Statistical Machine Learning

Lecture 11: Support Vector Machines & Flexible Discriminants

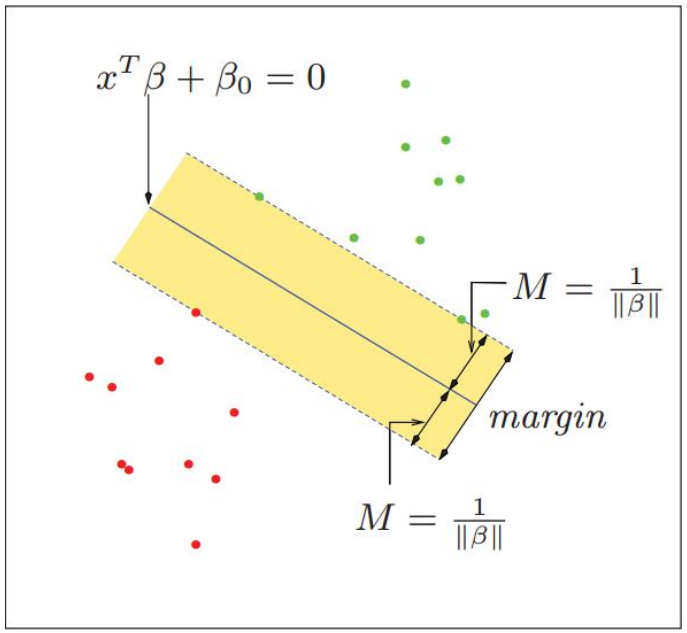
W.Q.Cui Research Group

Department of Statistics and Finance
University of Science and Technology of China

The Support Vector Classifier

Separable Case

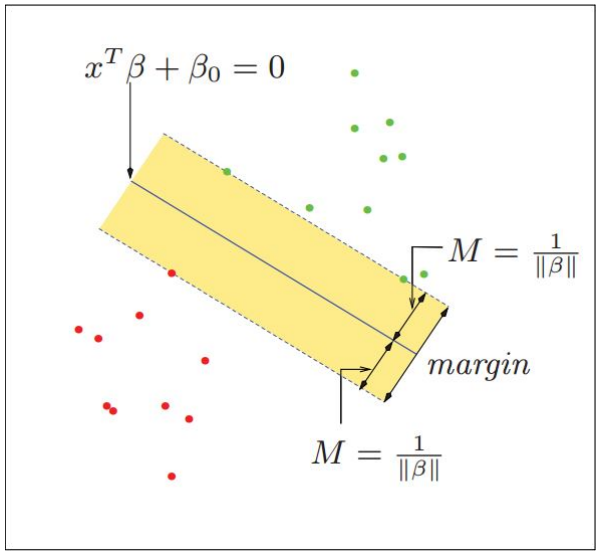
- **Training Data**:$(x_1, y_1), \ldots, (x_n, y_n)$,with $x_i \in \mathbb{R}^p; y_i \in \{-1, 1\}$.
- **Aim**: Find the separating hyperplane with biggest margin between training points for class $+1$ and -1.



- **Optimization problem** (if the margin is $2M$)

$$\max_{\beta, \beta_0, \|\beta\|=1} M \quad \text{subject to} \quad y_i(x_i^t\beta + \beta_0) \geq M, i = 1, \ldots, n$$

Separable Case

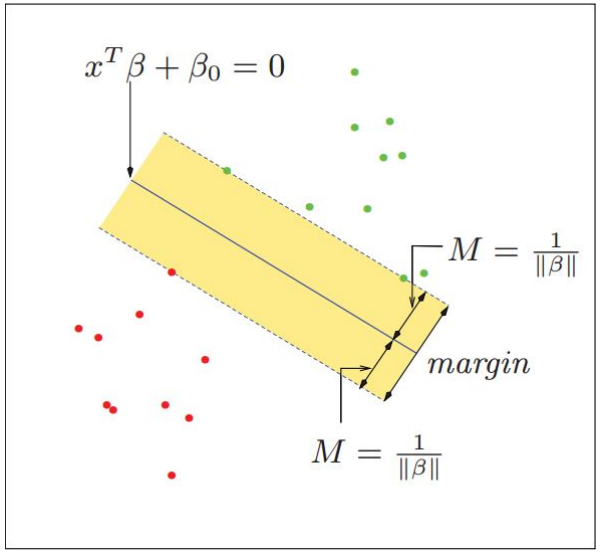


- Rephrasing the optimization problem
  - Distance of x_i to decision boundary is $|x_i^t\beta + \beta_0|/\|\beta\|$
  - Let the x_i's on margin have $|x_i^t\beta + \beta_0| = 1$
  - Then $M = 1/\|\beta\|$

  Then optimization problem can be restated as

$$\max_{\beta,\beta_0} \|\beta\|^2 \quad \text{subject to} \quad y_i(x_i^t\beta + \beta_0) \geq 1, i = 1, \ldots, n$$

Separable Case



- Rephrasing the optimization problem
  - Distance of x_i to decision boundary is $|x_i^t\beta + \beta_0|/\|\beta\|$
  - Let the x_i's on margin have $|x_i^t\beta + \beta_0| = 1$
  - Then $M = 1/\|\beta\|$

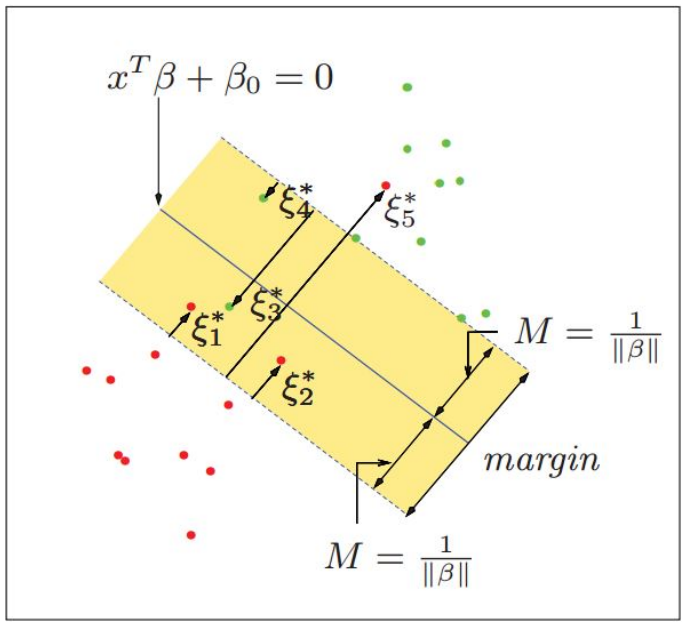
  Then optimization problem can be restated as

$$\max_{\beta,\beta_0} \|\beta\|^2 \quad \text{subject to} \quad y_i(x_i^t\beta + \beta_0) \geq 1, i = 1, \ldots, n$$

- This is a **QP** problem with linear inequality constraints.

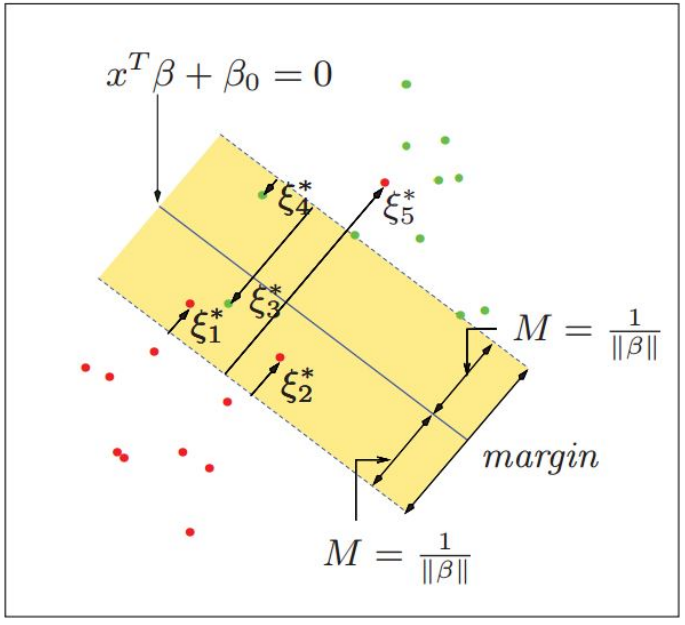
Non-separable Case

- **Training Data**: $(x_1, y_1), \ldots, (x_n, y_n)$ overlap $\longrightarrow$ no feasible solution to previous optimization



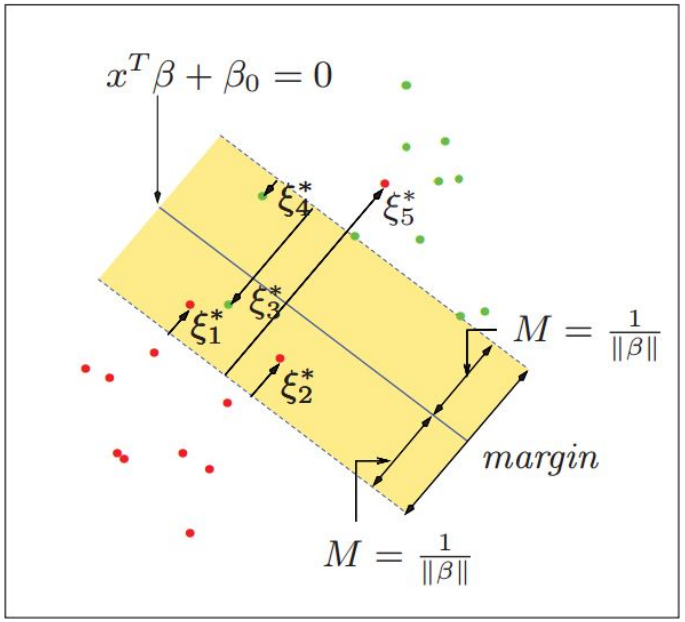
- **Instead**: Find hyperplane with biggest margin but allow some points to be on the wrong side of the margin.
- Introduce slack variables $\xi = (\xi_1, \xi_2, \ldots, \xi_n)$ for each example with $\xi_i \geq 0$.

Non-separable Case



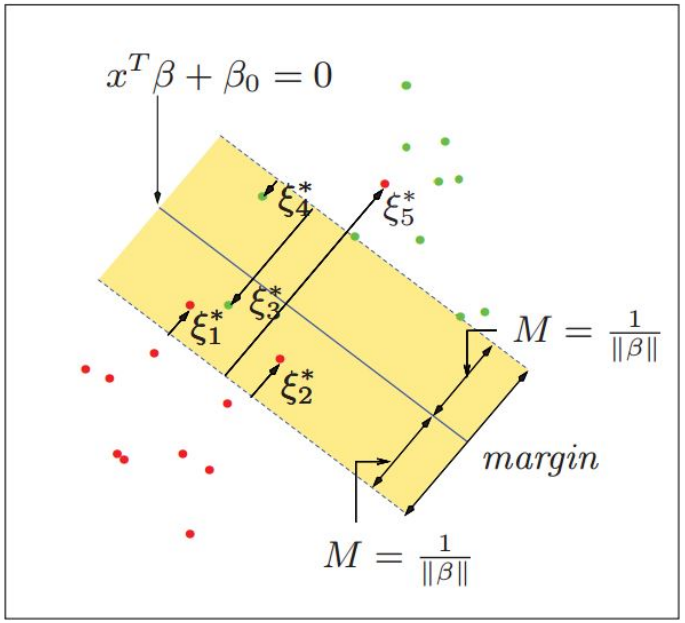
- Modify constraints: $y_i(x_i^t\beta + \beta_0) \geq M$ to $y_i(x_i^t\beta + \beta_0) \geq M - \xi_i$ with $\xi \geq 0, \forall i$ and $\sum \xi_i \leq C$
- Intuitively appealing — measure amount of overlap in real distances to margin — but it does not lead to a convex problem.

