STAT 535

11/21/23

Lecture 16

Happy Thanksgiving Weekend!

Non-linear SVM

GP

HW6 = last HW

Project ?

LVI ←

16 Kernels

17 NN as GP

18 Boost

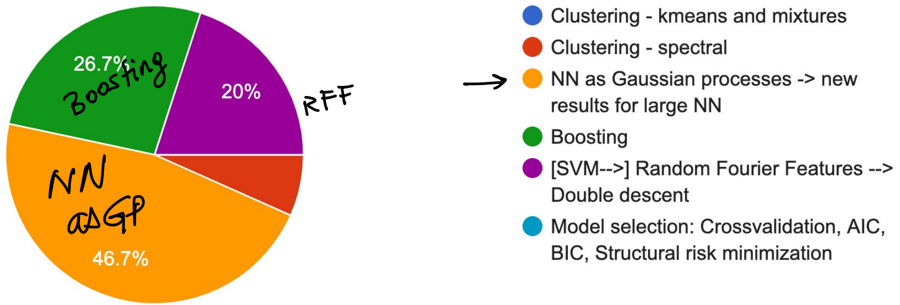
19

20 Clust ?

Project Results

First choice

15 responses



Second choice

15 responses

Lecture V: Support Vector Machines

Marina Meilă
mmp@stat.washington.edu

Department of Statistics
University of Washington

November, 2023

so far

Linear SVM's
- The margin and the expected classification error
- Maximum Margin Linear classifiers
- Linear classifiers for non-linearly separable data

• Lin SVM — linear D₀

$f(x) = w^T x + b$

- linearly separable $\mathcal{D}$
- non-linearly sep. $\mathcal{D}$

Non linear SVM

D₀ non-linear

- The "kernel trick"
- Kernels
- Prediction with SVM

Extensions
- L_1 SVM
- Multi-class and One class SVM
- SV Regression

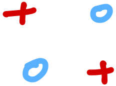
Reading HTF Ch.: Ch. 12.1–3, Murphy Ch.: Ch 14 (14.1,14.2–14.2.4 kernels, 14.4 and equations (14.28,14.29) kernel trick, 14.5.1.–3 Support Vector Machines), Bach Ch.: 7.1–7.4, 7.7

Additional Reading: C. Burges - "A tutorial on SVM for pattern recognition"

These notes: Appendices (convex optimization) are optional.

Non-linear SVM

How to use linear classifier on data that is not linearly separable?



An old trick

1. Map the data $x^{1:n}$ to a higher dimensional space

$$x \to z = \phi(x) \in \mathcal{H}, \text{with dim } \mathcal{H} >> n.$$

$f(x) = w^T x + b$

2. Construct a linear classifier $w^T z + b$ for the data in $\mathcal{H}$

$f(x) = w^T z_{(x)} + b$

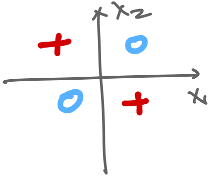
$= w^T \varphi(x) + b$

In other words, we are implementing the non-linear classifier

$$f(x) \;=\; w^T\phi(x) + b \;=\; w_1\phi_1(x) + w_2\phi_2(x) + \ldots + w_m\phi_m(x) + b \tag{31}$$

m features $\varphi_{1:m}(x)$ linear in features

Example



- Data $\{(x, y)\}$ below are not linearly separable

	x		y		z	
	-1	-1	1	-1	-1	1
+	-1	1	-1	-1	1	-1
+	1	-1	-1	1	-1	-1
	1	1	1	1	1	1
				φ_1	φ_2	φ_3

- We map them to 3 dimensions by

$$z = \phi(x) = [\underbrace{x_1}_{\varphi_1}\ x_2\ \underbrace{x_1 x_2}_{\varphi_3}].$$

- Now the classes can be separated by the hypeplane $z_3 = 0$ (which happens to be the maximum margin hyperplane). Hence,
  - $\boxed{w = [0\ 0\ 1]}$ (a vector in $\mathcal{H}$)
  - $b = 0$
  - and the classification rule is $f(\phi(x)) = w^T\phi(x) + b$.
- If we write f as a function of the original x we get

$$f(x) \;=\; x_1 x_2 \leftarrow$$

a quadratic classifier.

Large margin

ML : $\varphi \in \mathbb{R}^m$ m very large $\Leftarrow$

Bias-Var tradeoff ✓

Computation ?

