Support Vector Machines and Flexible Discriminants

Thomas Lonon

Division of Financial Engineering
Stevens Institute of Technology

April 17, 2017

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

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

where β is a unit vector.

A classification rule induced by $f(x)$ is

$$G(x) = \text{sign}(x^T \beta + \beta_0)$$

Maximum Margin Optimization

$$\max_{\beta,\beta_0,\|\beta\|=1} M$$

$$\text{subject to } y_i(x_i^T\beta + \beta_0) \geq M, i = 1, \ldots, N$$

This can be more conveniently rephrased as

$$\min_{\beta,\beta_0} \|\beta\|$$

$$\text{subject to } y_i(x_i^T\beta + \beta_0) \geq 1, i = 1, \ldots, N$$

Note that $M = 1/\|\beta\|$

Dealing with Overlap

Define the slack variables $\xi = (\xi_1, \xi_2, \ldots, \xi_N)$. We then modify our previous constraints in one of two ways:

$$y_i(x_i^T \beta + \beta_0) \geq M - \xi_i$$

or

$$y_i(x_i^T \beta + \beta_0) \geq M(1 - \xi_i)$$

$\forall i, \xi_i \geq 0, \sum_{i=1}^{N} \xi_i \leq$ constant

Define $M = 1/\|\beta\|$. With these overlapping datasets, our optimization for the margins becomes:

$$\min \|\beta\|$$

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

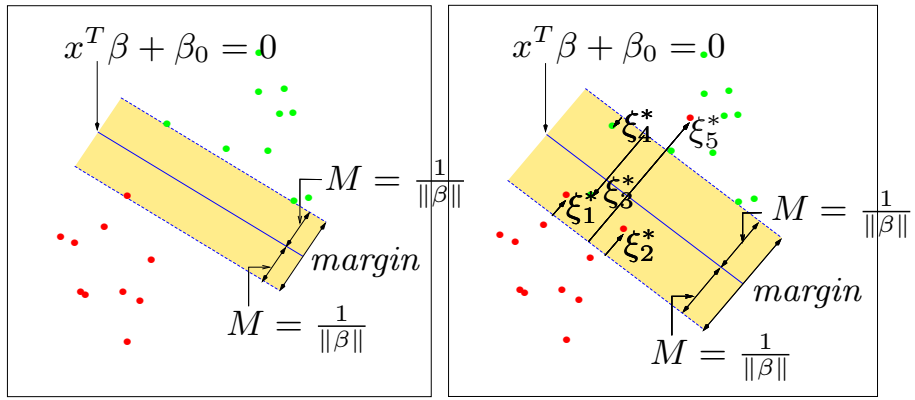

FIGURE 12.1. *Support vector classifiers. The left panel shows the separable case. The decision boundary is the solid line, while broken lines bound the shaded maximal margin of width* $2M = 2/\|\beta\|$. *The right panel shows the nonseparable (overlap) case. The points labeled* ξ_j^* *are on the wrong side of their margin by an amount* $\xi_j^* = M\xi_j$; *points on the correct side have* $\xi_j^* = 0$. *The margin is maximized subject to a total budget* $\sum \xi_i \leq$ *constant. Hence* $\sum \xi_j^*$ *is the total distance of points on the wrong side of their margin.*

We express out optimization problem (to take advantage of Lagrange multipliers) as:

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

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

where the "cost" parameter C replaces the constant from earlier.

The Lagrange (primal) function is:

$$L_P = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i[y_i(x_i^T\beta + \beta_0) - (1 - \xi_i)] - \sum_{i=1}^{N} \mu_i\xi_i$$

We minimize this primal function with respect to β, β_0, and ξ_i. Setting the respective derivatives to 0 gives us:

$$\beta = \sum_{i=1}^{N} \alpha_i y_i x_i$$

$$0 = \sum_{i=1}^{N} \alpha_i y_i$$

$$\alpha_i = C - \mu_i, \forall i$$

including positivity constraints $\alpha_i, \mu_i, \xi_i \geq 0 \forall i$

Plugging the derivatives into the original optimization problem allows us to obtain the Lagrangian dual objective function

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \alpha_i \alpha_{i'} y_i y_{i'} x_i^T x_{i'}$$

We maximize L_D subject to $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$. We also include the constraints:

$$\begin{aligned} \alpha_i [y_i (x_i^T \beta + \beta_0) - (1 - \xi_i)] &= 0 \\ \mu_i \xi_i &= 0 \\ y_i (x_i^T \beta + \beta_0) - (1 - \xi_i) &\geq 0 \end{aligned}$$

