BIO STAT 527

5/3/23

Lecture 12

C - SVM $\rightarrow$ selecting C

Non-linear SVM

- Kernel trick
- Nonlinear SVM prediction
- RBF - effect of σ

Project

submit results: clustering alg1.txt, alg2.txt

5/23 $\leftrightarrow$ ≈ 10% - 15%

Report: exam week ≈ 90% - 85%

Lecture V: Support Vector Machines and Kernel Machines

Marina Meilă
mmp@stat.washington.edu

Department of Statistics
University of Washington

April, 2023

Linear SVM's

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

Non linear SVM

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

Extensions

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

Reading AoNPS Ch.: Ch. 12.1–3, HTF 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)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-linearly separable problems and their duals

regularization

The **C-SVM**

$$\text{minimize}_{w,b,\xi} \quad \frac{1}{2}||w||^2 + C\sum_i \xi_i \tag{20}$$

$$\text{s.t.} \quad y^i(w^T x^i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

slack

In the above, ξ_i are the slack variables. Dual[3]:

$1^T\alpha$ $\alpha^T \tilde{G} \alpha$

$$\text{maximize}_\alpha \quad \boxed{\sum_i \alpha_i} - \frac{1}{2}\boxed{\sum_i \alpha_i\alpha_j \boxed{y^i y_j x^{iT} x_j}} \tag{21}$$

$\tilde{G}_{ij}$

$$\text{s.t.} \quad \boxed{C \geq} \alpha_i \geq 0 \text{ for all } i$$

$$\sum_i \alpha_i y^i = 0$$

$\Rightarrow$ two types of SV

- $\alpha_i < C$ data point x^i is "on the margin" $\Leftrightarrow$ $y^i(w^T x^i + b) = 1$ (original SV)
- $\alpha_i = C$ data point x^i cannot be classified with margin 1 (**margin error**) $\Leftrightarrow$ $y^i(w^T x^i + b) < 1$

[3] Lagrangean $L(w, b, \xi, \alpha, \mu) = \frac{1}{2}||w||^2 + C\sum_i \xi_i - \sum_i \alpha_i[y^i(w^T x^i + b) - 1 + \xi_i] - \sum_i \mu_i\xi_i$ with $\alpha_i \geq 0, \; \xi_i \geq 0, \; \mu_i \geq 0$

The margin and the expected classification error

Theorem Let $\mathcal{F} = \{\mathrm{sgn}\,(w^T x),\ ||w|| \le \Lambda,\ ||x|| \le R\}$ and let $\rho > 0$ be any "margin". Then for any $f \in \mathcal{F}$, w.p $1 - \delta$ over training sets

gen. error

$$\Pr[err] \equiv L_{01}(f) \ \le\ \underbrace{\hat{L}_\rho + \sqrt{\frac{c}{n}\left(\frac{R^2\Lambda^2}{\rho^2}\ln n^2 + \ln\frac{1}{\delta}\right)}}_{r.h.s(5)} \qquad (5)$$

$\leftarrow$ find ρ that minimizes

where c is a universal constant and $\hat{L}_\rho$ is the fraction of the training examples for which

$$y^i w^T x_i < \rho \qquad (6)$$

- a data point i that satisfies (6) for some ρ is called a **margin error**
- For $\rho = 0$ the margin error rate $\hat{L}_\rho$ is equal to $\hat{L}_{01}$

Choosing C

- $C \nearrow \Rightarrow \sum_{i=1}^{n} \xi_i \searrow$ penalized more $\Rightarrow$ fewer margin errors / smaller

  $\Rightarrow \|W\| \nearrow \Leftrightarrow$ margin $\searrow$ ρ

- try C $\Rightarrow$ for all ρ $\hat{L}_\rho = \frac{\#\text{ margin errors }(\rho)}{n}$ $\rightarrow \min_\rho$ r.h.s (5) $\equiv$ Bound (C)