Non-separable Case



- Modify constraints: $y_i(x_i^t\beta + \beta_0) \geq M$ to $y_i(x_i^t\beta + \beta_0) \geq M - \xi_i$ with $\xi \geq 0, \forall i$ and $\sum \xi_i \leq C$
- Intuitively appealing — measure amount of overlap in real distances to margin — but it does not lead to a convex problem.
- **Instead**: Modify $y_i(x_i^t\beta + \beta_0) \geq M$ to $y_i(x_i^t\beta + \beta_0) \geq M(1 - \xi_i)$ with $\xi \geq 0, \forall i$ and $\sum \xi_i \leq C$
- Overlap now measured in relative distances which changes with M but now have a convex problem!

Non-separable Case: Optimization Formulation

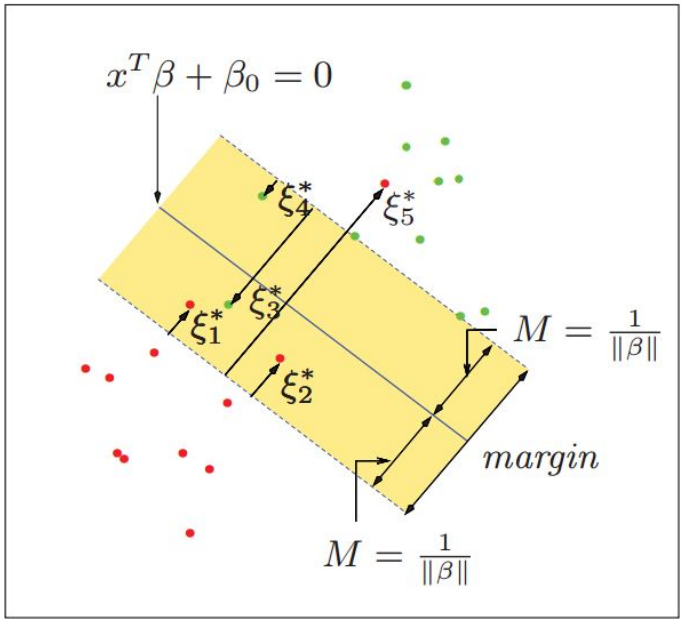


- $\xi \geq 0$ in

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

  is the proportional amount by which the prediction $f(x_i) = x_i^t\beta + \beta_0$ is on the wrong side of the margin.
- $\xi > 1 \implies$ prediction on wrong side of margin
- $\sum \xi_i \leq K \implies$ at most K training pts on wrong side of margin.

Non-separable Case: Optimization Formulation



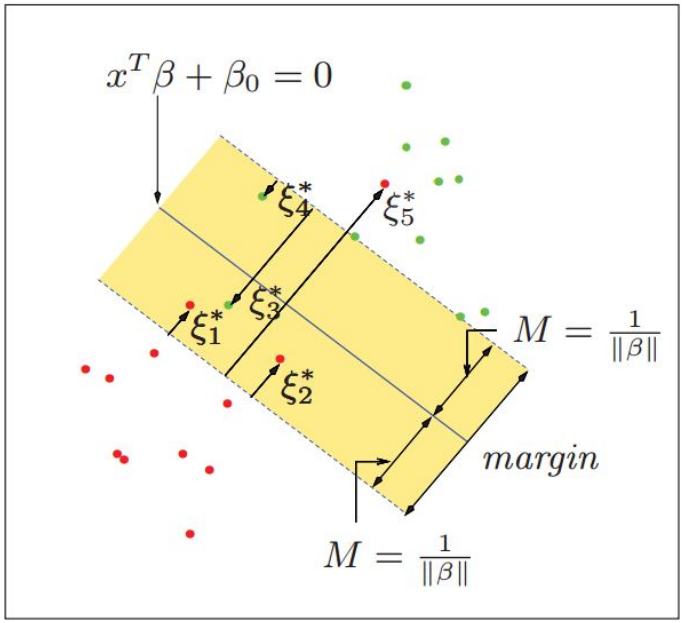
- **Optimization Statement I**

$$\max_{\beta,\beta_0,\|\beta\|=1} M \quad \text{subject to} \quad y_i(x_i^t\beta + \beta_0) \geq M(1-\xi), \forall i$$

$$\xi \geq 0, \forall i, \quad \sum \xi_i \leq constant$$

- Can drop constraint $\|\beta\| = 1$ by choosing $|x_i^t\beta + \beta_0| = 1$ for x_i on the margin.
- $\Longrightarrow M = 1/\|\beta\|$ **and** constraint $y_i(x_i^t\beta + \beta_0) \geq M(1-\xi_i)$ becomes $y_i(x_i^t\beta + \beta_0) \geq 1 - \xi_i$

Non-separable Case: Optimization Formulation



- **Optimization Statement II**

$$\max_{\beta,\beta_0} \|\beta\|^2 \quad \text{subject to} \quad y_i(x_i^t\beta + \beta_0) \geq 1 - \xi, \forall i$$

$$\xi_i \geq 0, \forall i, \quad \sum \xi_i \leq constant$$

- Note x_i's inside their class boundary do not play a role in shaping the decision boundary.
- $\Longrightarrow$ $\beta^{(SVM)}$ differs from $\beta^{(LDA)}$ as $\beta^{(LDA)}$ is influenced by all the training points.

Computing the Support Vector Classifier

- **Optimization Statement III**

$$\max_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n}\xi_i \quad \text{subject to} \quad y_i(x_i^t\beta + \beta_0) \geq 1 - \xi, \forall i$$

$$\xi_i \geq 0, \forall i$$

where cost parameter C replaces the constraint $\sum \xi \leq constant$ (separable case corresponds to $C = \inf$).

- To solve this constrained optimization construct its Lagrangian $\mathcal{L}_P(\beta, \beta_0, \xi, \alpha, \mu) =$
$\frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i[y_i(x_i^t\beta+\beta_0) - (1-\xi_i)] - \sum_{i=1}^{n}\mu_i\xi_i$
with $\alpha \geq 0$ and $\mu_i \geq 0\ \forall i$.
- If $(\beta^*, \beta_0^*, \xi^*)$ is minimum for the constraint problem then $\exists \alpha^*$ and μ^* s.t. $(\beta^*, \beta_0^*, \xi^*, \alpha^*, \mu^*)$ is a stationary point of $\mathcal{L}_P$.

The KKT conditions for the primal problem

For $i = 1, \ldots, n$:

$$\frac{\partial \mathcal{L}_P}{\partial \beta} = \beta - \sum_i \alpha_i x_i y_i = 0$$

$$\frac{\partial \mathcal{L}_P}{\partial \beta_0} = -\sum_i \alpha_i y_i = 0$$

$$\frac{\partial \mathcal{L}_P}{\partial \xi_i} = C - \alpha_i - \mu_i = 0$$

$$y_i(x_i^t \beta + \beta_0) - (1 - \xi_i) \geq 0$$

$$\xi_i \geq 0$$

$$\alpha_i \geq 0$$

$$\mu_i \geq 0$$

$$\alpha_i[y_i(x_i^t \beta + \beta_0) - (1 - \xi_i)] = 0$$

$$\mu_i \xi_i = 0$$

Computing the Lagrangian Dual

- Set the derivatives $\partial \mathcal{L}_P / \partial \beta, \ldots$ to 0 gives

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

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

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

- Remember $\alpha_i, \mu_i, \xi_i \geq 0, \quad \forall i$
- By substituting these into the Lagrangian get the dual

$$\begin{aligned} \mathcal{L}_D(\alpha, \mu) &= \inf_{\beta, \beta_0, \xi} \mathcal{L}_P(\beta, \beta_0, \xi, \alpha, \mu) \\ &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{i'=1}^{n} \alpha_i \alpha_{i'} y_i y_{i'} x_i^t x_{i'} \end{aligned}$$

Dual optimization problem

- Maximize

$$\mathcal{L}_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{i'=1}^{n} \alpha_i \alpha_{i'} y_i y_{i'} x_i^t x_{i'}$$

  subject to the constraints

$$0 \leq \alpha_i \leq C, \quad \text{for } i = 1, \ldots, n \quad \text{and} \quad \sum_i \alpha_i y_i = 0$$

- An easier QP than the original primal problem especially if $p > n$.
- Solving the dual problem for the SVM is equivalent to solving the primal problem as
  - The primal problem is convex
  - constraints are affines

Comments on the solution

Solution for β has the form

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

- $\hat{\alpha}_i$ is non-zero only if $y_i(\hat{\beta}_i^t x_i + \hat{\beta}_0) = 1 - \xi_i$
- These observations are the support vectors.
- For support vectors with $\xi_i = 0_{\text{(on the margin)}}$ then $0 < \hat{\alpha}_i < C$
- The other support vectors have $\xi_i > 0$ with $\hat{\alpha}_i = C$

The Mixture Example

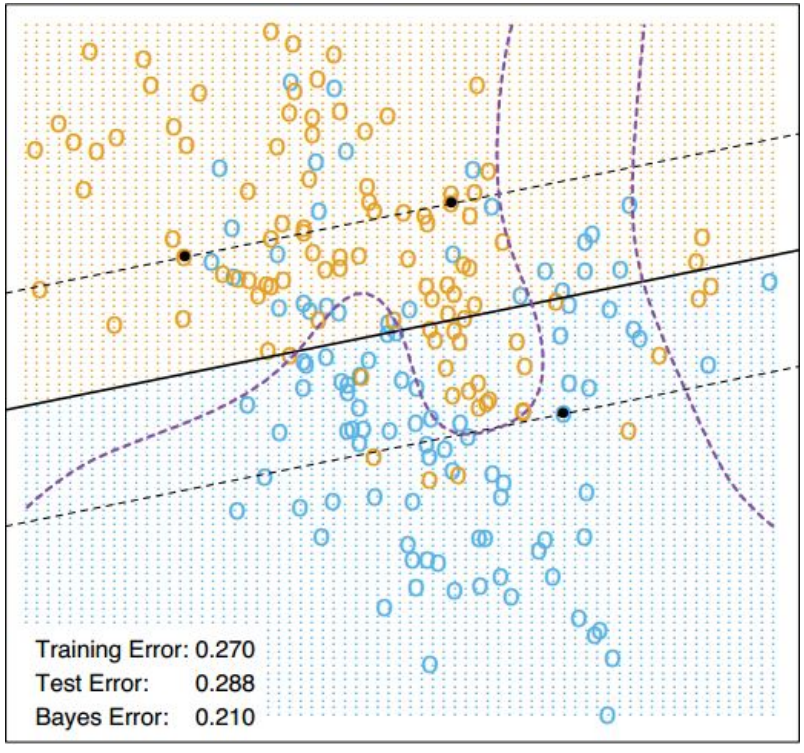


Figure: $C = 10000$

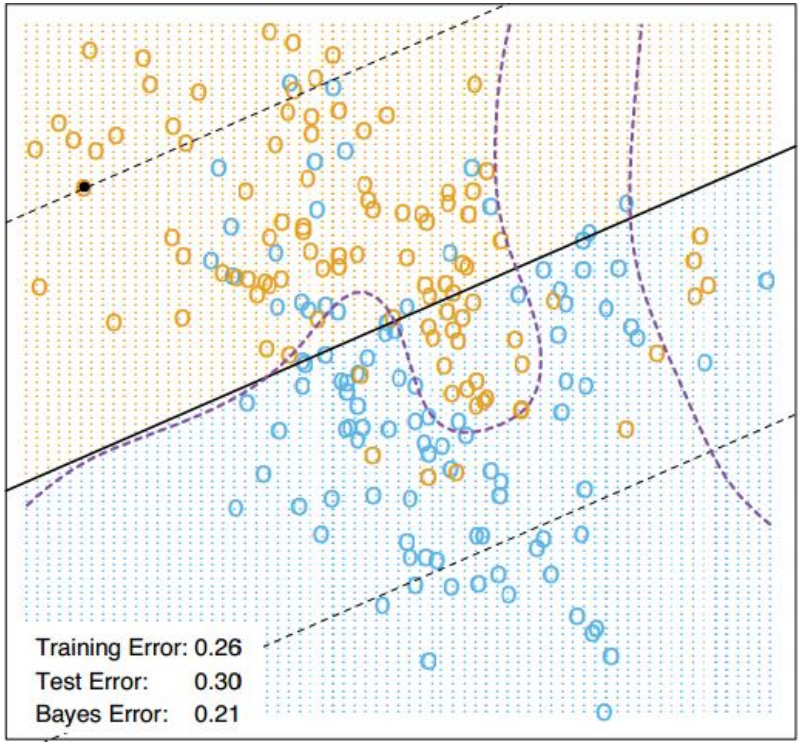


Figure: $C = 0.01$

Support vectors are points

- on the wrong side of boundary and
- on the correct side of the boundary but in the margin

The Support Vector Machines and Kernels

Computing the SVM for Classification

- Can find non-linear decision boundaries by using basis expansion.
- Select basis functions $h_m(x), m = 1, \ldots, M$ and proceed as before.
- Fit SV classifier using inputs $h(x_i) = (h_1(x), \ldots, h_M(x_i))^t$
- This produces the non-linear function

$$\hat{f}(x) = h(x)^t \hat{\beta} + \hat{\beta}_0$$

- The classifier is $\hat{G}(x) = sign\{\hat{f}(x)\}$
- The Support Vector Machine is an extension of this idea where M can get very large or even infinite.