If we phrase the problem so that it only involves the inner products, the Lagrange dual function is:

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \alpha_i \alpha_{i'} y_i y_{i'} \langle h(x_i), h(x_{i'}) \rangle$$

and the solution function $f(x)$ can be written

$$\begin{aligned} f(x) &= h(x)^T \beta + \beta_0 \\ &= \sum_{i=1}^{N} \alpha_i y_i \langle h(x), h(x_i) \rangle + \beta_0 \end{aligned}$$

[1]

These are expressed through inner products of the transformation functions $h(x)$. We actually don't require that the transformation is specified, but rather only knowledge of the kernel:

$$K(x, x') = \langle h(x), h(x') \rangle$$

Popular choices of K in SVM are

- **d^{th}-Degree Polynomial:** $K(x, x') = (1 + \langle x, x' \rangle)^d$
- **Radial basis:** $K(x, x') = e^{-\gamma \|x - x'\|^2}$
- **Neural Network:** $K(x, x') = \tanh(\kappa_1 \langle x, x' \rangle + \kappa_2)$

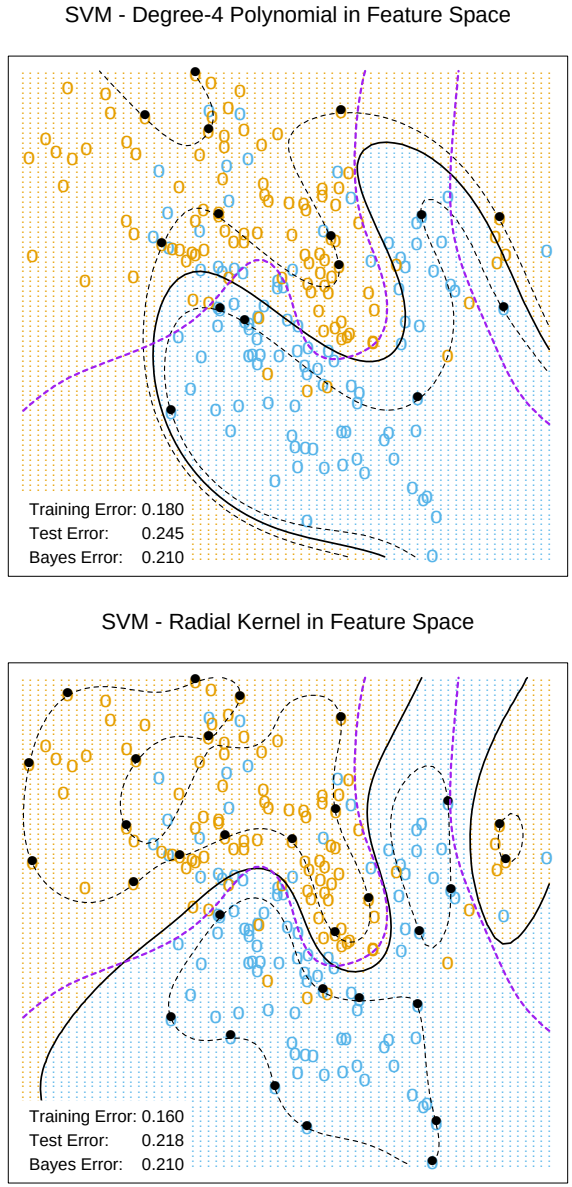

FIGURE 12.3. *Two nonlinear SVMs for the mixture data. The upper plot uses a 4th degree polynomial*

The solution to these equations can be written as:

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

With $f(x) = h(x)^T\beta + \beta_0$, we want to consider the optimization problem

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

This has the form *loss + penalty*

This is utilizing the "hinge" loss function, $L(y, f) = [1 - yf]_+$, which is reasonable for two-class classification

Loss Functions

Loss Function	$L[y, f(x)]$	Minimizing Function
Binomial Deviance	$\log[1 + e^{-yf(x)}]$	$f(x) = \log \frac{\mathbb{P}(Y=+1\|x)}{\mathbb{P}(Y=-1\|x)}$
SVM Hinge Loss	$[1 - yf(x)]_+$	$f(x)$ $= \text{sign}[\mathbb{P}(Y = +1\|x) - \frac{1}{2}]$
Squared Error	$[y - f(x)]^2$ $= [1 - yf(x)]^2$	$f(x) = 2\mathbb{P}(Y = +1\|x) - 1$
"Huberised" Square Hinge Loss	$-4yf(x), yf(x) < -1$ $[1 - yf(x)]_+^2, \text{ow}$	$f(x) = 2\mathbb{P}(Y = +1\|x) - 1$