>> Kernel Trick ✓

Non-linear SV problem $x \leftarrow \varphi(x)$

Computation in SV $\equiv$ Scalar product ($\equiv$ dot product)

- Primal problem minimize $\frac{1}{2}||w||^2$ s.t $y^i(\boxed{w^T\phi(x^i)} + b) - 1 \geq 0$ for all i.
  $\langle w, \varphi(x^i)\rangle$
- Dual problem

n :
$$\max_{\alpha_{1:n}} \sum_i \alpha_i - \frac{1}{2}\sum_i \alpha_i\alpha_j \underbrace{y^i y_j \phi(x^i)^T\phi(x_j)}_{\bar{G}_{ij}} \text{ s.t. } \alpha_i \geq 0 \text{ for all } i \text{ and } \sum_i y^i\alpha_i = 0 \quad (32)$$

$\langle \varphi(x^i), \varphi(x^j)\rangle$

NEW:
$$G_{ij} = \phi(x^i)^T\phi(x^j) \quad \text{and} \quad \bar{G} = \mathrm{diag}\,\{y^{1:n}\}^T G\, \mathrm{diag}\,\{y^{1:n}\}. \quad (33)$$

- $\bar{G}_{ij}$ has been redefined in terms of ϕ
- Dual problem

$$\max_{\alpha} 1^T\alpha - \frac{1}{2}\alpha^T\bar{G}\alpha \quad \text{s.t. } \alpha_i \geq 0,\ y^T\alpha = 0 \quad (34)$$

- Same as (19)!

(P) dim w = dim φ LARGE

(D) $G \in \mathbb{R}^{n\times n}$

$n \ll m$

$\langle x, x'\rangle$ = scalar product

Nonlinear SVM

1. choose φ ← so that $\varphi^T(x)\varphi(x')$ is fast to compute

2. compute G, $\bar{G}$

$$G_{ij} = \varphi(x^i)^T\varphi(x^j)$$

3. solve dual $\max_{\alpha \geq 0} \; 1^T\alpha - \frac{1}{2}\alpha^T\bar{G}\alpha$, s.t $y^T\alpha = 0$

$$\Rightarrow \alpha^*_{1:n}$$

Prediction $w = \sum_{i=1}^{n} \alpha_i^* y^i \varphi(x^i)$ $\in \mathbb{R}^m$

$$f(x) = w^T\varphi(x) + b = \left(\sum_{i=1}^{n}\alpha_i^* y^i \varphi(x^i)\right)^T \varphi(x) = \sum_{i=1}^{n} \alpha_i^* y^i \varphi(x^i)^T \varphi(x) + b$$

data

$\varphi : \mathcal{X} \rightarrow \mathcal{H}$

Feature Map

$\dim \mathcal{X} = d$

$\dim \mathcal{H} = m \gg d$ possibly ∞

The "Kernel Trick"

d idea The result (34) is the celebrated **kernel trick** of the SVM literature. We can make the following remarks.

1. The ϕ vectors enter the SVM optimization problem only trough the Gram matrix, thus only as the scalar products $\phi(x^i)^T\phi(x_j)$. We denote by $K(x, x')$ the function

$$K(x, x') = K(x', x) = \phi(x)^T\phi(x') \tag{35}$$

K is called the **kernel** function. If K can be computed efficiently, then the Gram matrix G can also be computed efficiently. This is exactly what one does in practice: we choose ϕ implicitly by choosing a kernel K. Hereby we also ensure that K can be computed efficiently.

2. Once G is obtained, the SVM optimization is independent of the dimension of x and of the dimension of $z = \phi(x)$. The complexity of the SVM optimization depends only on n the number of examples. This means that we can choose a very high dimensional ϕ without any penalty on the optimization cost.
3. Classifying a new point x. As we know, the SVM classification rule is

$$f(x) = w^T\phi(x) + b = \sum_{i=1}^{n} \alpha_i y^i \phi(x^i)^T\phi(x) = \sum_{i=1}^{n} \alpha_i y^i K(x^i, x) \tag{36}$$

Hence, the classification rule is expressed in terms of the support vectors and the kernel only. No operations other than scalar product are performed in the high dimensional space H.

Kernels

The previous section shows why SVMs are often called **kernel machines**. If we choose a kernel, we have all the benefits of a mapping in high dimensions, without ever carrying on any operations in that high dimensional space. The most usual kernel functions are

1) $K(x, x') = (1 + x^T x')^p$	the polynomial kernel of degree p	
2) $K(x, x') = \tanh(\sigma x^T x' - \beta)$	the "neural network" kernel	
3) $K(x, x') = e^{-\frac{\|\|x-x'\|\|^2}{\sigma^2}}$	the Gaussian or **radial basis function** (RBF) kernel	
smoothing	it's ϕ is ∞-dimensional $= m$	

0) $K(x, x') = x^T x'$ $\qquad \varphi(x) = x$

1) $K(x, x') = [x_1 \ x_2 \ x_1x_2] \begin{bmatrix} x_1' \\ x_2' \\ x_1'x_2' \end{bmatrix} \neq$