Computing the SVM for Classification

- Can represent the optimization problem

$$\max_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n}\xi_i \quad \text{subject to} \quad y_i(x_i^t\beta + \beta_0) \geq 1 - \xi, \forall i$$

$$\xi_i \geq 0, \forall i$$

  and its solution that only involves input features via inner products.

- $\Longrightarrow$ for particular choices of h these inner products can be computed very cheaply.

Computing the SVM for Classification

- The dual optimization problem is to maximize

$$\mathcal{L}_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle h(x_i), h(x_j) \rangle$$

  subject to $\sum_i \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C$ for $i = 1, \ldots, n$.

- The solution function $\hat{f}(x)$ can be written as

$$\begin{aligned} \hat{f}(x) &= h(x)^t \hat{\beta} + \hat{\beta}_0 \\ &= \sum_{i=1}^{n} \hat{\alpha}_i y_i \langle h(x), h(x_i) \rangle + \hat{\beta}_0 \end{aligned}$$

- Both these equations only involve $h(x)$ through inner-products.
- $\Longrightarrow$ only need knowledge of the **kernel function** that computes the inner-product in the transformed space

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

The Kernel function

- The solution $\hat{f}$ can then be written as

$$\begin{aligned}\hat{f}(x) &= h(x)^t\hat{\beta} + \hat{\beta}_0 \\ &= \sum_{i=1}^{n} \hat{\alpha}_i y_i \langle h(x), h(x_i)\rangle + \hat{\beta}_0 \\ &= \sum_{i=1}^{n} \hat{\alpha}_i y_i K(x, x_i) + \hat{\beta}_0\end{aligned}$$

- K should be a symmetric positive semi-definite function.
- Popular choices for K in the SVM literature are

$$\begin{aligned}d\text{th-Degree polynomial:} \quad & K(x, x') = (1 + \langle x, x'\rangle)^d \\ \text{Radial basis:} \quad & K(x, x') = exp\{-\gamma\|x - x'\|^2\} \\ \text{Neural network:} \quad & K(x, x') = tanh\{\gamma_1\langle x, x'\rangle + \gamma_2\}\end{aligned}$$

Example: Kernel and induced feature mapping

- Consider a 2D feature vector $X = (X_1, X_2)$ and a polynomial kernel of degree 2:

$$\begin{aligned} K(X, X') &= (1 + \langle X, X' \rangle)^2 \\ &= (1 + X_1 X_1' + X_2 X_2')^2 \\ &= 1 + 2X_1 X_1' + 2X_2 X_2' + (X_1 X_1')^2 + (X_2 X_2')^2 + 2X_1 X_1' X_2 X_2' \end{aligned}$$

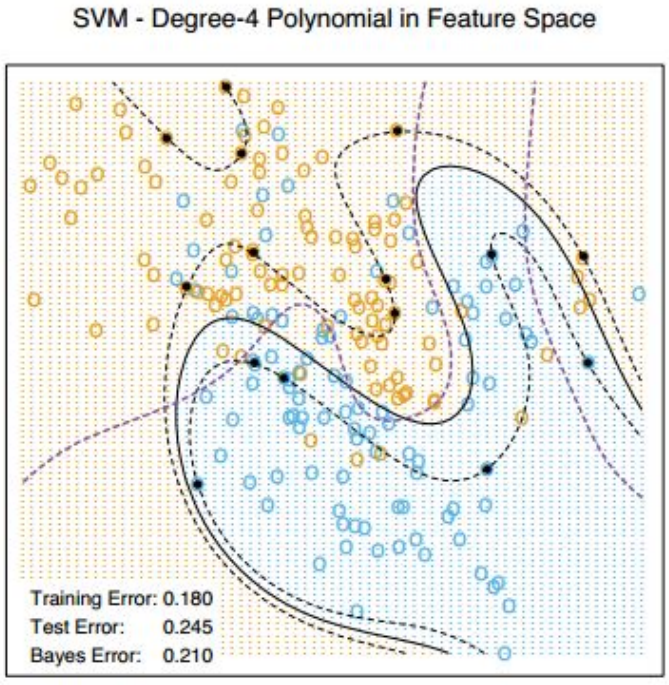
- In this case $M = 6$ with

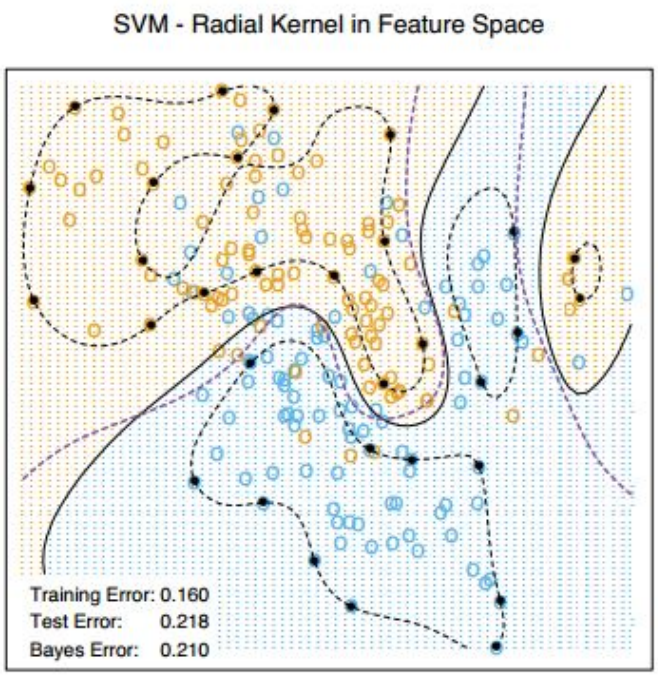
$$\begin{aligned} h_1(X) &= 1 & h_2(X) &= \sqrt{2} X_1 \\ h_3(X) &= \sqrt{2} X_2 & h_4(X) &= X_1^2 \\ h_5(X) &= X_2^2 & h_6(X) &= \sqrt{2} X_1 X_2 \end{aligned}$$

- Obviously then

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

Example SVM Decision Boundaries





$C = 1$ in both cases

Dashed purple line is the Bayes decision boundary.
Dashed black lines are $yf(x) = \pm 1$.

The Role of C

- Large C discourages any positive ξ_i which may lead to overfit wiggly boundary.
- Small C encourages a small value of $\|\beta\| \Longrightarrow$ a smoother decision boundary.

The SVM as a Penalization Method

With $f(x) = h(x)^t\beta + \beta_0$ consider this optimization problem:

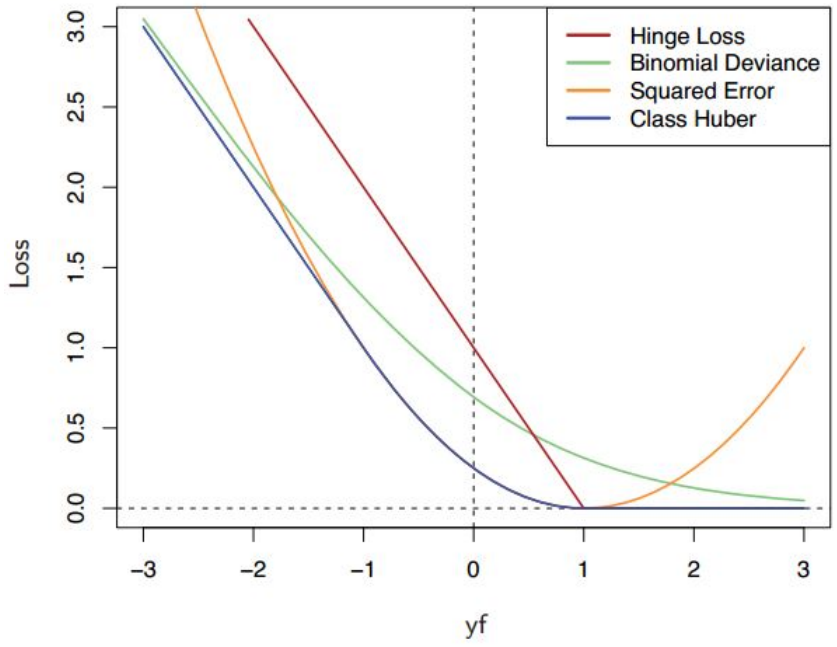
$$\min_{\beta,\beta_0} \sum_{i=1}^{n} \max(0, 1 - y_i f(x_i)) + \frac{\lambda}{2}||\beta||^2$$

- This cost function has the form

  Loss + **Penalty**

- soln this optimization problem $=$ soln to typical formulation of SVM If $\lambda = 1/C$.
- Loss function $max(0, 1 - yf(x))$ is known as the **Hinge Loss**.

Comparison of loss functions



Loss function	$L(y, f(x))$	Minimizing $f(x)$
Binomial deviance	$\log(1+\exp\{-yf(x)\})$	$\log \frac{P(Y=1\|x)}{P(Y=0\|x)}$
SVM Hinge Loss	$\max\{0,1-yf(x)\}$	$\text{sign}\{P(Y=1\|x)-.5\}$
Squared error	$(1-yf(x))^2$	$2P(Y=1\|x)-1$
Huberised squared err	$\begin{cases} -4yf(x) & \text{if } yf(x) < 1 \\ \max\{0,1-yf(x)\} & \text{otherwise} \end{cases}$	$2P(Y=1\|x)-1$

Function Approximation and Reproducing Kernels

- Suppose positive definite kernel K has eigen-expansion

$$K(x, x') = \sum_{m=1}^{\infty} \delta_m \phi_m(x) \phi_m(x) = \sum_{m=1}^{\infty} h_m(x) h_m(x)$$

  if $h_m(x) = \sqrt{\delta_m}\phi_m(x)$.

- Then $f(x) = \beta_0 + \sum_m \beta_m h_m(x) = \beta_0 + \sum_m \beta_m \sqrt{\delta_m}\phi_m(x)$

- Let $\theta_m = \sqrt{\delta_m}\beta_m \implies$ can write the SVM cost function as

$$\min_{\beta_0, \theta} \sum_{i=1}^{n} \max\{0, 1 - y_i(\beta_0 + \sum_{m=1}^{\infty} \theta_m \phi_m(x_i))\} + \frac{\lambda}{2} \sum_m \frac{\theta_m^2}{\delta_m}$$

- Theory of RKHS guarantees $\exists$ a finite dimensional solution

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

Function Approximation and Reproducing Kernels

- An equivalent version of the SVM optimization is

$$\min_{\beta_0,\alpha} \sum_{i=1}^{n} \max\{0, 1 - y_i(\beta_0 + \sum_{j=1}^{n} \alpha_j K(x_j, x_i))\} + \frac{\lambda}{2}\alpha^t \boldsymbol{K}\alpha$$

  where $\boldsymbol{K} = (K(x_i, x_j))_{ij}$ is the $n \times n$ matrix of pairwise kernel evaluations.

- Such models are quite general and can be expressed more generally as

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{n} \max\{0, 1 - y_i f(x_i)\} + \lambda J(f)$$

  where $\mathcal{H}$ is the structured space of functions and $J(f)$ is a regularizer on this space.