[1]

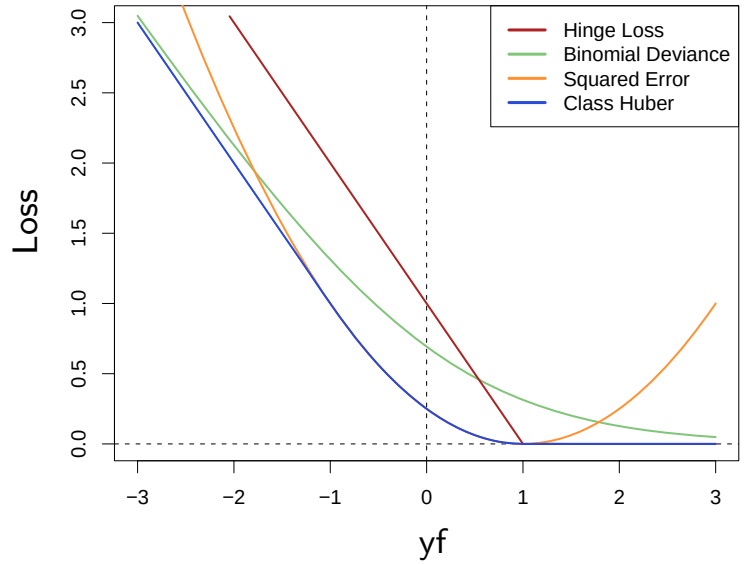

FIGURE 12.4. *The support vector loss function (hinge loss), compared to the negative log-likelihood loss (binomial deviance) for logistic regression, squared-error loss, and a "Huberized" version of the squared hinge loss. All are shown as a function of yf rather than f, because of the symmetry between the $y = +1$ and $y = -1$ case. The deviance and Huber have the same asymptotes as the SVM loss, but are rounded in the interior All are scaled to have the limiting left-tail slope of -1.*

SVM for Regression

The linear regression model has the form:

$$f(x) = x^T\beta + \beta_0$$

Where β is estimated by minimizing

$$H(\beta, \beta_0) = \sum_{i=1}^{N} V(y_i - f(x_i)) + \frac{\lambda}{2}\|\beta\|^2$$

where

$$V_\epsilon(r) = \begin{cases} 0 & \text{if } |r| < \epsilon \\ |r| - \epsilon & \text{otherwise} \end{cases}$$

We can compare this error measure V_ϵ to more robust measures used in statistics, such as the Huber

$$V_H(r) = \begin{cases} r^2/2 & \text{if } |r| \leq c \\ c|r| - c^2/2 & |r| > c \end{cases}$$

This reduces from quadratic to linear the contributions of observations with absolute value greater than c, which makes fitting less sensitive to outliers.

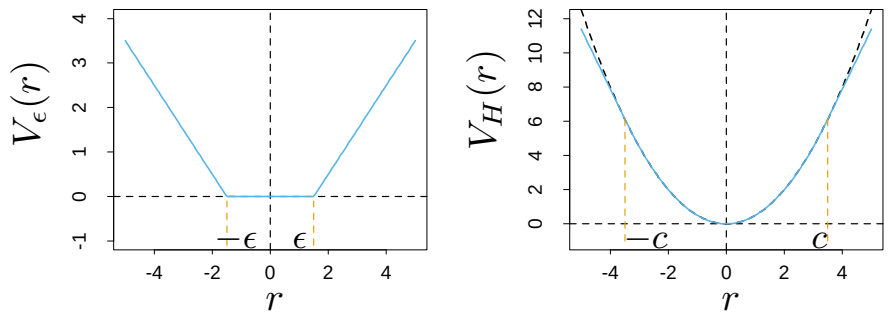

FIGURE 12.8. *The left panel shows the ϵ-insensitive error function used by the support vector regression machine. The right panel shows the error function used in Huber's robust regression (blue curve). Beyond $|c|$, the function changes from quadratic to linear.*