Slow $= 1 + 2x_1x_1' + 2x_2x_2' + 2x_1x_2x_1'x_2' + x_1^2x_1'^2 + x_2^2x_2'^2$ $\quad$ m multiplications, m-1 add

$\parallel$

Fast $= (1 + x^Tx')^2$ $\quad$ p=2 $\quad$ d mult+d additions +1

$\Longrightarrow$ See next page

$$(1 + x^T x')^2 = (1 + x_1 x_1' + x_2 x_2')^2$$

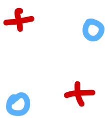
$$= 1 + x_1^2 x_1'^2 + x_2^2 x_2'^2 + 2 x_1 x_1' + 2 x_2 x_2' + 2 x_1 x_2 x_1' x_2'$$

$$= \underbrace{\begin{bmatrix} 1 & x_1^2 & x_2^2 & \sqrt{2} x_1 & \sqrt{2} x_2 & \sqrt{2} x_1 x_2 \end{bmatrix}}_{\varphi(x)} \underbrace{\begin{bmatrix} 1 \\ x_1'^2 \\ x_2'^2 \\ \sqrt{2} x_1' \\ \sqrt{2} x_2' \\ \sqrt{2} x_1' x_2' \end{bmatrix}}_{\varphi(x')}$$

$x \in \mathbb{R}^d$, $d = 2$

$\varphi(x) \in \mathbb{R}^m$ $\quad m = d(d+1)$

$K(x, x') \equiv \varphi^T(x) \varphi(x') \sim d$ operations



:

- more features than needed
- features scaled unequally (e.g. $\sqrt{2}$ factor)
- $\varphi(x)$ not explicitly computed

BUT can solve original problem
Same solution w padded with 0's

can model any quadratic decision boundary

$$k(x,x') = e^{-\frac{\|x-x'\|^2}{\cancel{2}}} = e^{-\frac{(x-x')^2}{\cancel{2}}}$$

$x, x' \in \mathbb{R}$

$$e^u = \sum_{k=0}^{\infty} \frac{u^k}{k!} = 1 + \frac{u}{1} + \frac{u^2}{2} + \frac{u^3}{3!} + \cdots = 1 + \frac{x^2 + x'^2 - 2xx'}{1} + \frac{(x^2 + x'^2 - 2xx')^2}{2} + \cdots$$

$$u = (x-x')^2 = x^2 + (x')^2 - 2xx'$$

$$= \underbrace{\begin{bmatrix} 1 & a_1 x & a_2 x^2 & a_3 x^3 & \cdots \end{bmatrix}}_{\varphi^T(x)} \begin{bmatrix} 1 \\ a_1 x' \\ a_2 (x')^2 \\ a_3 (x')^3 \\ \vdots \end{bmatrix}$$

$a_{1,2,3} \ldots$ = by identifying polynomial coefficients

i.e. k-th term = homogeneous polynomial of degree k

terms $1{:}k$ = polynomial of degree k

The Mercer condition

- How do we verify that a chosen K is is a valid kernel, i.e that there exists a ϕ so that $K(x, x') = \phi(x)^T\phi(x')$?
- This property is ensured by a positivity condition known as the Mercer condition.

Mercer condition

Let $(\mathcal{X}, \mu)$ be a finite measure space. A symmetric function $K : \mathcal{X} \times \mathcal{X}$, can be written in the form $K(x, x') = \phi(x)^T\phi(x')$ for some $\phi : \mathcal{X} \to \mathcal{H} \subset \mathbb{R}^m$ iff

$$\int_{\mathcal{X}^2} K(x, x')g(x)g(x')d\mu(x)d\mu(x') \geq 0 \quad \text{for all } g \text{ such that } ||g(x)||_{L_2} < \infty \tag{37}$$

$g^T G g \geq 0$
$\forall g$

- In other words, K must be a positive semidefinite operator on L_2.
- If K satisfies the Mercer condition, there is no guarantee that the corresponding ϕ is unique, or that it is finite-dimensional.

given K : $\exists \varphi : \mathcal{X} \to \mathcal{H}$
so that $\varphi(x^i)^T\varphi(x^j) = K(x, x')$

$G = [K(x^i, x^j)]_{i,j=1:n} \succeq 0$

Kernel machines

$x \in \mathcal{X}$ not necessarily $\mathbb{R}^d$!