Function Approximation and Reproducing Kernels

Example

- Suppose $\mathcal{H}$ is the space of additive functions

$$f(x) = \sum_{j=1}^{p} f_j(x_j) \quad \textbf{and} \quad J(f) = \sum_{j=1}^{p} \int \{f_j''(x_j)\}^2 dx_j$$

- Then the solution to

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{n} \max\{0, 1 - y_i f(x_i)\} + \lambda J(f)$$

  is an additive cubic spline and has a kernel representation

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i K(x, x_i) \quad \textbf{with} \quad K(x, x') = \sum_{j=1}^{p} K_j(x_j, x_j')$$

- Each of the K_j is the kernel appropriate for the univariate smoothing spline in x_j.

Function Approximation and Reproducing Kernels

- If have
  - any of the kernels mentioned **and**
  - any convex loss function

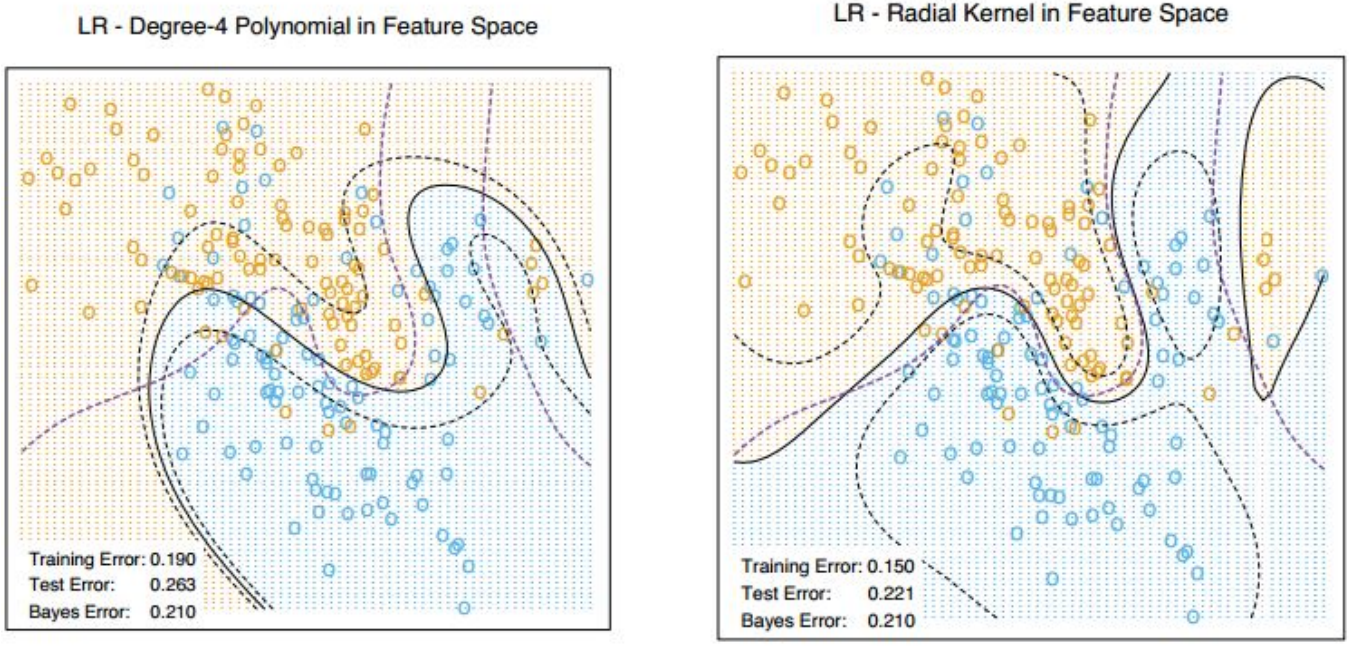
  then optimization leads to a finite representation

$$\sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda J(f) \Longrightarrow \hat{f}(x) = \hat{\beta}_0 + \sum_{i=1}^{n} \hat{\alpha}_i K(x, x_i)$$

- Can use the binomial log-likelihood as a loss function.
- In this cases

$$\hat{f}(x) = \hat{\beta}_0 + \sum_{i=1}^{n} \hat{\alpha}_i K(x, x_i) = \log \frac{\hat{P}(Y = +1|x)}{\hat{P}(Y = -1|x)}$$

Example Decision Boundaries Log-Likelihood Loss



Dashed purple line is the Bayes decision boundary.

Dashed black lines are posterior probabilities of 0.75 and 0.25 for the $+1$.

SVMs and the Curse of Dimensionality

- For the 2nd degree polynomial kernel of p dimensional features is

$$K(X, X') = 1 + 2\sum_{j=1}^{p} X_j X'_j + \sum_{j=1}^{p}(X_j X'_j)^2 + 2\sum_{i=1}^{p}\sum_{j=i+1}^{p} X_i X'_i X_j X'_j$$

- Kernel cannot adapt to concentrate on just a subset of the features.
- Disaster if p is large and only a small subset of features are relevant.
- If knew beforehand which features were relevant could adapt the kernel accordingly, but...
- $\Longrightarrow$ SVM kernel methods alone are not ideal for discovering structure.

SVMs and the Curse of Dimensionality

Example

- Case I: No noise features
  - Class 1:
    $X = (X_1, X_2, X_3, X_4)^t$, where each $X_i \sim \mathcal{N}(0, 1)$ and are indept.
  - Class 2:
    $X = (X_1, X_2, X_3, X_4)^t$, with $X_i \sim \mathcal{N}(0, 1)$ and $9 \leq \sum_{j=1}^{4} X_j^2 \leq 16$

- Case II: case I augmented with noise features
  - Class 1:
    $X = (X_1, \ldots, X_{10})^t$, where each $X_i \sim \mathcal{N}(0, 1)$ and are indept.
  - Class 2:
    $X = (X_1, \ldots, X_{10})^t$, with $X_i \sim \mathcal{N}(0, 1)$ and $9 \leq \sum_{j=1}^{4} X_j^2 \leq 16$

SVMs and the Curse of Dimensionality

For the two cases

- Generated 100 training examples for each class and 1000 test examples.

- Run 50 simulations and averaged the test error.

	Method	Test Error (SE)	
		No Noise Features	Six Noise Features
1	SV Classifier	0.450 (0.003)	0.472 (0.003)
2	SVM/poly 2	0.078 (0.003)	0.152 (0.004)
3	SVM/poly 5	0.180 (0.004)	0.370 (0.004)
4	SVM/poly 10	0.230 (0.003)	0.434 (0.002)
5	BRUTO	0.084 (0.003)	0.090 (0.003)
6	MARS	0.156 (0.004)	0.173 (0.005)
	Bayes	0.029	0.029

BRUTO fits an additive spline model adaptively.
MARS fits a low-order interaction model adaptively.

Summary of Results

	Method	Test Error (SE)	
		No Noise Features	Six Noise Features
1	SV Classifier	0.450 (0.003)	0.472 (0.003)
2	SVM/poly 2	0.078 (0.003)	0.152 (0.004)
3	SVM/poly 5	0.180 (0.004)	0.370 (0.004)
4	SVM/poly 10	0.230 (0.003)	0.434 (0.002)
5	BRUTO	0.084 (0.003)	0.090 (0.003)
6	MARS	0.156 (0.004)	0.173 (0.005)
	Bayes	0.029	0.029

- **Note**: For each SVM classifier an optimal C was used.
- A hyperplane cannot separate classes $\Longrightarrow$ linear SVM does poorly.
- Polynomial SVMs do better but are affected by the noise features.
- Second-degree polynomial kernel does best as true decision boundary is a 2nd-degree polynomial.
- Higher degree kernels do much worse.

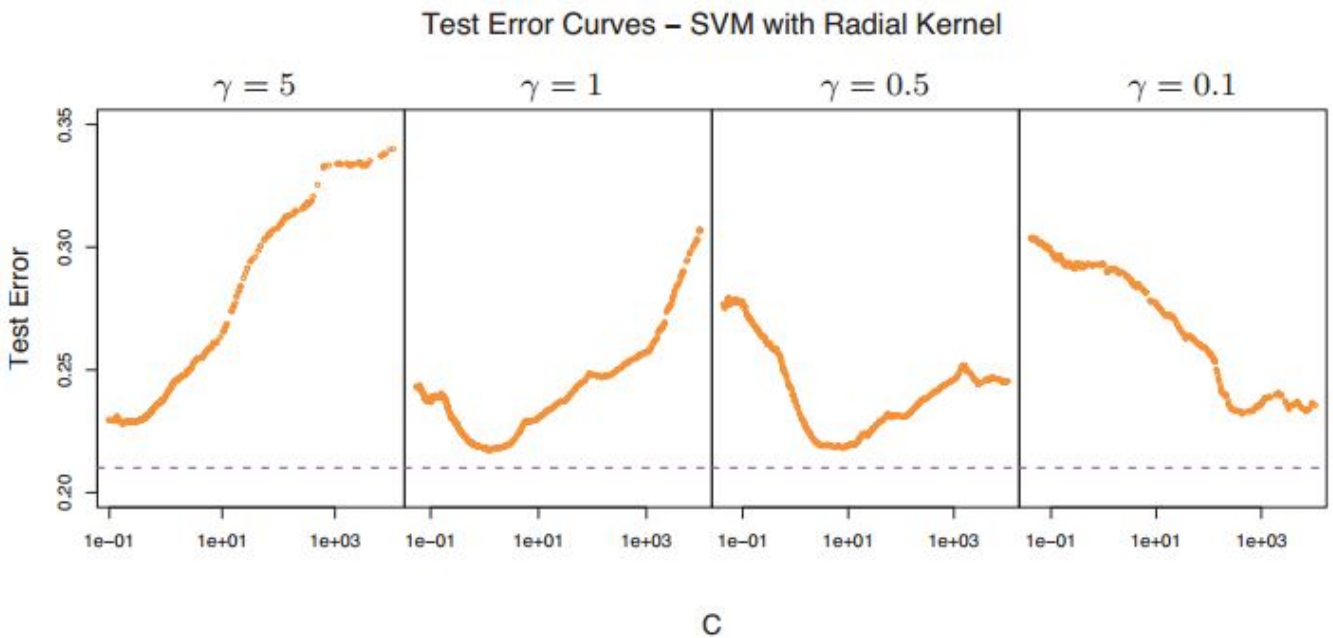
Summary of Results

	Method	Test Error (SE)	
		No Noise Features	Six Noise Features
1	SV Classifier	0.450 (0.003)	0.472 (0.003)
2	SVM/poly 2	0.078 (0.003)	0.152 (0.004)
3	SVM/poly 5	0.180 (0.004)	0.370 (0.004)
4	SVM/poly 10	0.230 (0.003)	0.434 (0.002)
5	BRUTO	0.084 (0.003)	0.090 (0.003)
6	MARS	0.156 (0.004)	0.173 (0.005)
	Bayes	0.029	0.029

- BRUTO performs well as the decision boundary is additive.
- Both BRUTO and MARS can ignore redundant variables
- $\Longrightarrow$ their performances do not deteriorate so much with the addition of noise features

A Path Algorithm for the SVM Classifier

- C is the regularization parameter for the SVM.
- High C leads to overfitting.
- Must consider parameters of the kernel when setting C:



- Usually set C with cross-validation.

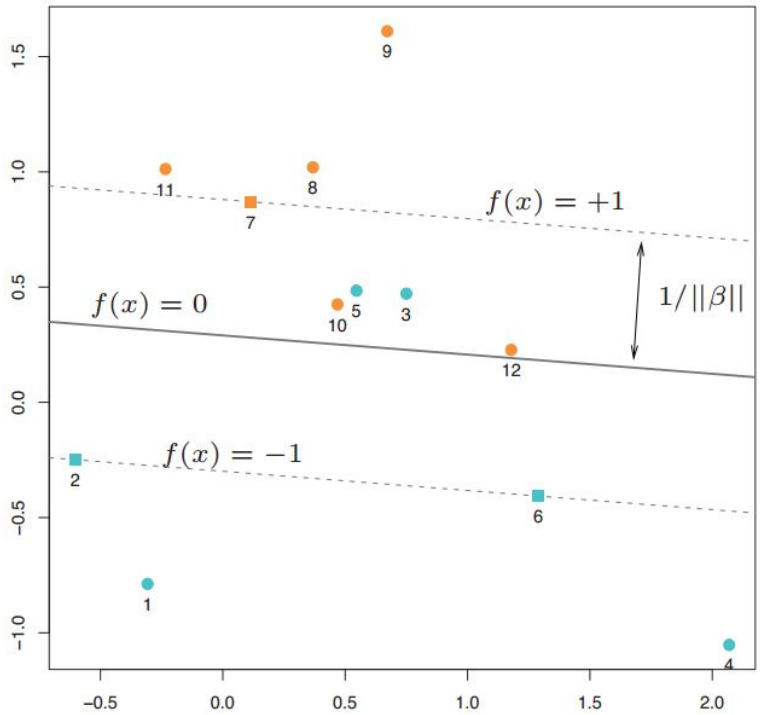
A Path Algorithm for the SVM Classifier

Efficient computation of the SVM models obtained by varying C.