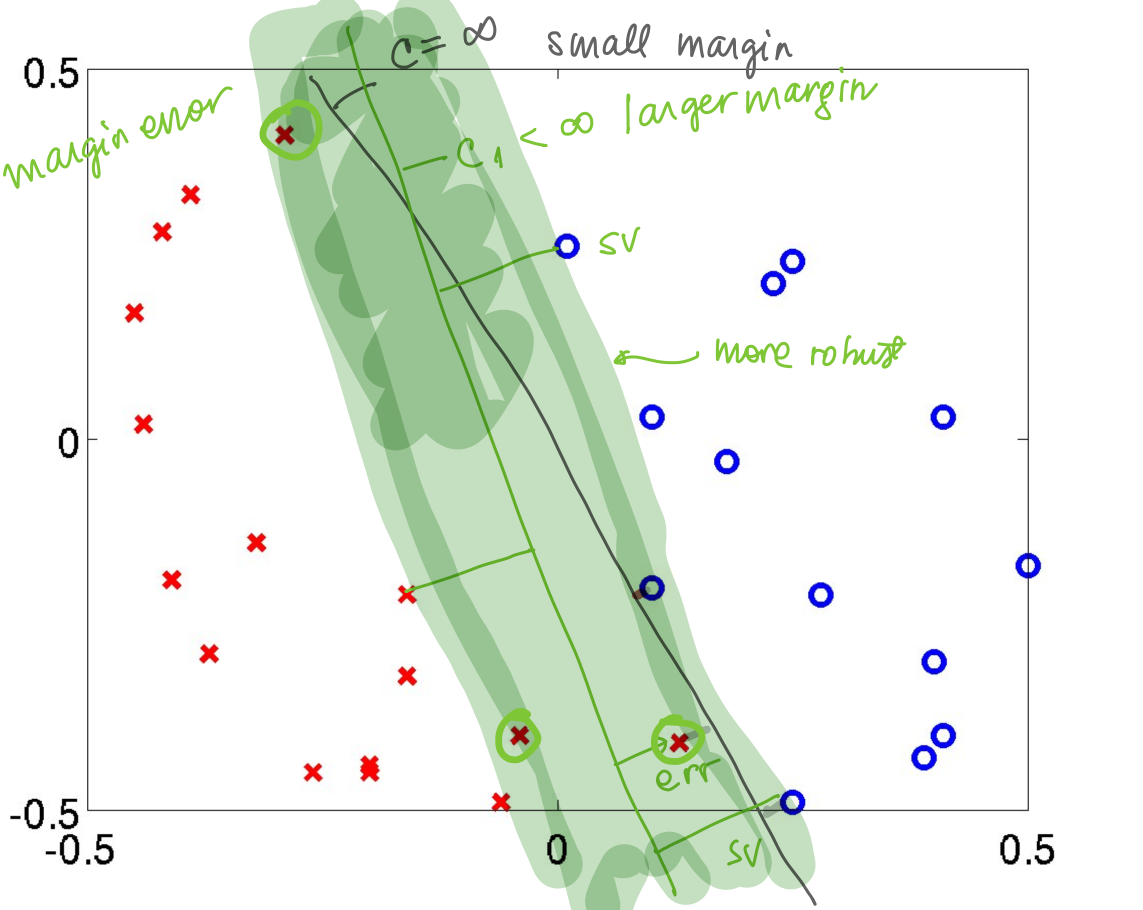
  (for P[err])

  compute SVM(C)

  $\min_C$ Bound (C)

- ν - SVM $\quad \nu = \frac{1}{C}$ (heuristic)

- Cross Validation


C = ∞ small margin
$C_1 < \infty$ larger margin
margin error
sv
more robust
err
sv
0.5
0
-0.5
-0.5
0
0.5

The ν-SVM

regularization

$$\begin{aligned} \text{minimize}_{w,b,\xi,\rho} \quad & \frac{1}{2}||w||^2 - \nu\rho + \frac{1}{n}\sum_i \xi_i && (22)\\ \text{s.t.} \quad & y^i(w^T x^i + b) \geq \rho - \xi_i && (23)\\ & \xi_i \geq 0 && (24)\\ & \rho \geq 0 && (25) \end{aligned}$$

where $\nu \in [0,1]$ is a parameter.
Dual[4]:

$$\begin{aligned} \text{maximize}_{\alpha} \quad & -\frac{1}{2}\sum_i \alpha_i\alpha_j y^i y^j x^{iT} x^j && (26)\\ \text{s.t.} \quad & \frac{1}{n} \geq \alpha_i \geq 0 \text{ for all } i && (27)\\ & \sum_i \alpha_i y^i = 0 && (28)\\ & \sum_i \alpha_i \geq \nu && (29) \end{aligned}$$