text
string
tree
$\xrightarrow{\varphi}$ create features — directly e.g. neural network
— by kernels

distance in $\mathbb{R}^d$ (or $\mathbb{R}^m$) difference in coordinates

$$\|x - x'\|^2 = (x_1 - x_1')^2 + \dots$$

Kernel simpler kernels

Kernels and Euclidean distances are close relatives

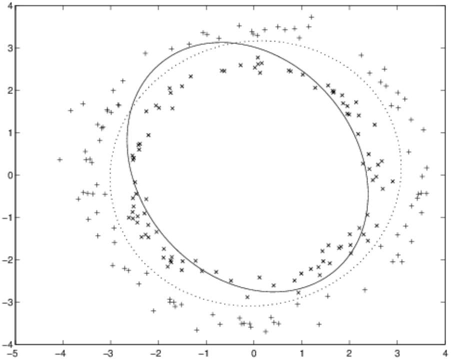
$$\|x - x'\|^2 = \langle x - x', x - x' \rangle = \langle x, x \rangle + \langle x', x' \rangle - 2\langle x, x' \rangle$$

$$= \|x\|^2 + \|x'\|^2 - 2x^T x$$

Given $\|x\|, \|x'\|, \|x - x'\| \Rightarrow \langle x, x' \rangle$

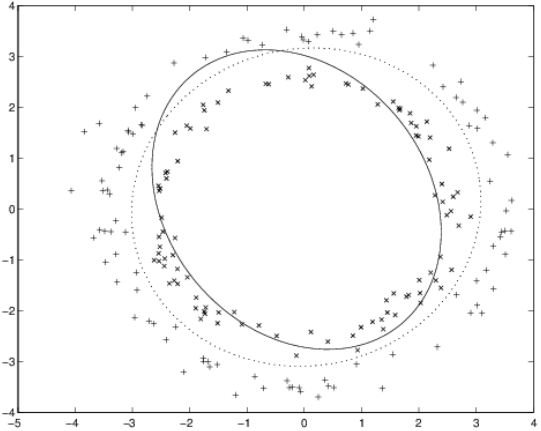
Given kernel $k(\,,\,) \equiv \langle\,,\,\rangle \Rightarrow \|x - x'\|$

Quadratic kernel



- C-SVM, polynomial degree 2 kernel, $n = 200$, $C = 10000$
- The two ellipses show that a constant shift to the data ($x^i \leftarrow x^i + v$, $v \in \mathbb{R}^n$) can affect non-linear kernel classifiers.

RBF kernel and Support Vectors



Prediction with SVM

- Estimating b
  - For any i support vector, $w^T x^i + b = y^i$ because the classification is tight
  - Alternatively, if there are slack variables, $w^T x^i + b = y^i(1 - \xi_i)$
  - Hence, $b = y^i(1 - \xi_i) - w^T x^i$
  - For non-linear SVM, where w is not known explicitly, $w = \sum_j \alpha_j y^j \phi(x_j)$. Hence, $b = y^i(1 - \xi_i) - \sum_{j=1}^n \alpha_j y^j K(x^i, x^j)$ for any i support vector
- Given new x

$$\hat{y}(x) \;=\; \mathrm{sgn}(w^T x + b) \;=\; \mathrm{sgn}\left(\sum_{i=1}^n \alpha_i y^i K(x^i, x) + b\right). \tag{38}$$

L1-SVM

- If the regularization $||w||^2$, based on l_2 norm, is replaced with the l_1 norm $||w||_1$, we obtain what is known as the **Linear L1-SVM**

$$\min_{w,b} ||w||_1 + C\sum_i \xi_i \quad \text{s.t}\; y^i(w^T x^i + b) \geq 1 - \xi_i,\; \xi_i \geq 0 \text{ for all } i = 1 : n \tag{39}$$

- The use of the l_1 norm promotes sparsity in the entries of w

- The **Non-linear L1-SVM** is

$$f(x) = \sum_i (\alpha_i^+ + \alpha_i^-) y^i K(x_i, x) + b \quad \text{classifier} \tag{40}$$

$$\min_{\alpha_\pm, b} \sum_i (\alpha_i^+ + \alpha_i^-) + C\sum_i \xi_i \quad \text{s.t}\; y^i f(x^i) \geq 1 - \xi_i,\; \xi_i, \alpha_i^\pm \geq 0 \text{ for all } i = 1 : n \tag{41}$$

- This formulation enforces $\alpha_i^+ = 0$ or $\alpha_i^- = 0$ for all i. If we set $w_i = \alpha_i^+ - \alpha_i^-$, we can write $f(x) = \sum_i w_i y^i K(x^i, x) + b$, a linear classifier in the non-linear features $K(x^i, x)$.
- The L1-SVM problems are Linear Programs
- The dual L1-SVM problems are also linear programs
- The L1-SVM is no longer a Maximum Margin classifier

Multi-class and One class SVM

Multiclass SVM

For a problem with K possible classes, we construct K separating hyperplanes $w_r^T x + b_r = 0$.