If $\hat{\beta}, \hat{\beta}_0$ are the minimizer of H, we have the solution

$$\hat{\beta} = \sum_{i=1}^{N} (\hat{\alpha}_i^* - \hat{\alpha}_i) x_i$$

$$\hat{f}(x) = \sum_{i=1}^{N} (\hat{\alpha}_i^* - \hat{\alpha}_i) \langle x, x_i \rangle + \beta_0$$

where $\hat{\alpha}_i^*, \hat{\alpha}_i$ are positive and solve the quadratic programming problem

$$\min_{\alpha, \alpha_i^*} \epsilon \sum_{i=1}^{N} (\alpha_i^* + \alpha_i) - \sum_{i=1}^{N} y_i (\alpha_i^* - \alpha_i) + \frac{1}{2} \sum_{i,i'=1}^{N} (\alpha_i^* - \alpha_i)(\alpha_{i'}^* - \alpha_{i'}) \langle x_i, x_{i'} \rangle$$

subject to constraints

$$0 \leq \alpha_i, \alpha_i^* \leq 1/\lambda$$

$$\sum_{i=1}^{N} (\alpha_i^* - \alpha_i) = 0$$

$$\alpha_i \alpha_{i'} = 0$$

Regression and Kernels

For a set of basis functions $\{h_m(x)\}, m = 1, 2, \ldots, M$

$$f(x) = \sum_{m=1}^{M} \beta_m h_m(x) + \beta_0$$

To estimate β and β_0 we minimize

$$H(\beta, \beta_0) = \sum_{i=1}^{N} V(y_i - f(x_i)) + \frac{\lambda}{2} \sum \beta_m^2$$

The solution has the form:

$$\hat{f}(x) = \sum_{i=1}^{N} \hat{a}_i K(x, x_i)$$

with $K(x, y) = \sum_{m=1}^{M} h_m(x) h_m(y)$

Virtues of the LDA

- It is a simple prototype classifier
- LDA is the estimated Bayes classifier if the observations are multivariate Gaussian in each class, with a common covariance matrix.
- The decision boundaries created by LDA are linear
- LDA provides natural low-dimensional views of the data
- Often LDA produces the best classification results.

[1]

Failures of the LDA

- Often linear boundaries do not adequately separate the classes
- A single prototype per class is insufficient
- We may have too many predictors

[1]

There are three possible fixes to these problems, the FDA, PDA, and MDA.

Flexible Discriminant Analysis

If our training sample has the form (g_i, x_i) for $i = 1, \ldots, N$ then we solve

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

More generally, we can fine up to $L \leq K - 1$ sets of independent scorings for the class labels, $\theta_1, \ldots, \theta_L$ and L corresponding linear maps $\eta_\ell(X) = X^T \beta_\ell$. The scores and maps are chosen to minimize the average squared residual (ASR)

$$ASR = \frac{1}{N} \sum_{\ell=1}^{L} \left[\sum_{i=1}^{N} (\theta_\ell(g_i) - x_i^T \beta_\ell)^2 \right]$$

It can be shown that the sequence of discriminant vectors ν_ℓ from Chapter 4 are identical to the sequence β_ℓ up to a constant.

We can replace the linear regression fits $\eta_\ell(x) = x^T\beta_\ell$ by more flexible nonparametric fits. Such as generalized additive fits, spline functions, MARS models, etc.

The regression problems are then defined via

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

where J is an appropriate regularizer.

When the nonparametric regression procedure can be represented as a linear operator, we denote this operation as $\mathbf{S}_\lambda$

Computing the FDA Estimates

1. *Multivariate nonparametric regression:* Let $\mathbf{S}_\lambda$ be the linear operator that fits the chosen model and let $\eta^*(x)$ be the vector of fitted regression functions
2. *Optimal scores:* Compute the eigen-decomposition of $\mathbf{Y}^T\hat{\mathbf{Y}} = \mathbf{Y}^T\mathbf{S}_\lambda\mathbf{Y}$, where the eigenvectors, Θ are normalized: $\Theta^T\mathbf{D}_\pi\Theta = \mathbf{I}$. Here $\mathbf{D}_\pi = \mathbf{Y}^T\mathbf{Y}/N$ is a diagonal matrix of the estimated class prior probabilities
3. *Update* the model from step 1 using the optimal scores $\eta(x) = \Theta^T\eta^*(x)$

PDA

Suppose the regression procedure used in the FDA amounted to a linear regression onto a basis expansion $h(X)$, with a penalty on the coefficients:

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

The choice of Ω depends on the problem. Such as if $\eta_\ell(x) = h(x)\beta_\ell$ is an expansion on spline functions, Ω might constrain η_ℓ to be smooth over $\mathbb{R}^p$

- Enlarge the set of predictors X via a basis expansion $h(X)$
- Use LDA in the enlarged space, where the penalized distance is given by:

$$D(x, \mu) = (h(x) - h(\mu))^T (\Sigma_W + \lambda\Omega)^{-1} (h(x) - h(\mu))$$

  where Σ_W is the within-class covariance matrix of $h(x_i)$
- Decompose the classification subspace using a penalized metric:

$$\max u^T \Sigma_{\text{Bet}} u, \text{subject to } u^T (\Sigma_W + \lambda\Omega) u = 1$$

MDA

A Gaussian mixture model for the k^{th} class has density

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

where the *mixing proportions* π_{kr} sum to one.