- The solution of loss+penalty formulation of SVM is

$$\beta_\lambda = \frac{1}{\lambda}\sum_{i=1}^{n}\alpha_i y_i x_i$$

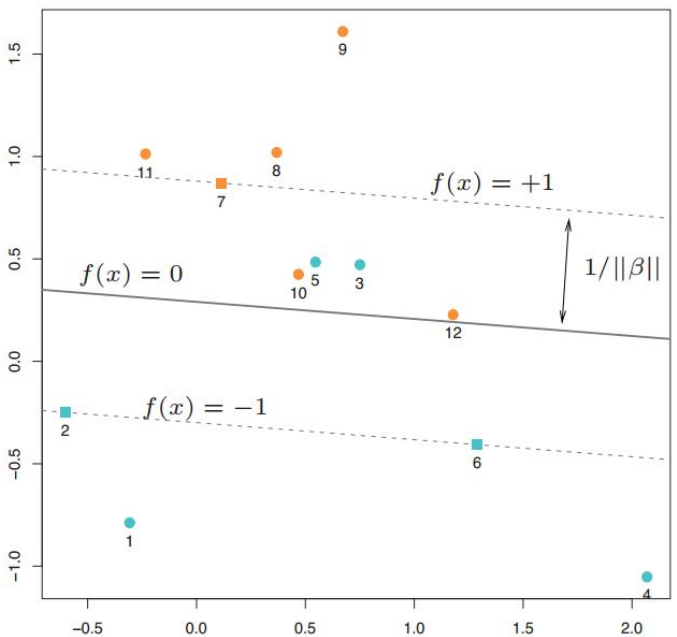
where α_i's are Lagrange multipliers and each $\alpha_i \in [0, 1]$.



A Path Algorithm for the SVM Classifier

Path algorithm relies on the observation

- Labelled points (x_i, y_i) fall into 3 distinct groups
  - **Correctly classified and outside their margin** $\Longrightarrow \alpha_i = 0$
  - **Correctly classified and on margin** $\Longrightarrow \alpha_i \in (0, 1)$
  - **Inside their margins** $(y_i f(x_i) < 1) \implies \alpha_i = 1$



A Path Algorithm for the SVM Classifier

- The idea for the path algorithm is:

$$
\begin{aligned}
\text{Initially } \lambda \text{ is large} &\Longrightarrow \text{margin, } 1/\|\beta_\lambda\|, \text{ is wide} \\
&\Longrightarrow \text{all points have } y_i f(x_i) < 1 \\
&\Longrightarrow \text{all } \alpha_i = 1 \\
\text{As } \lambda \text{ decreases} &\Longrightarrow \text{margin narrows} \\
&\Longrightarrow \text{pts move from inside to outside their margins} \\
&\Longrightarrow \text{their } \alpha_i\text{'s will change from 1 to 0}
\end{aligned}
$$

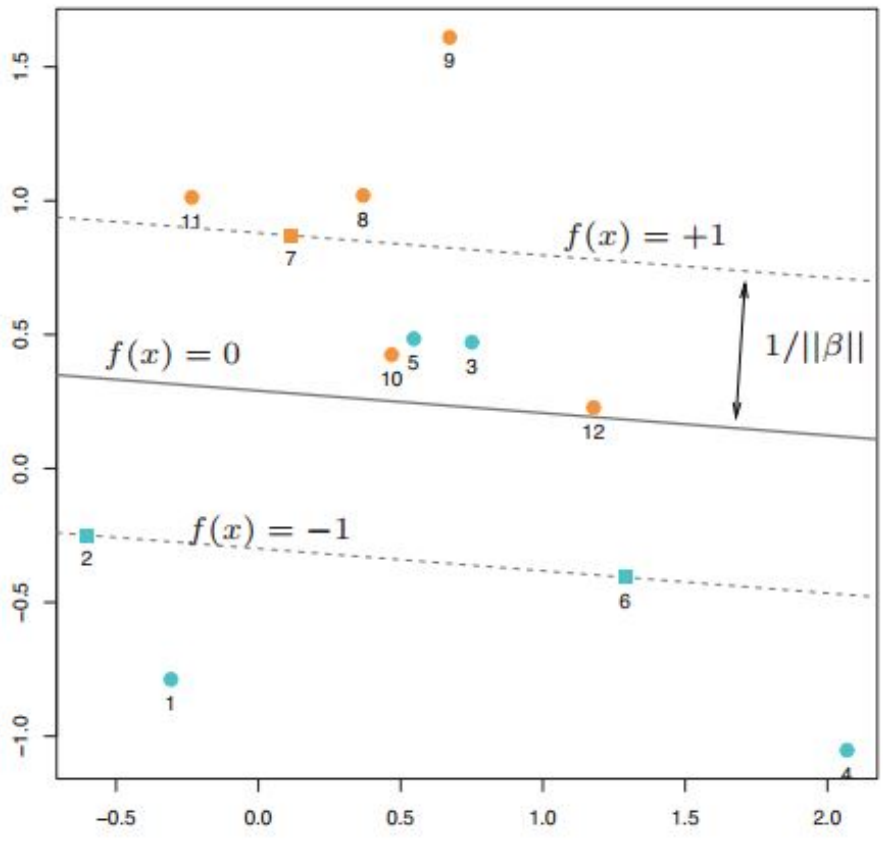
- By the continuity of the $\alpha_i(\lambda)$ these transition points will linger on the margin during the transition.

- All that changes as λ decreases are the $\alpha_i \in (0, 1)$ of those points on the margin – $y_i f(x_i) = 1$.

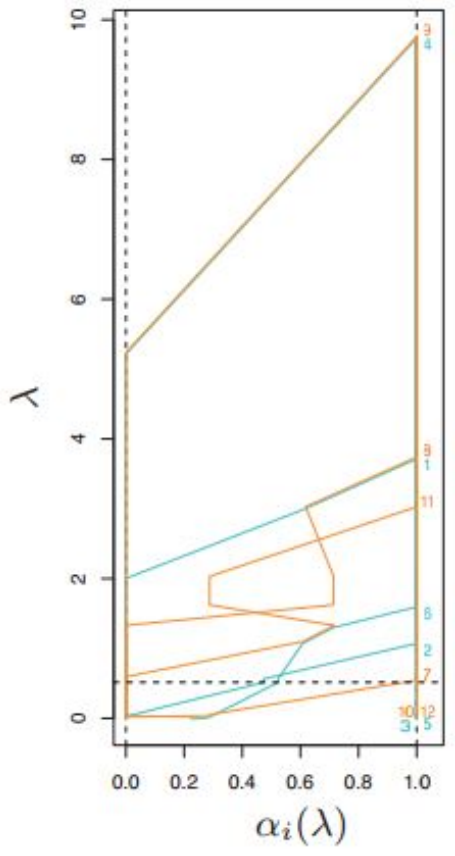
- This gives a small set of linear equations that prescribe how $\alpha_i(\lambda)$ and β_λ change during these transitions.

- This results in a piecewise linear path for each $\alpha_i(\lambda)$.

Example Paths



$\lambda = .05$



Support Vector Machines for Regression

- Can adapt SVMs for regression with a quantitative response.
- Consider the linear regression model

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

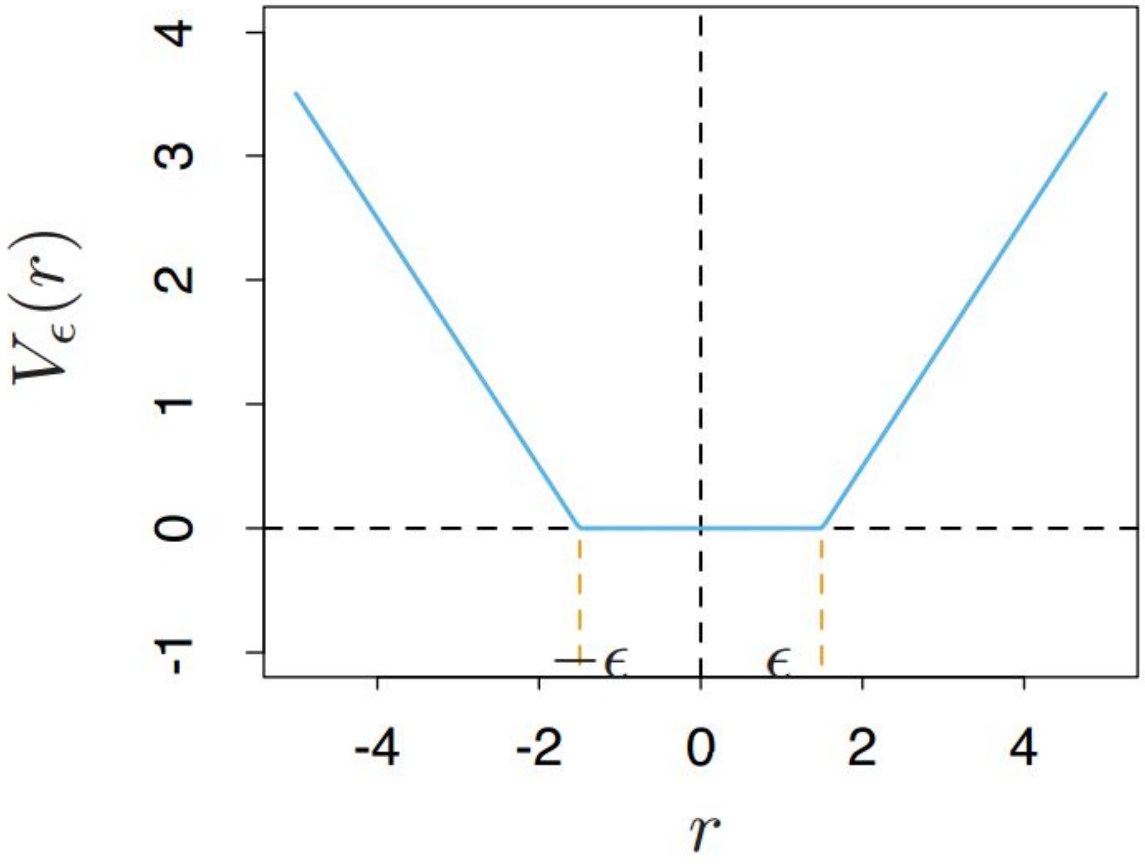
- To estimate β consider minimization of

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

with the ϵ-sensitive error measure

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

ϵ-sensitive Error Measure



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

Properties of ϵ-sensitive Error

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

- "Low-error" points are ignored by V_ϵ.
- V_ϵ has linear tails $\Longrightarrow$ less sensitive to outliers.
- But V_ϵ flattens the contributions of points with small residuals.