$$\text{minimize} \quad \frac{1}{2}\sum_{r=1}^{K} ||w_r||^2 + \frac{C}{n}\sum_{i,r} \xi_{i,r} \tag{42}$$

$$\text{s.t.} \quad w_{y^i}^T x^i + b_{y^i} \geq w_r^T x^i + b_r + 1 - \xi_{i,r} \quad \text{for all } i = 1:n,\ r \neq y^i \tag{43}$$

$$\xi_{i,r} \geq 0 \tag{44}$$

One-class SVM This SVM finds the "support regions" of the data, by separating the data from the origin by a hyperplane. It's mostly used with the Gaussian kernel, that projects the data on the unit sphere. The formulation below is identical to the ν-SVM where all points have label 1.

$$\text{minimize} \quad \frac{1}{2}||w||^2 - \nu\rho + \frac{1}{n}\sum_i \xi_i \tag{45}$$

$$\text{s.t.} \quad w^T x^i + b \geq \rho - \xi_i \tag{46}$$

$$\xi_i \geq 0 \tag{47}$$

$$\rho \geq 0 \tag{48}$$

SV Regression

The idea is to construct a "tolerance interval" of $\pm\epsilon$ around the regressor f and to penalize data points for being outside this tolerance margin. In words, we try to construct the smoothest function that goes within ϵ of the data points.

$$\begin{aligned}
\text{minimize} \quad & \frac{1}{2}||w||^2 + C\sum_i(\xi_i^+ + \xi_i^-) && (49)\\
\text{s.t.} \quad & \epsilon + \xi_i^+ \;\geq\; w^T x^i + b - y^i \;\geq\; -\epsilon - \xi_i^- && (50)\\
& \xi_i^\pm \;\geq\; 0 && (51)\\
& \rho \;\geq\; 0 && (52)
\end{aligned}$$

The above problem is a linear regression, but with the kernel trick we obtain a kernel regressor of the form $f(x) = \sum_i(\alpha_i^- - \alpha_i^+)K(x^i, x) + b$