$\nu \approx 10\%$ n small
$\approx 10^{-3}$ n large

Properties If $\rho > 0$ then:

- ν is an upper bound on #margin errors/n (if $\sum_i \alpha_i = \nu$)
- ν is a lower bound on #(original support vectors + margin errors)/n
- ν-SVM leads to the same w, b as C-SVM with $\boxed{C = 1/\nu}$

[4] Lagrangean $L(w,b,\xi,\rho,\alpha,\mu,\delta) = \frac{1}{2}||w||^2 - \nu\rho + \frac{1}{n}\sum_i \xi_i - \sum_i \alpha_i[y^i(w^T x^i + b) - \rho + \xi_i] - \sum_i \mu_i\xi_i - \delta\rho$ with $\alpha_i \geq 0,\ \delta \geq 0,\ \mu_i \geq 0$

A simple error bound

$$L_{01}(f_n) \;\leq\; E\left[\frac{\#\text{support vectors of } f_{n+1}}{n+1}\right] \tag{30}$$

where f_n denotes the SVM trained on a sample of size n.
Exercise Use the Homework 6 to prove this result.

Non-linear SVM

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

An old trick

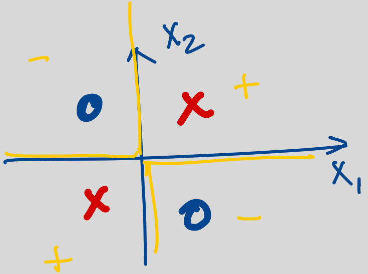
1. Map the data $x^{1:n}$ to a higher dimensional space

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

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

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

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



$$(x_1, x_2) \longrightarrow z = (x_1, x_2, x_1x_2)$$

$$f(z) = z_3 \equiv x_1x_2$$

linear in z

non-linear

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

- We map them to 3 dimensions by

$$z = \phi(x) = [x_1 \; x_2 \; x_1 x_2].$$

- Now the classes can be separated by the hypeplane $z_3 = 0$ (which happens to be the maximum margin hyperplane). Hence,
  - $w = [0 \; 0 \; 1]$ (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

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

a quadratic classifier.

Non-linear SV problem

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

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

$$G_{ij} = \boxed{\phi(x^i)^T \phi(x^j)} \text{ and } \bar{G} = \operatorname{diag}\{y^{1:n}\}^T G \operatorname{diag}\{y^{1:n}\}. \tag{33}$$

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

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

- Same as (19)!

$x \in \mathbb{R}^d$

want to add <u>all</u> interactions $\Rightarrow \varphi(x) \equiv z \in \mathbb{R}^{d + \frac{d(d+1)}{2}}$

d large $\Rightarrow$ $\varphi(x)$ intractable !!

Example

$$(x_1, x_2) \longmapsto (\sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2, 1) = z$$

feature map

$$z^T z' = 2x_1x_1' + 2x_2x_2' + 2x_1x_2x_1'x_2' + x_1^2x_1'^2 + x_2^2x_2'^2 + 1 \quad \sim d^2$$

$$= (1 + x_1x_1' + x_2x_2')^2 \quad \sim d$$

kernel trick

The "Kernel Trick"

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

faster than (dim φ)

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

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

$\varphi = ?$

(Taylor expansion)

When is $K(,)$ a kernel?

when $\exists \varphi$ such that $\varphi(x)^T \varphi(x') = K(x, x')$

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

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

Kernel SVM summary

- Data $\mathcal{D} = \{(x^{1:n}, y^{1:n})\}$
- Choose kernel K $\Rightarrow$ $G = [K(x^i, x^j)]_{i,j=1:n}$

$$\tilde{G} = [y^i y^j G_{ij}]_{i,j=1:n}$$

- Choose C
- Solve (D) $\max_{\alpha} \; 1^T\alpha - \frac{1}{2}\alpha^T \tilde{G} \alpha$ s.t $\sum_{i=1}^{n} \alpha_i y^i = 0$, $0 \leq \alpha_i \leq C$

$n \times n$

- solution $\alpha_{1:n}$ $\Rightarrow$ $w = \sum_{i=1}^{n} \alpha_i \varphi(x^i) y^i$, $f(x) = \sum_{i=1}^{n} \alpha_i y^i \varphi(x^i)^T \varphi(x) + b$

$\Downarrow$

compute b as before

$$\sum_{i=1}^{n} \alpha_i y^i K(x^i, x^{i_0}) + b = y^{i_0}$$

$i_0 = $ s.v.