Solution to SVR

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

- If $\hat{\beta}, \hat{\beta}_0$ minimize H then

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

and

$$\hat{f}(x) = \sum_{i=1}^{n} (\hat{\alpha}_i^* - \hat{\alpha}) \langle x, x_i \rangle + \hat{\beta}_0$$

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

Solution to SVR

$$\hat{f}(x) = \sum_{i=1}^{n} (\hat{\alpha}_i^* - \hat{\alpha}) \langle x, x_i \rangle + \hat{\beta}_0$$

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

$$\min_{\alpha^*, \alpha} \sum_{i=1}^{n} (\alpha_i^* + \alpha_i) - \sum_{i=1}^{n} y_i (\alpha_i^* - \alpha_i) + \frac{1}{2} \sum_{i,j=1}^{n} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) \langle x_i, x_j \rangle$$

subject to the constraints

$$0 \leq \alpha_i, \alpha_i^* \leq 1/\lambda, \quad \forall i$$

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

$$\alpha_i^* \alpha_i = 0, \quad \forall i$$

Solution to SVR

$$\hat{f}(x) = \sum_{i=1}^{n} (\hat{\alpha}_i^* - \hat{\alpha}) \langle x, x_i \rangle + \hat{\beta}_0$$

- Normally only a subset of the $(\hat{\alpha}_i^* - \hat{\alpha}_i)$ are nonzero.
- x_i's, with non-zero $(\hat{\alpha}_i^* - \hat{\alpha}_i)$, are the support vectors.
- If scale responses, $V_\epsilon(r/\sigma)$, then can use preset values for ϵ.
- Quantity λ can be estimated by cross-validation.
- As solution only depends on $\langle x_i, x_j \rangle$'s can generalize the method to richer spaces using kernels.

Regression and Kernels

- Approximate of f with a set of basis functions $\{h_m(x)\}_{m=1}^{M} s$

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

- To estimate β and β_0 minimize

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

for some error measure $V(r)$.

- For any choice of $V(r)$ the solution $\hat{f}(x)$ has the form

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

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

Regression and Kernels: Squared Error

Consider the case when $V(r) = r^2$ and $\beta_0 = 0$.

- Let $\boldsymbol{H}$ be the $n \times M$ matrix

$$\begin{pmatrix} h_1(x_1) & h_2(x_1) & \cdots & h_M(x_1) \\ h_1(x_2) & h_2(x_2) & \cdots & h_M(x_2) \\ \vdots & \vdots & \vdots & \vdots \\ h_1(x_n) & h_2(x_n) & \cdots & h_M(x_n) \end{pmatrix}$$

- Suppose $M > n$.

- Estimate β by minimizing

$$H(\beta) = (y - \boldsymbol{H}\beta)^t(y - \boldsymbol{H}\beta) + \lambda||\beta||^2$$

- The solution is $\hat{\beta} = (\boldsymbol{H}^t\boldsymbol{H} + \lambda I_M)^{-1}\boldsymbol{H}^t y$

- For test point x its estimate is $(h_1(x), \ldots, h_M(x))\hat{\beta} = h(x)^t\hat{\beta}$

Regression and Kernels: Squared Error

This expression can be re-written as:

$$
\begin{aligned}
h(x)^t\hat{\beta} &= h(x)^t(\boldsymbol{H}^t\boldsymbol{H} + \lambda I_M)^{-1}\boldsymbol{H}^t y \\
&= h(x)^t(\boldsymbol{H}^t\boldsymbol{H} + \lambda I_M)^{-1}\boldsymbol{H}^t(\boldsymbol{H}\boldsymbol{H}^t + \lambda I_n)(\boldsymbol{H}\boldsymbol{H}^{\boldsymbol{t}} + \lambda I_n)^{-1} y \\
&= h(x)^t(\boldsymbol{H}^t\boldsymbol{H} + \lambda I_M)^{-1}(\boldsymbol{H}^t\boldsymbol{H}\boldsymbol{H}^t + \lambda\boldsymbol{H}^t)(\boldsymbol{H}\boldsymbol{H}^{\boldsymbol{t}} + \lambda I_n)^{-1} y \\
&= h(x)^t(\boldsymbol{H}^t\boldsymbol{H} + \lambda I_M)^{-1}(\boldsymbol{H}^t\boldsymbol{H} + \lambda I_n)\boldsymbol{H}^t(\boldsymbol{H}\boldsymbol{H}^{\boldsymbol{t}} + \lambda I_n)^{-1} y \\
&= h(x)^t\boldsymbol{H}^t(\boldsymbol{H}\boldsymbol{H}^{\boldsymbol{t}} + \lambda I_n)^{-1} y \\
&= h(x)^t\boldsymbol{H}^t(K + \lambda I_n)^{-1} y, \qquad K = \boldsymbol{H}\boldsymbol{H}^t \\
&= \sum_{i=1}^{n} \hat{\alpha}_i K(x, x_i), \qquad \hat{\alpha} = (K + \lambda I_n)^{-1} y
\end{aligned}
$$

where $K = \boldsymbol{H}\boldsymbol{H}^t = (\langle h(x_i), h(x_j)\rangle)_{i,j}$ = Gram matrix

$$
\boldsymbol{H}h(x) = \begin{pmatrix} h_1(x_1) & \cdots & h_M(x_1) \\ \vdots & \vdots & \vdots \\ h_1(x_n) & \cdots & h_M(x_n) \end{pmatrix} \begin{pmatrix} h_1(x) \\ \vdots \\ h_M(x) \end{pmatrix} = \begin{pmatrix} K(x, x_1) \\ \vdots \\ K(x, x_n) \end{pmatrix}
$$

Regression and Kernels

- Thus only the inner product kernel $K(x_i, x_j)$ need be evaluated, at the n training points for each i, j and at points x for predictions.
- Careful choice of $h_m \Longrightarrow \boldsymbol{H}\boldsymbol{H}^t$ requires $n^2/2$ evaluations of K, rather than the direct cost $n^2 M$.
- This property depends on the choice of squared norm $\|\beta\|^2$ in the penalty.
- It does not hold for the L_1 norm, which may lead to a superior model.

Generalizing Linear Discriminant Analysis

Virtues of LDA

- **Simple prototype classifier**
  - Compute x's Mahalanobis distance to each centroid.
  - Assign x to class with closes centroid.
  - Use a pooled covariance matrix for the Mahalanobis distance.
- **LDA corresponds to Bayes classifier** if
  - points in each class are $\mathcal{N}(\mu_k, \Sigma) \leftarrow$ same pooled covariance across classes
  - This assumption unlikely to be true.
- **LDA gives a linear decision boundary**
  - Results in simple decision rules
- **LDA provides a low-dimensional view of the data**
  - For K classes can learn $K - 1$ orthogonal projections

Weaknesses of LDA

- Linear decision boundaries may not be adequate
- Single prototype may not be sufficient
- Many (correlated) parameters makes LDA perform poorly

Generalization of LDA presented

- **FlexibleDA**
  - Recast LDA problem as a linear regression problem.
  - Generalize linear regression to more flexible, nonparametric regression.
- **PenalizedDA**
  - Have a high dimensional and correlated set of predictors.
  - Fit an LDA model, but penalize its coefficients to be smooth.
- **MixtureDA**
  - Model each class by $K \geq 2$ Gaussians with different centroids.
  - Every component Gaussian has the same covariance matrix.

Flexible Discriminant Analysis

Optimal Scoring

- Have K classes each with a label $\{1, 2, \ldots, K\}$
- Let function

$$\theta : \{1, 2, \ldots, K\} \to \mathbb{R}$$

assign a score to each class.

- For data $\{(x_i, g_i)\}_{i=1}^n$ with $x_i \in \mathbb{R}^p, \quad g_i \in \{1, 2, \ldots, K\}$ want to solve:

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

subject to :

$$\sum_i \theta(g_i) = 0 \qquad \frac{1}{n} \sum_i \theta^2(g_i) = 1$$

Optimal scoring in matrix notation

- Create the $n \times K$ indicator response matrix Y , that is

$$Y_{ik} = \begin{cases} 1, & if \quad g_i = k \\ 0, & otherwise \end{cases}$$

- Let $\Theta = (\theta(1), \theta(2), \ldots, \theta(k))^t$
- The optimization problem can be written as:

$$\min_{\Theta,\beta} (Y\Theta - X\beta)^t (Y\Theta - X\beta)$$

subject to:

$$\frac{1}{n}\Theta^t Y^t Y \Theta = 1$$

Solution to optimal scoring

- The optimization problem

$$\min_{\Theta,\beta}(Y\Theta - X\beta)^t(Y\Theta - X\beta) \qquad \frac{1}{n}\Theta^t Y^t Y\Theta = 1$$

is solved by

$$\hat{\beta} = (X^t X)^{-1} X^t Y\hat{\Theta}$$

where $\hat{\Theta}$ is a solution to the generalized e-value problem:

$$Y^t X(X^t X)^{-1} X^t Y\Theta = \mu Y^t Y\Theta$$

More general optimal scoring

- More generally can find up to $L \leq K-1$ indept scorings:

$$\theta_l : \{1, 2, \ldots, K\} \to \mathbb{R}, \quad l = 1, \ldots, L$$

and L vectors of linear coefficients $\beta_1, \ldots, \beta_L$

- Want to solve:

$$\min_{\beta_l, \theta_l} \sum_{l=1}^{L} \sum_{i=1}^{n} (Y\Theta_l - x_i^t \beta_l)^2$$

subject to:

$$\frac{1}{n} \Theta_l^t Y^t Y \Theta_l = 1, \quad l = 1, \ldots, L$$

and

$$\Theta_l^t \Theta_k = 0, l \neq k$$

Link to LDA

- The sequence of discriminant vectors by LDA are identical to the sequence $\hat{\beta}_l$ up to a constant.
- Mahalanobis distance of point x to the kth class centroid $\hat{\mu_k}$

$$\delta(x, \hat{\mu_k}) = \sum_{l=1}^{K-1} \omega_l (\hat{\beta}_l^t x - \hat{\beta}_l^t \mu_k)^2 + D(x)$$

where ω_l is defined in terms of the mean squared residual r_l^2 of the lth optimally scored fit:

$$\omega_l = \frac{1}{r_l^2(1 - r_l^2)}$$

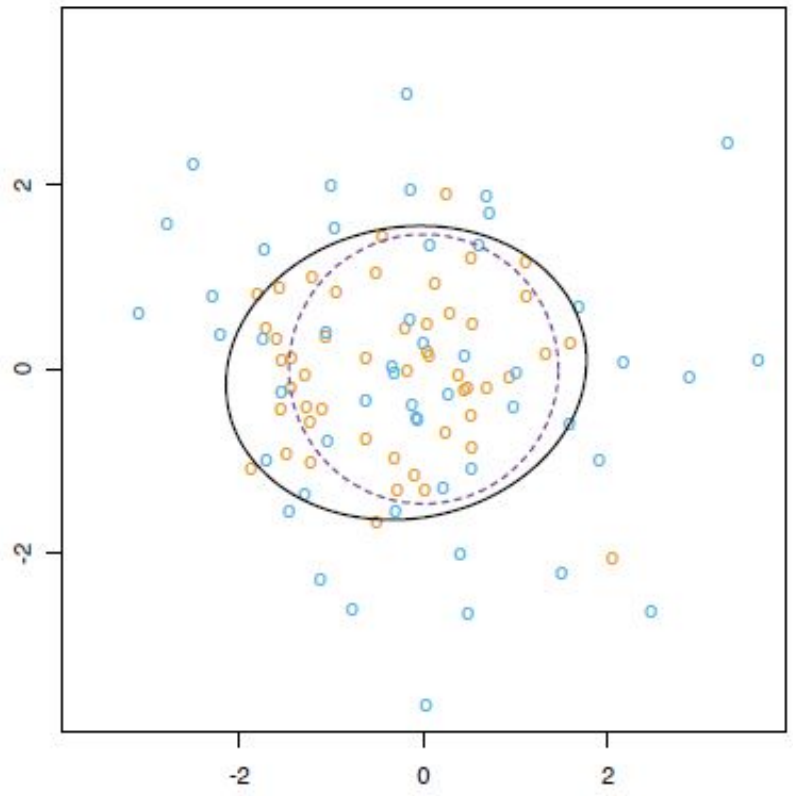
- LDA can be performed by a sequence of linear regressions, followed by classification to the closest class centroid in the space of fits....