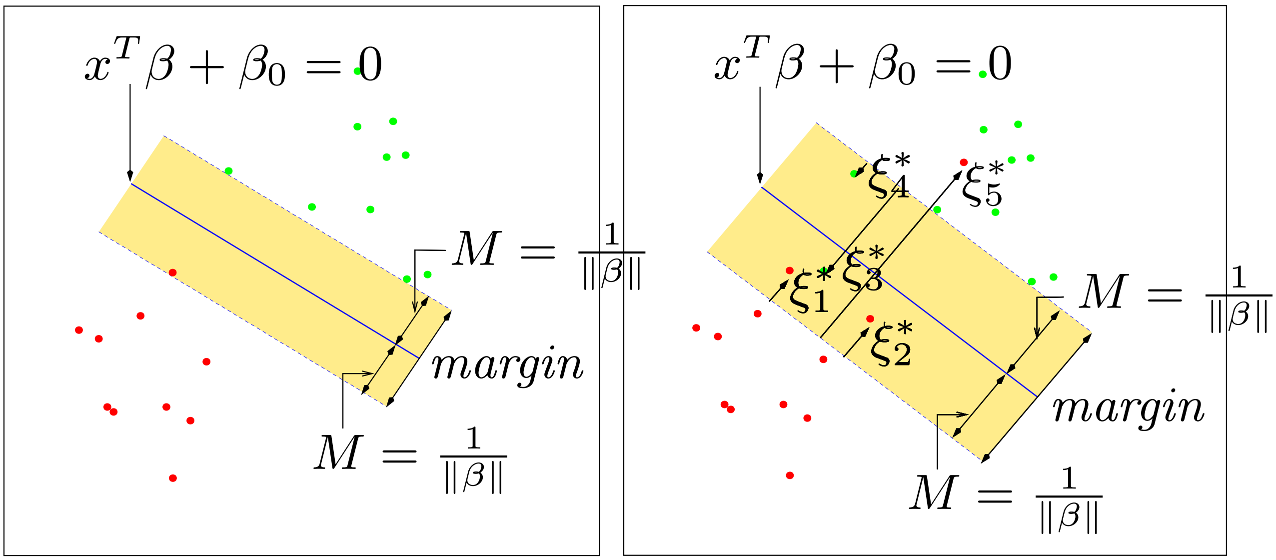
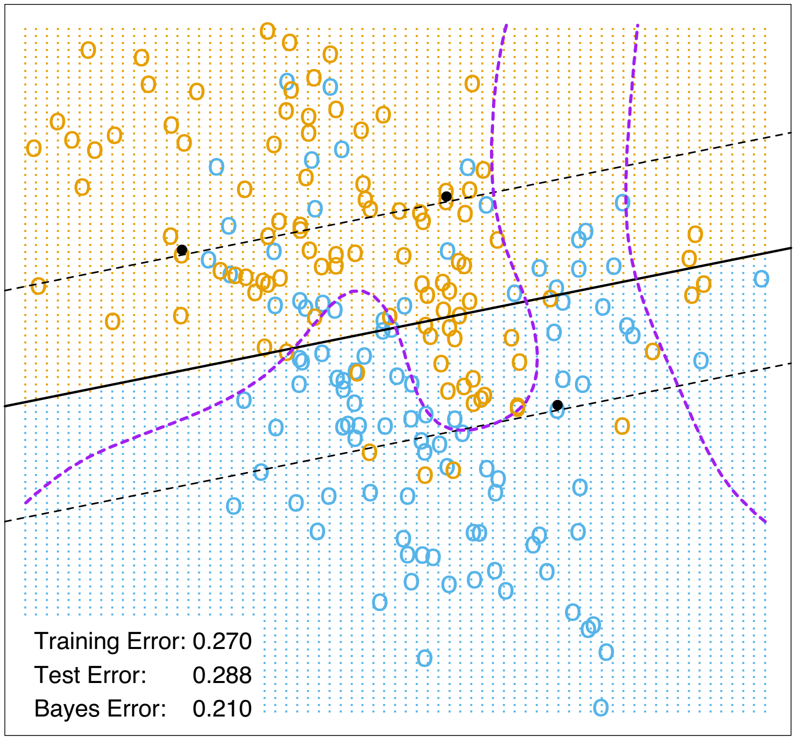
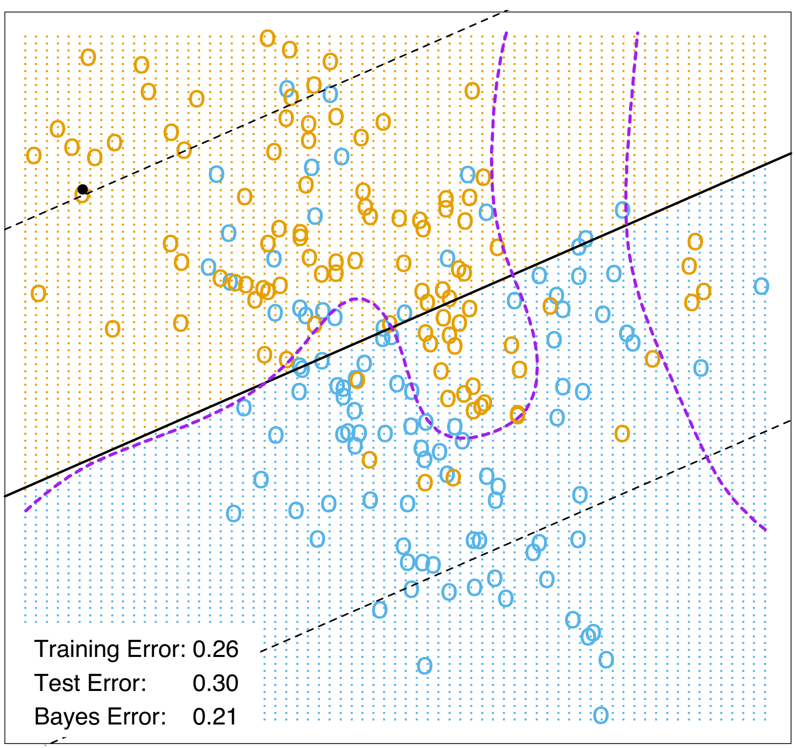


FIGURE 12.1. *Support vector classifiers. The left panel shows the separable case. The decision boundary is the solid line, while broken lines bound the shaded maximal margin of width* $2M = 2/\|\beta\|$*. The right panel shows the nonseparable (overlap) case. The points labeled* ξ_j^* *are on the wrong 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.*



$C = 10000$



$C = 0.01$

FIGURE 12.2. *The linear support vector boundary for the mixture data example with two overlapping*

