpha}_i y_i K(x, x_i) + \hat{\beta}_0.$$

SVM = Penalization method

If $f(x) = h(x)^T\beta + \beta_0$, then

$$\min_{\beta_0,\beta} \sum_{i=1}^{N} [1 - y_i f(x_i)]_+ + \frac{\lambda}{2}\|\beta\|^2. \tag{1}$$

This is the loss + penalty function. It provides the same solution as

$$\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{subject to } \xi_i \geq 0, y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i \ \forall i,$$

discriminant analysis

LDA: linear discriminant analysis QDA: quadratic discriminant analysis FDA: flexible discriminant analysis PDA: penalized discriminant analysis MDA: mixture discriminant analysis R package: mda

classes:

$$\mathcal{G} = \{1, 2, \cdots, K\}.$$

K classes. score function:

$$\theta : \mathcal{G} \mapsto \mathbf{R}^1.$$

Reference: Elements of statistical learning chapter 12

assign scores to the classes. training data: $(g_i, x_i), i = 1, 2, \cdots, N$. optimization:

$$\min_{\beta,\theta} \sum_{i=1}^{N} (\theta(g_i) - x_i^T \beta)^2.$$

FDA: flexible discriminant analysis

More generally, build $L \leq K-1$ sets of independent scorings for the class labels, $\theta_1, \theta_2, \cdots, \theta_L$, and L corresponds to the linear maps

$$\eta_l(X) = X^T \beta_l, \ l = 1, 2, \cdots, L.$$

We can generalize $\eta_l(x) = x^T \beta_l$ to be more flexible, nonparametric fits and add a J as a regularizer appropriate for some forms of nonparametric regression.

$$ASR(\{\theta_l, \eta_l\}_{l=1}^{L}) = \frac{1}{N} \sum_{l=1}^{L} \left[\sum_{i=1}^{N} (\theta_l(g_i) - \eta_l(x_i))^2 + \lambda J(\eta_l) \right].$$

fda(formula, data, weights, theta, dimension, eps, method, keep.fitted, ...)

Reference: Elements of statistical learning chapter 12

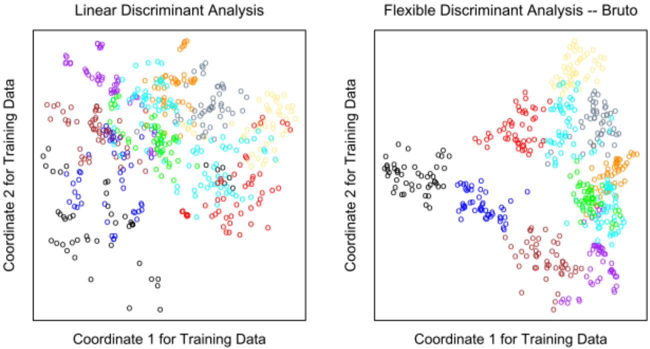

FIGURE 12.10. *The left plot shows the first two LDA canonical variates for the vowel training data. The right plot shows the corresponding projection when FDA/BRUTO is used to fit the model; plotted are the fitted regression functions $\hat{\eta}_1(x_i)$ and $\hat{\eta}_2(x_i)$. Notice the improved separation. The colors represent the eleven different vowel sounds.*

Reference: Elements of statistical learning chapter 12

```
data(iris)
irisfit <- fda(Species ~ ., data = iris)
confusion(irisfit, iris)
confusion(predict(irisfit, iris), iris$Species)
plot(irisfit)
coef(irisfit)
posteriors <- predict(irisfit, type = "post")
confusion(softmax(posteriors), iris[, "Species"])
marsfit <- fda(Species ~ ., data = iris, method = mars)
marsfit2 <- update(marsfit, degree = 2)
#include interactions up to 2nd degree
marsfit3 <- update(marsfit, theta = marsfit$means[, 1:2])
#start from the fitted coef's in marsfit
```

Reference: Elements of statistical learning chapter 12

```
> coef(irisfit)
                   [,1]        [,2]
Intercept    -2.1264786 -6.72910343
Sepal.Length -0.8377979  0.02434685
Sepal.Width  -1.5500519  2.18649663
Petal.Length  2.2235596 -0.94138258
Petal.Width   2.8389936  2.86801283
```

Reference: Elements of statistical learning chapter 12

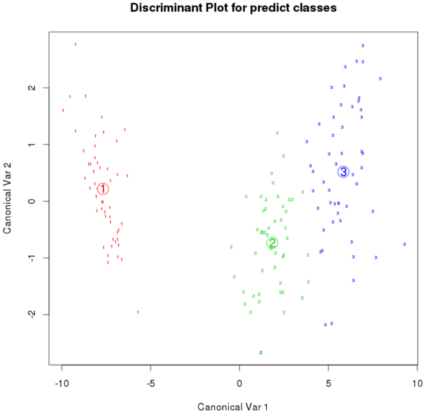

Figure 1: plot(irisfit)

Reference: Elements of statistical learning chapter 12

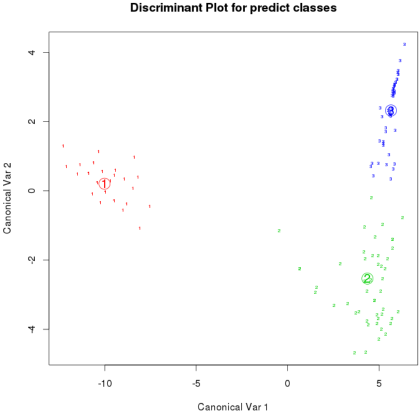

Figure 2: plot(marsfit)