- Can replace the linear regression fits $x_i^t \beta_l$ with more flexible, nonparametric fits: $\eta_l(x)$
- $\Longrightarrow$ more flexible classifier than LDA
- General form of the regression problem

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

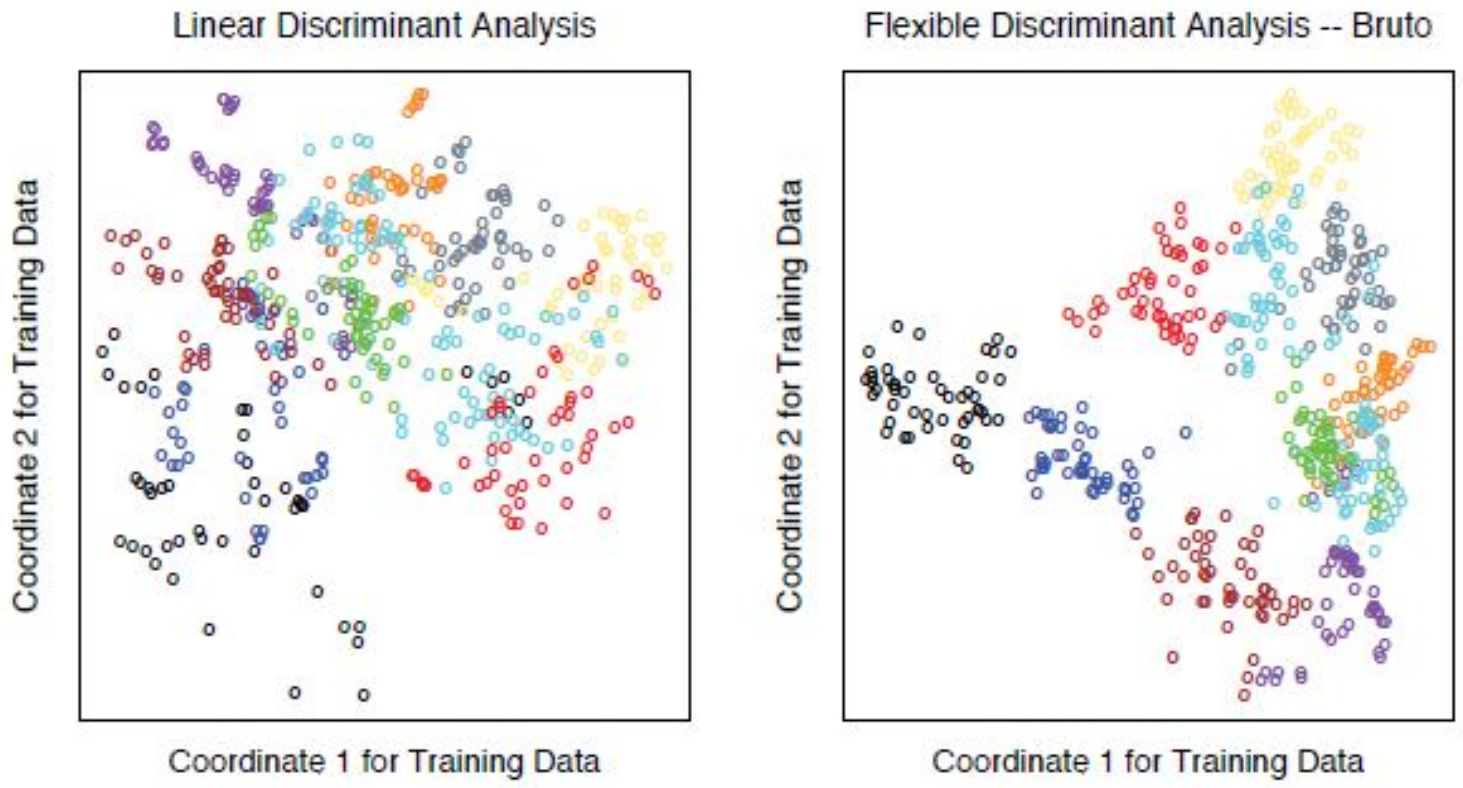
where J is an appropriate regularizer.

Simple Example



Dashed purple line is the Bayes decision boundary.
Black ellipse is the decision boundary found by FDA using 4 degree-twopolynomial regression.

LDA canonical variates for the vowel data



Vowel recognition data performance results

	Technique	Error Rates	
		Training	Test
(1)	LDA	0.32	0.56
	Softmax	0.48	0.67
(2)	QDA	0.01	0.53
(3)	CART	0.05	0.56
(4)	CART (linear combination splits)	0.05	0.54
(5)	Single-layer perceptron		0.67
(6)	Multi-layer perceptron (88 hidden units)		0.49
(7)	Gaussian node network (528 hidden units)		0.45
(8)	Nearest neighbor		0.44
(9)	FDA/BRUTO	0.06	0.44
	Softmax	0.11	0.50
(10)	FDA/MARS (degree = 1)	0.09	0.45
	Best reduced dimension (=2)	0.18	0.42
	Softmax	0.14	0.48
(11)	FDA/MARS (degree = 2)	0.02	0.42
	Best reduced dimension (=6)	0.13	0.39
	Softmax	0.10	0.50

Computing the FDA Estimate

- FDA computations simpler when can write non-parametric regression as a linear operator
- $\Longrightarrow \hat{y} = S_\lambda y, \quad e.g.$ additive splines
- Optimal scoring can be computed by a single eigen-decomposition.
- The algorithm is as follows...

Computing the FDA Estimate

- Create the $n \times K$ indicator response matrix Y, that is

$$Y_{ik} = \begin{cases} 1, & if \quad g_i = k \\ 0, & otherwise \end{cases}$$

- **Multivariate nonparametric regression:**
  - Fit a nonparametric regression of Y on X.
  - Let $\hat{Y} = S_\lambda Y$
  - Let $\eta^*(x)$ be the vector of fitted regression functions.
- **Optimal scores:**
  - Compute the eigen-decomposition of $Y^t \hat{Y} = Y^t S_\lambda Y$
  - Eigenvectors Θ are normalized : $\Theta^t Y^t Y \Theta / n = I$
- **Update** the model : $\eta(x) = \Theta^t \eta^*(x)$

Penalized Discriminant Analysis

Penalized Discriminant Analysis

- Consider
  - linear regression onto a basis expansion with
  - a quadratic penalty on the coefficients
- Find $(\theta_l, \beta_l)_{l=1}^{L}$ which minimize:

$$\frac{1}{n}\sum_{l=1}^{L}\left[\sum_{i=1}^{n}(\theta_l(g_i) - h^t(x_i)\beta_l)^2 + \lambda\beta_l^t\Omega\beta_l\right]$$

- Choice of Ω depends on the problem.
- Generalization of LDA called penalized discriminant analysis (PDA).

Steps of Penalized Discriminant Analysis

- Extract a basis expansion $h(x_i)$ of each x_i.
- The penalized Mahalanobis distance to a class centroid is given by:

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

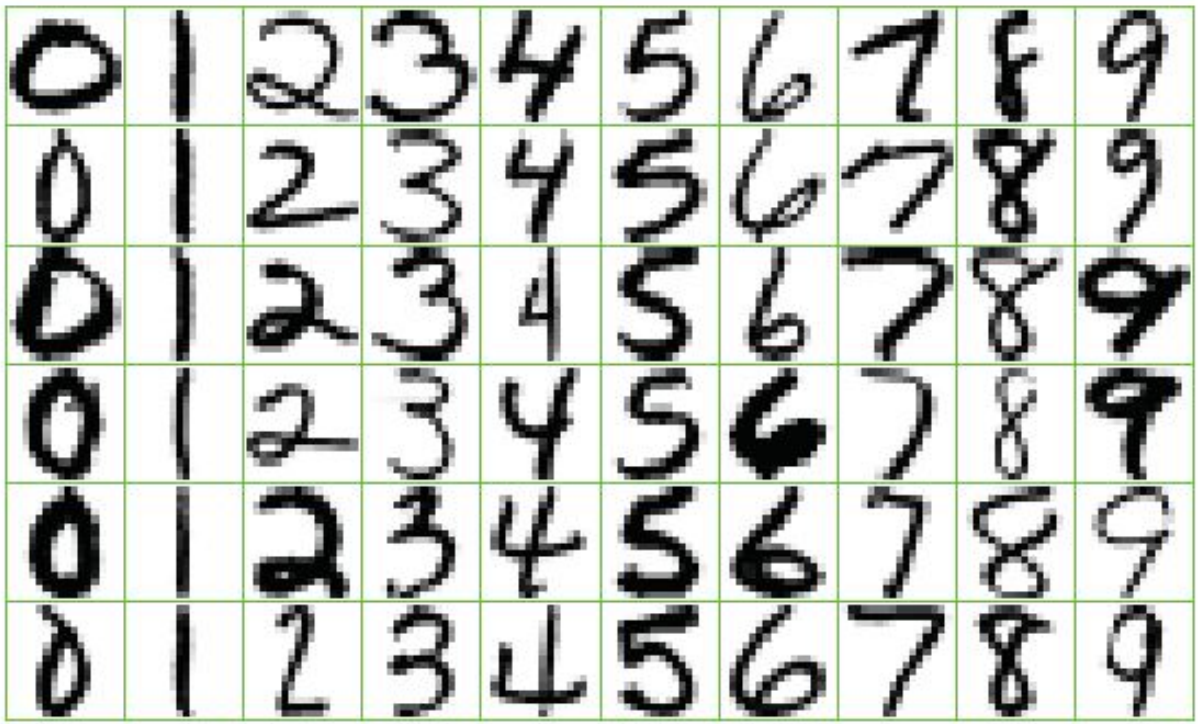
  where Σ_W = within-class covariance of the $h(x_i)$'s.
- Decompose the classification subspace using a penalized metric:

$$\max_u u^t \Sigma_B u \quad \text{subject to} \quad u^t (\Sigma_W + \lambda\Omega) u = 1$$

Note: Penalized Mahalanobis distance

- tends to give less weight to "rough" coordinates and more weight to "smooth" ones.
- the same applies to linear combinations that are rough or smooth if Ω not diagonal

Image Example: Classify Digits



Examples of handwritten digits from U.S. postal envelopes.

Set-up

- **Data:** $\{(g_i, I_i)\}, g_i \in \{0, 1, \ldots, 9\}$, I_i a 16×16 binary image.
  $\Longrightarrow$ have feature vectors $x_i \in [0,1]^{16^2}$ $\longleftarrow$ I_i's vectorized
- The Laplacian penalty functional

$$J(f) = \int_{\mathcal{R}} \left(\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \right)^2 dxdy$$

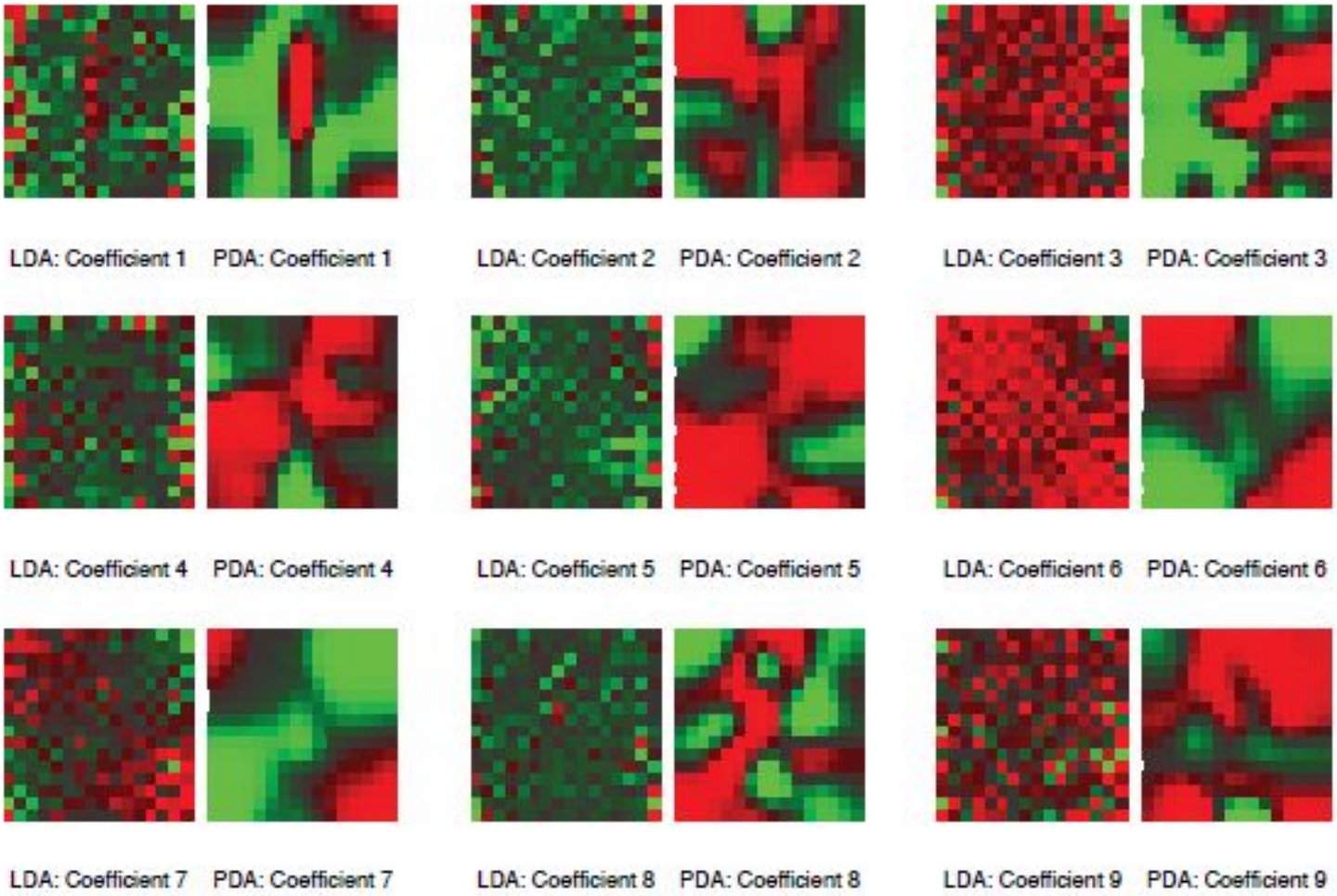
  measures smoothness of the function f over the region $\mathcal{R}$.
- Construct a discrete approximation for the coefficients β