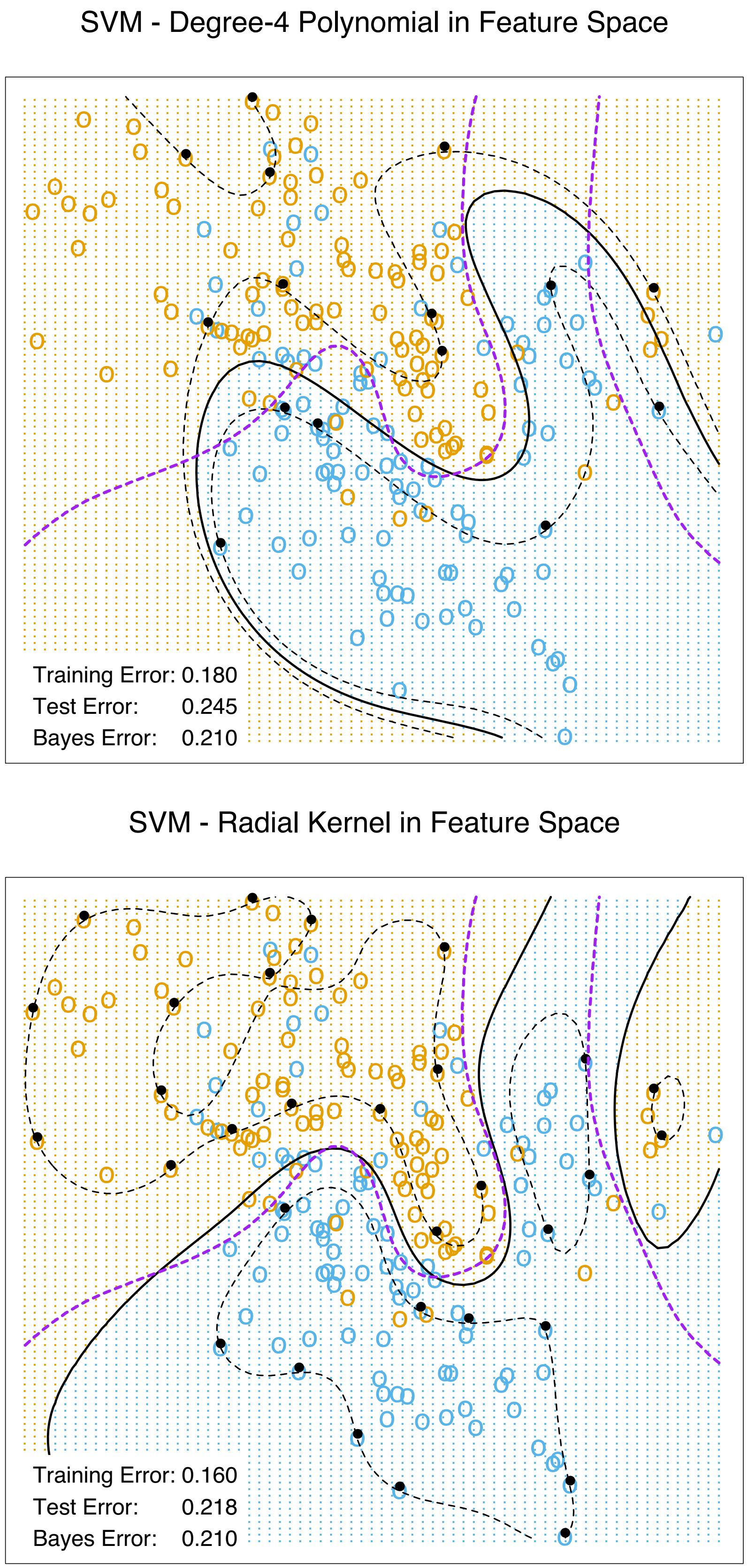


FIGURE 12.3. *Two nonlinear SVMs for the mixture data. The upper plot uses a 4th degree polynomial*