Reference: Elements of statistical learning chapter 12

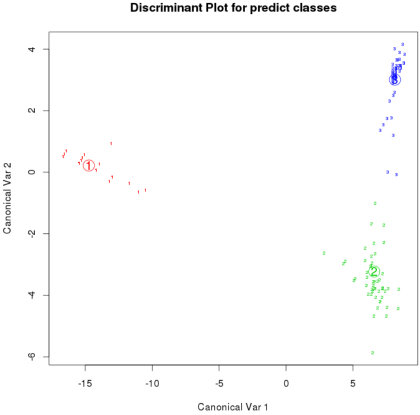

Figure 3: plot(marsfit1)

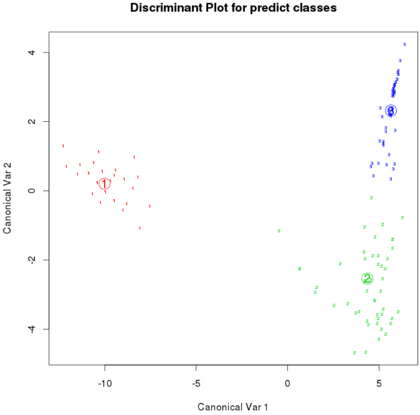

Figure 4: plot(marsfit2)

penalized discriminant analysis

Quadratic penalty on the coefficients:

$$ASR(\{\theta_l, \beta_l\}_{l=1}^{L}) = \frac{1}{N} \sum_{l=1}^{L} \left[\sum_{i=1}^{N} (\theta_l(g_i) - h^T(x_i)\beta_l)^2 + \lambda \beta_l^T \Omega \beta_l \right].$$

The choice of Ω depends on the problem setting. $\eta_l(x) = h(x)^T \beta_l$.

gen.ridge(x, y, weights, lambda=1, omega, df, ...)

mixture discriminant analysis

A Gaussian mixture model for the kth class has the density

$$P(X|G=k)=\sum_{r=1}^{R_k}\pi_{kr}\phi(X;\mu_{kr},\Sigma), \tag{2}$$

where $\sum_{r=1}^{R_k}\pi_{kr}=1$, R_k is the number of points in class k. Incorporating the class prior probabilities Π_k:

$$P(G=k|X=x)=\frac{\sum_{r=1}^{R_k}\pi_{kr}\phi(X;\mu_{kr},\Sigma)\Pi_k}{\sum_{l=1}^{K}\sum_{r=1}^{R_l}\pi_{lr}\phi(X;\mu_{lr},\Sigma)\Pi_l}.$$

mda(formula, data, subclasses, sub.df, tot.df, dimension, eps, iter, weights, method, keep.fitted, trace, ...)

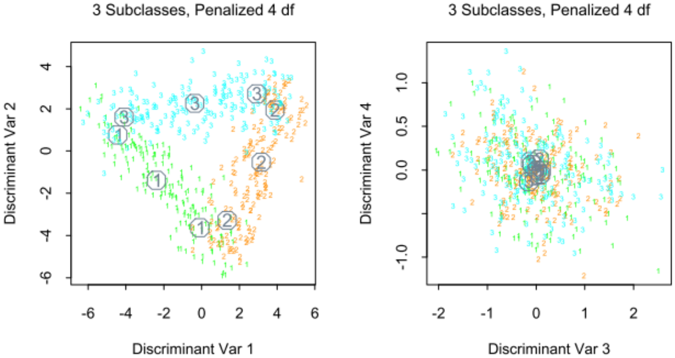

FIGURE 12.15. *Some two-dimensional views of the MDA model fitted to a sample of the waveform model. The points are independent test data, projected on to the leading two canonical coordinates (left panel), and the third and fourth (right panel). The subclass centers are indicated.*

Reference: Elements of statistical learning chapter 12

```
data(iris)
irisfit <- mda(Species ~ ., data = iris)
mfit=mda(Species~.,data=iris,subclass=2)
coef(mfit)
```

```
> coef(mfit)
                   [,1]        [,2]      [,3]      [,4]
Intercept     6.8563935 -15.1565801 -1.454555 -2.535648
Sepal.Length  0.5545477   1.3506122  1.016966  2.945456
Sepal.Width   1.5867703   2.4658435 -1.345301 -2.562105
Petal.Length -3.2435199   0.3621319  1.341652 -2.921295
Petal.Width  -2.3003933  -1.3635028 -4.516518  3.448416
```

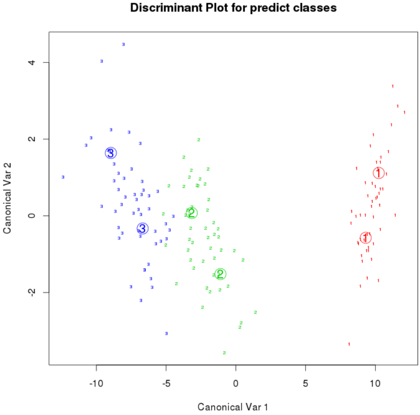

Figure 5: plot(mfit)

Questions?