$$J(\beta) = \beta^t \Delta^t \Delta \beta = \beta^t \Omega \beta$$

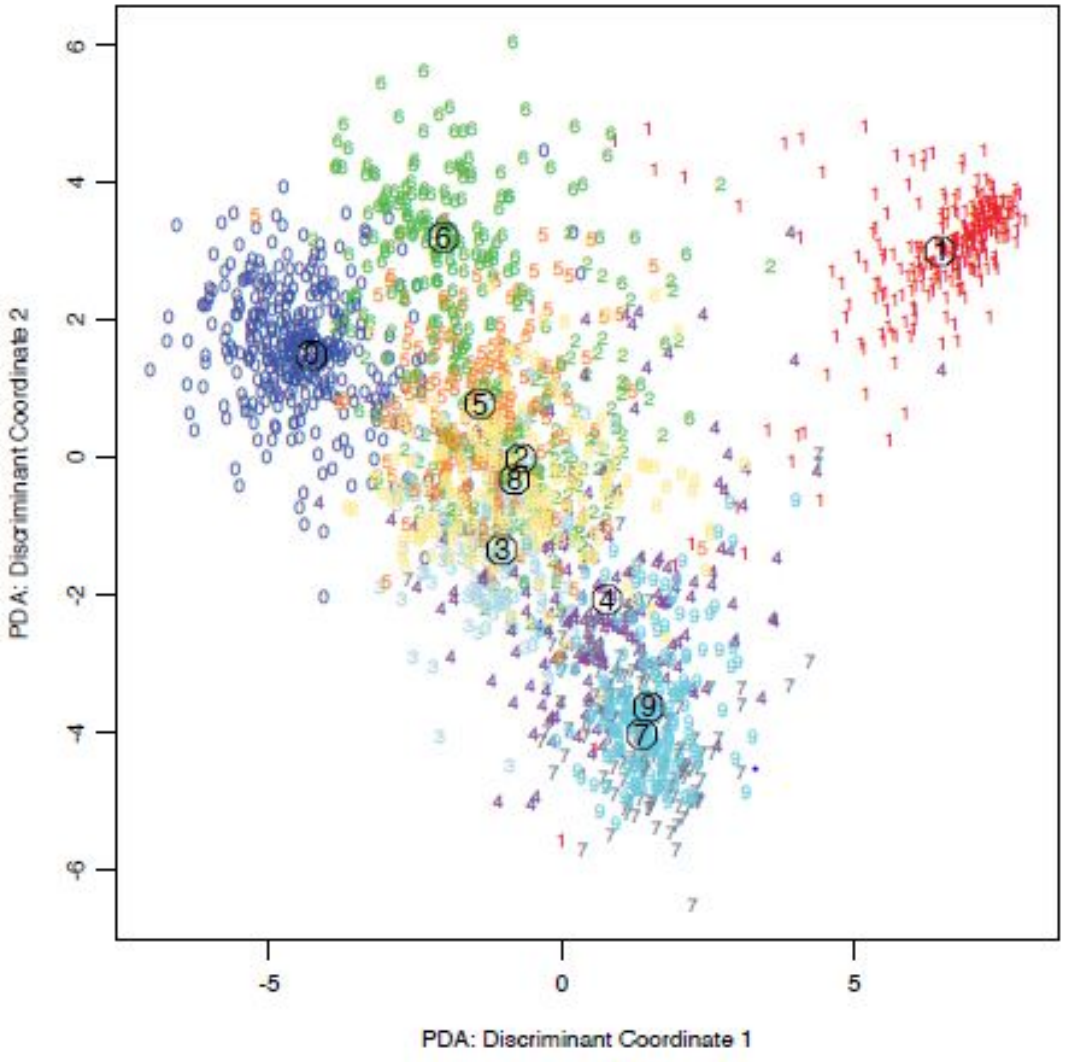
  where $\Delta = D \otimes E_{16} + E_{16} \otimes D$ -E_{16} is the 16×16 identity matrix- and

$$D = \begin{pmatrix} -1 & 1 & & & & & & & \\ & 1 & -2 & 1 & & & & & \\ & & 1 & -2 & 1 & & & & \\ & & & . & . & . & & & \\ & & & & 1 & -2 & 1 & & \\ & & & & & 1 & -2 & 1 & \\ & & & & & & & 1 & -1 \end{pmatrix}$$

Discriminant variates using LDA & PDA



First two penalized canonical variates of test points



Circles indicate the class centroids.

Mixture Discriminant Analysis

Overview: Mixture of Discriminant Analysis

- A method for classification (supervised) based on mixture models.
- Extension of LDA
- The mixture of normals is used to obtain a density estimation for each class

MDA & Gaussian Mixture Models

- GMM for the k-th class has density:

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

where $\sum_{r=1}^{R_k} \pi_{kr} = 1$

- Have R_k prototypes for each class and the same Σ for each component.

- Class posterior probabilities are given by:

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

where Π_k represents the class prior probabilities.

Estimation of Parameters

- Estimate the parameters by minimizing:

$$\sum_{k=1}^{K}\sum_{i\in I_k}\log\left\{\Pi_k\sum_{r=1}^{R_k}\pi_{kr}\phi(x_i;\mu_{kr},\Sigma)\right\},\quad I_k=\{i|g_i=k\}$$

  $\Longrightarrow$ maximize the joint-likelihood $P(G,X)$ of training data $\{(x_i,g_i)\}_{i=1}^n$.

- Direct optimization hard $\Longrightarrow$ use EM.

- Steps of EM
  - **E-step**: Given current estimate of parameters $\mu_{kr}^{(j)},\pi_{kr}^{(j)}$. Compute the responsibility for each point:

$$\omega_{kir}^{(j)}=\frac{\pi_{kr}^{(j)}\phi(x_i;\mu_{kr}^{(j)},\Sigma)}{\sum_{l=1}^{R_k}\pi_{lr}^{(j)}\phi(x_i;\mu_{lr}^{(j)},\Sigma)}\quad r=1,\ldots,R_k;i\in I_k;k=1,\ldots,K$$

  - **E-step**: Compute the weighted MLEs for all the parameters...

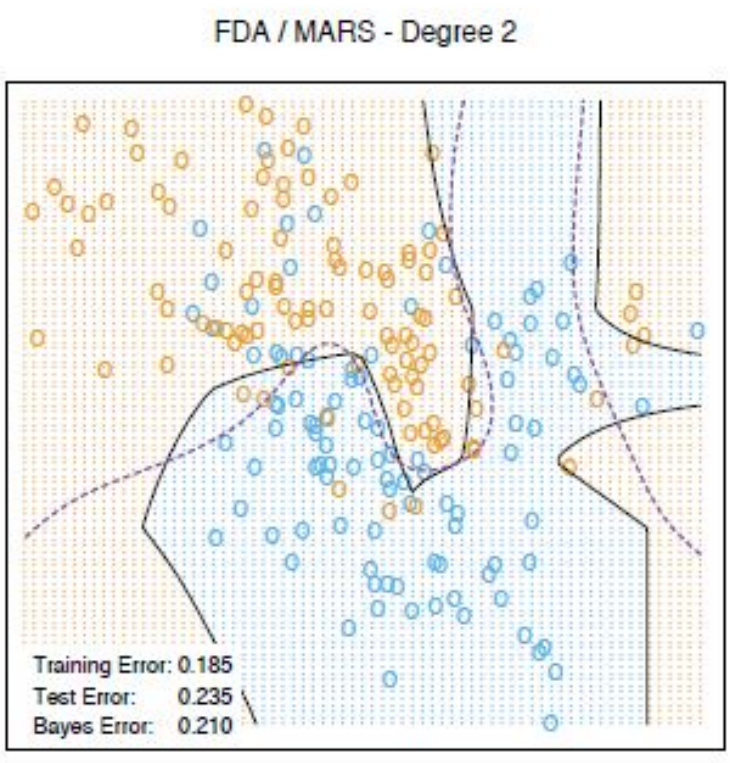
Estimation of Parameters

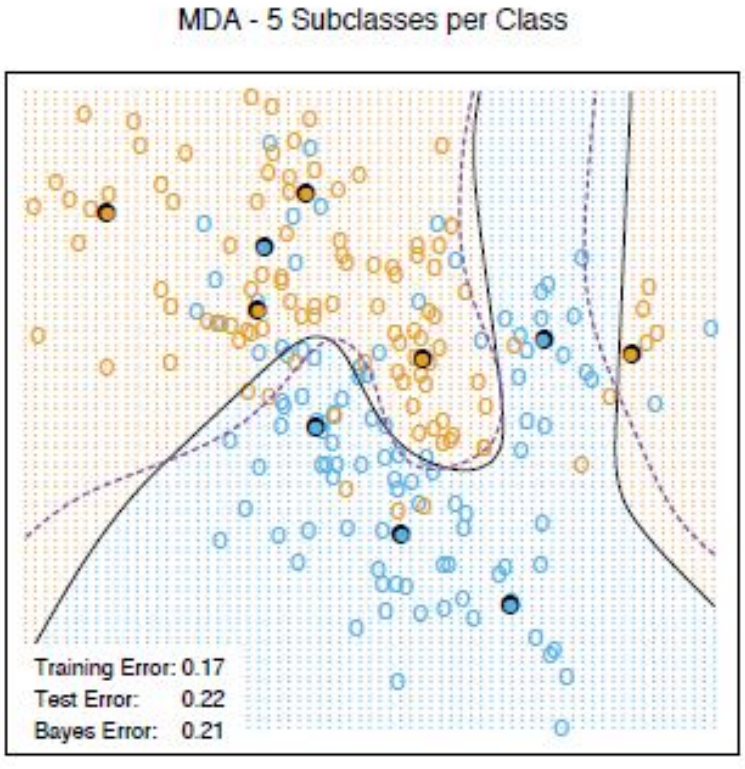
$$\mu_{kr}^{(j+1)} = \frac{\sum_{i \in I_k} \omega_{ikr}^{(j)} x_i}{\sum_{i \in I_k} \omega_{ikr}^{(j)}}, \quad \pi_{kr}^{(j+1)} = \frac{\sum_{i \in I_k} \omega_{ikr}^{(j)}}{|I_k|}$$

$$\Sigma = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in I_k} \sum_{r=1}^{R_k} \omega_{ikr}^{(j)} (x_i - \mu_{kr}^{(j)})(x_i - \mu_{kr}^{(j)})^t$$

- EM requires:
  - initialization of parameters and
  - setting the values R_k for $k = 1, \ldots, K$.

Example decision boundaries





Dashed purple line is the Bayes decision boundary.