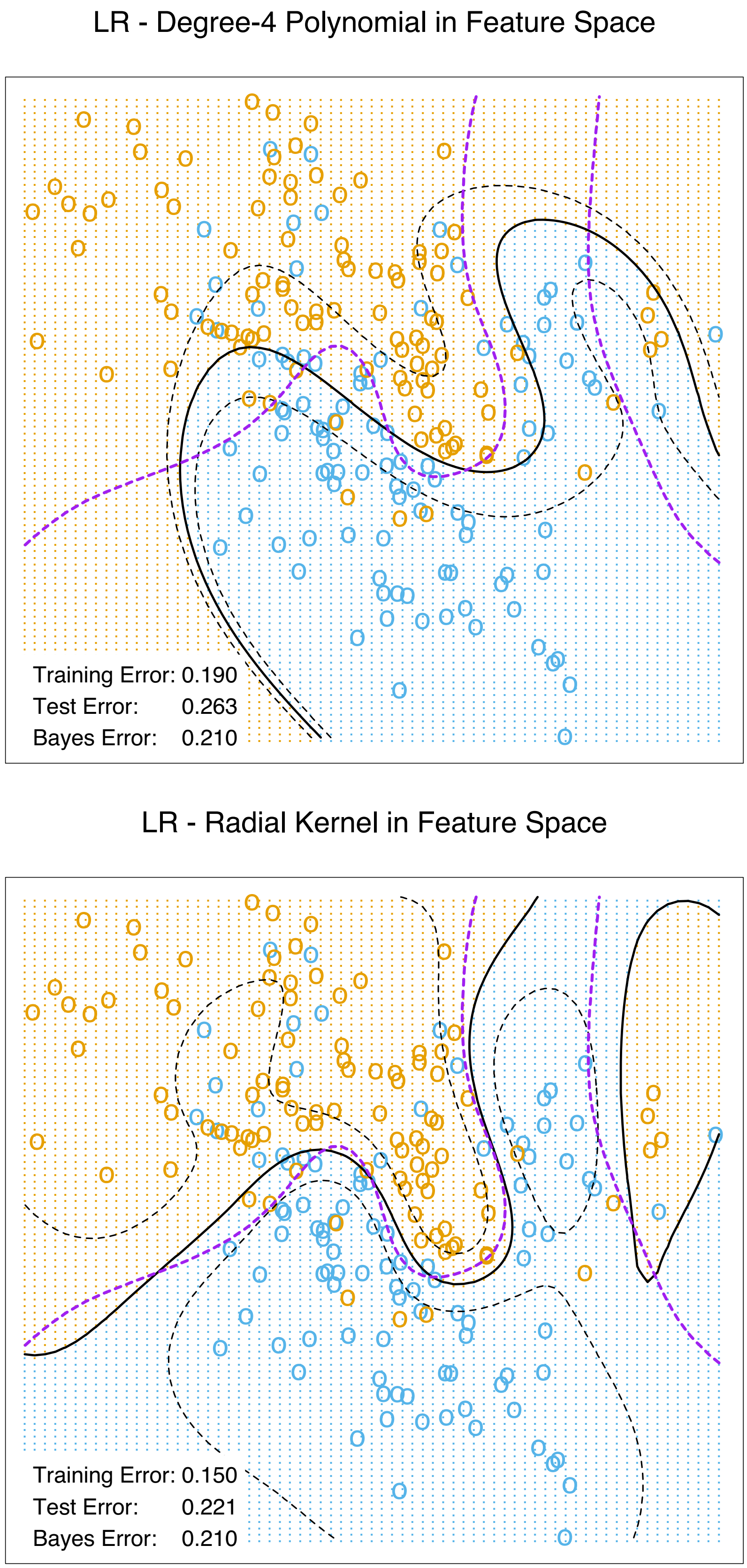


FIGURE 12.5. *The logistic regression versions of the SVM models in Figure 12.3, using the identical kernels*

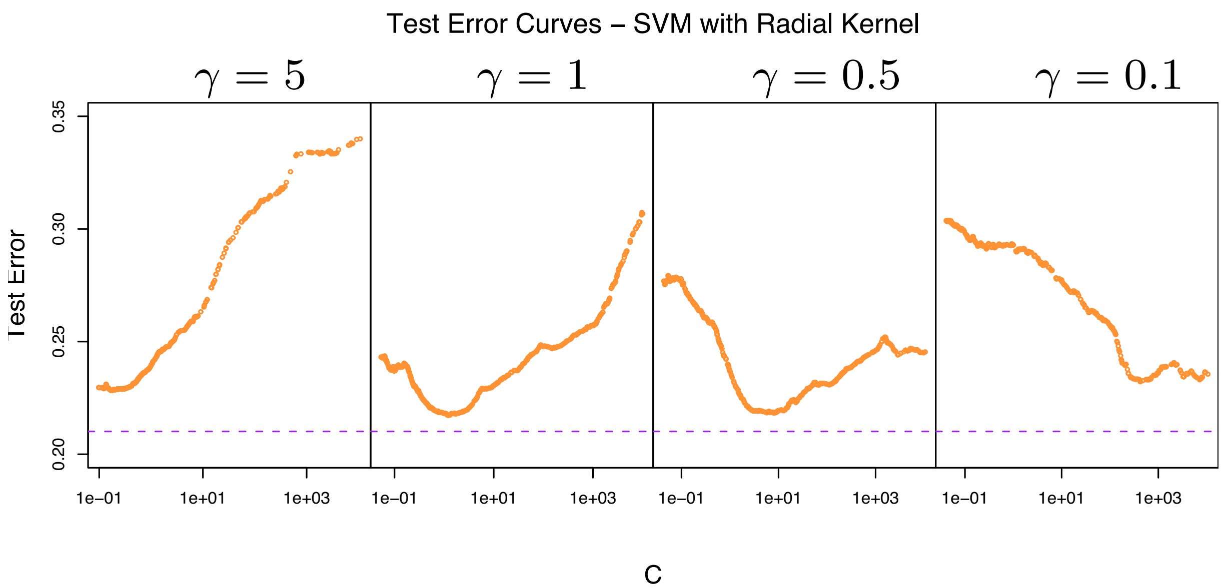


FIGURE 12.6. *Test-error curves as a function of the cost parameter C for the radial-kernel SVM classifier on the mixture data. At the top of each plot is the scale parameter γ for the radial kernel: $K_\gamma(x, y) = \exp -\gamma||x - y||^2$. The optimal value for C depends quite strongly on the scale of the kernel. The Bayes error rate is indicated by the broken horizontal lines.*