The class posterior probabilities are given by:

$$\mathbb{P}(G=k|X=x) = \frac{\sum_{r=1}^{R_k} \pi_{kr}\phi(X;\mu_{kr},\Sigma)\Pi_k}{\sum_{\ell=1}^{K}\sum_{r=1}^{R_\ell} \pi_{\ell r}\phi(X;\mu_{\ell r},\Sigma)\Pi_\ell}$$

where Π_k represents the class prior probabilities.[1]

E-step: Given the current parameters, compute the *responsibility* of sub-class c_{kr} within class k for each of the class-k observations ($g_i = k$):

$$W(c_{kr}|x_i, g_i) = \frac{\pi_{kr}\phi(x_i; \mu_{kr}, \Sigma)}{\sum_{\ell=1}^{R_\ell} \pi_{k\ell}\phi(x_i; \mu_{k\ell}, \Sigma)}$$

M-step: Compute the weighted MLE's for the parameters of each of the component Gaussians within each of the classes, using the weights from the E-step.[1]

- The dimension reduction step in LDA, FDA, or PDA is limited by the number of classes; in particular, for $K = 2$ classes no reduction is possible. MDA substitutes subclasses for classes, and then allows us to look at low-dimensional views of the subspace spanned by these subclass centroids. This subspace will often be an important one for discrimination.
- By using FDA or PDA in the M-step, we can adapt even more to particular situations. For example, we can fit MDA models to digitized analog signals and images, with smoothness constraints built in.[1]

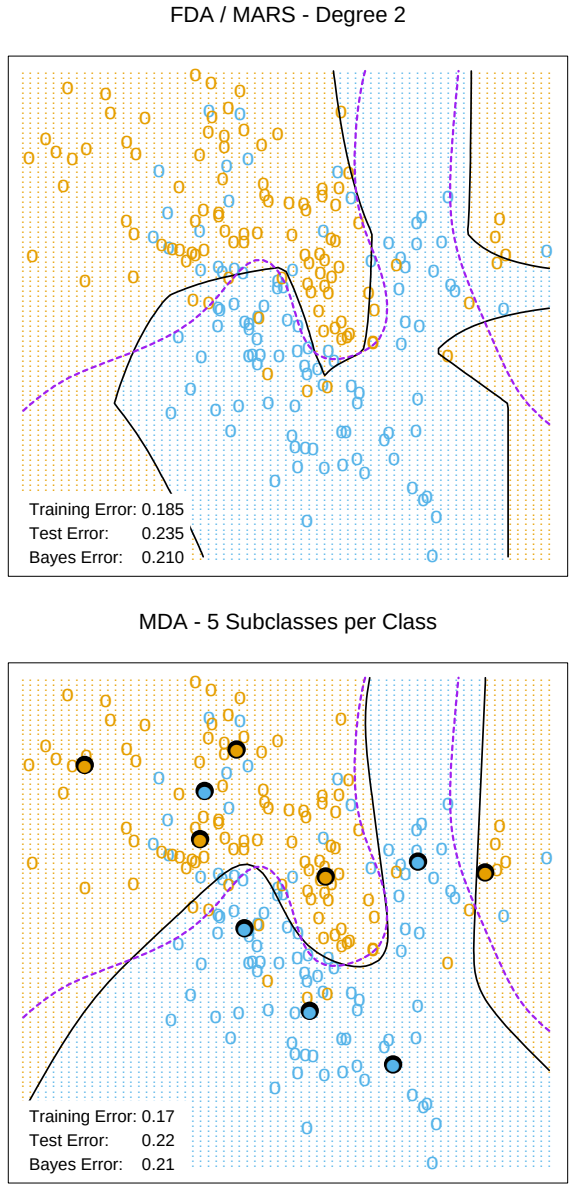

FIGURE 12.13. *FDA and MDA on the mixture data. The upper plot uses FDA with MARS as the regression*

[1] Robert Tibshirani Trevor Hastie and Jerome Friedman. *The Elements of Stastical Learning: Data Mining, Inference, and Prediction*. Number v.2 in Springer Series in Statistics. Springer, 2009.