$\Downarrow$

solve for b

- Classify new x

$$f_{w,b}(x) = \sum_{i=1}^{n} \alpha_i y^i K(x^i, x) + b$$

never represented

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.

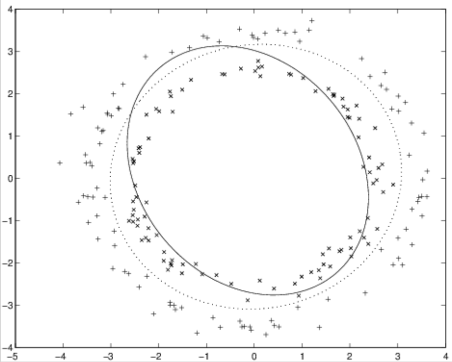
Mercer condition

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

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

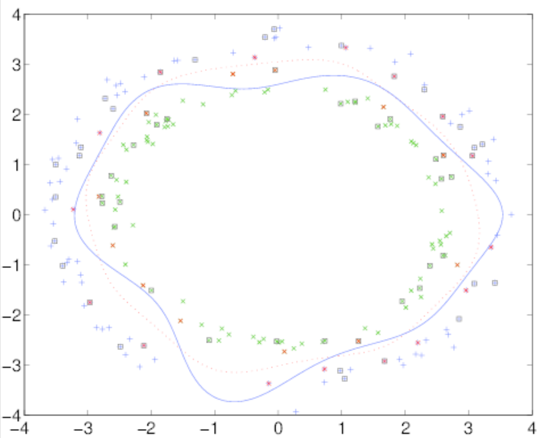
- In other words, K must be a positive semidefinite operator on L_2.
- If K satisfies the Mercer condition, there is no guarantee that the corresponding ϕ is unique, or that it is finite-dimensional.

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



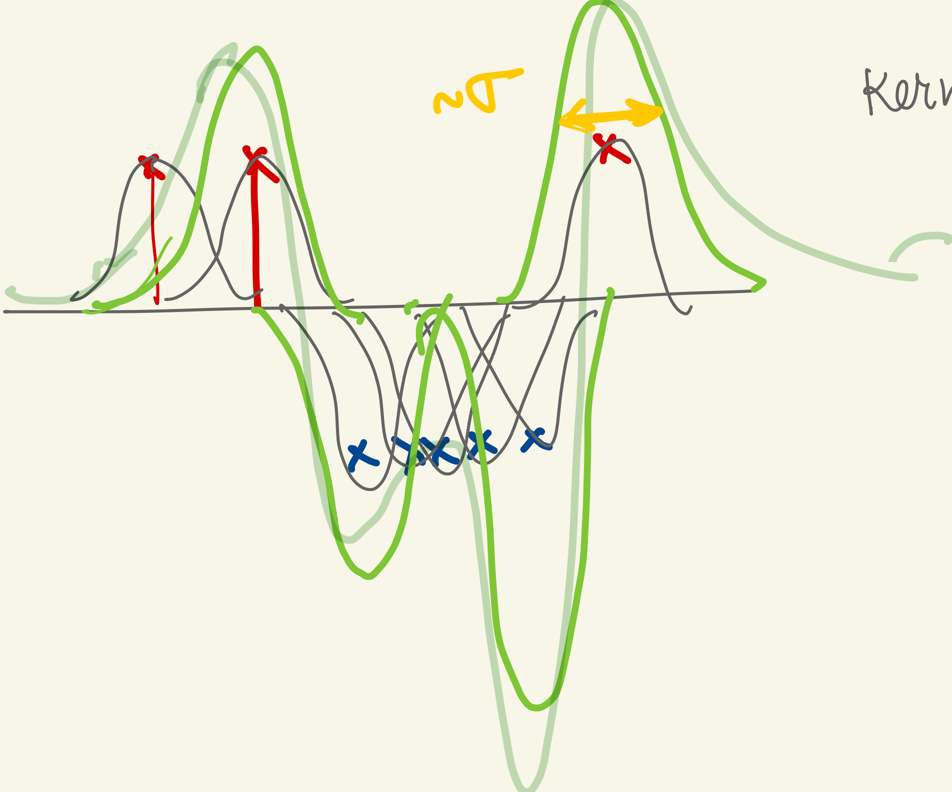
RBF Kernel

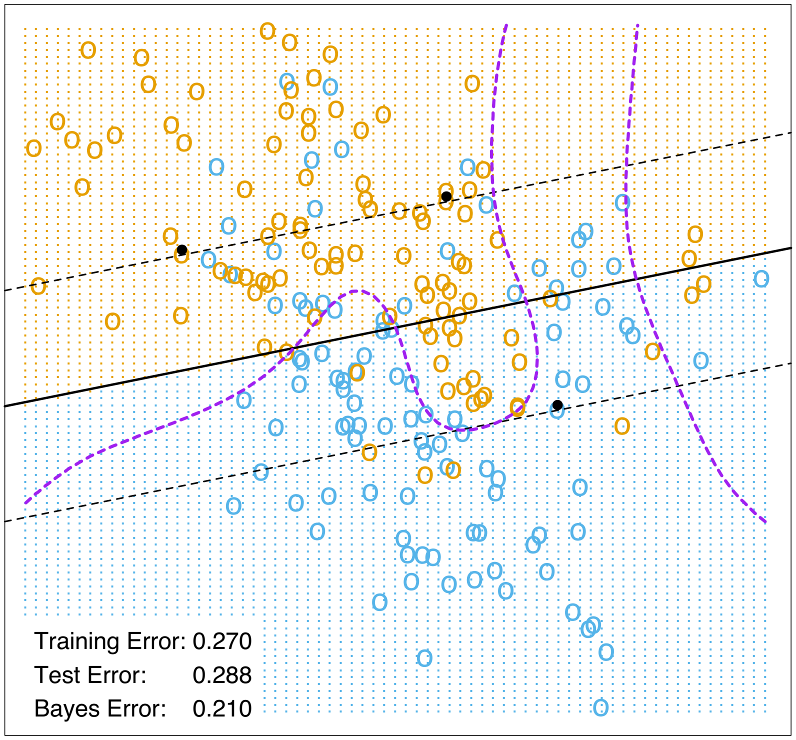
$$K(x,x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$$

$$f(x) = \sum \alpha_i K(x^i, x) y_i + b$$

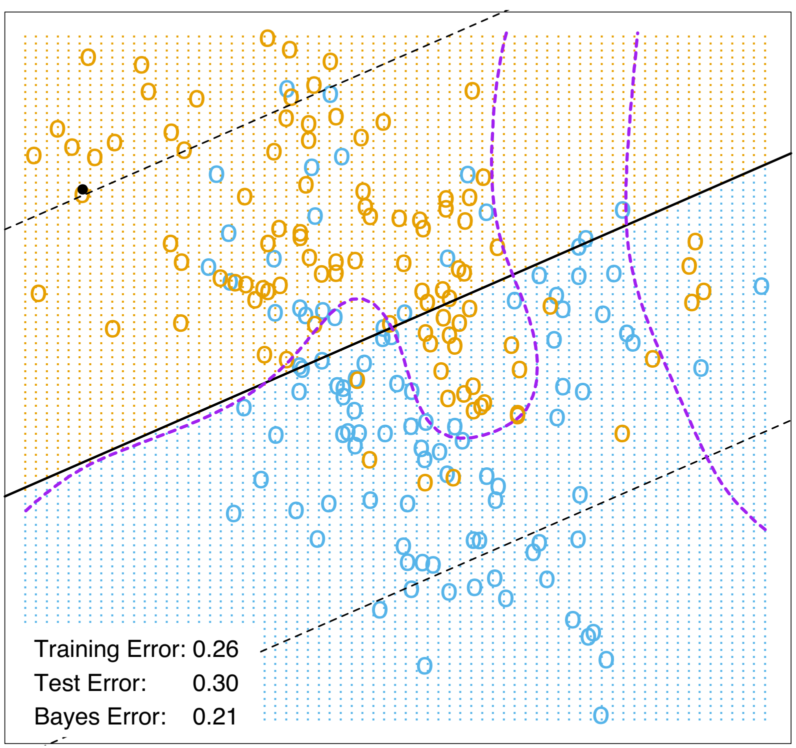
RBF

Kernel classification





$C = 10000$



$C = 0.01$

FIGURE 12.2. *The linear support vector boundary for the mixture data example with two overlapping classes, for two different values of C. The broken lines*

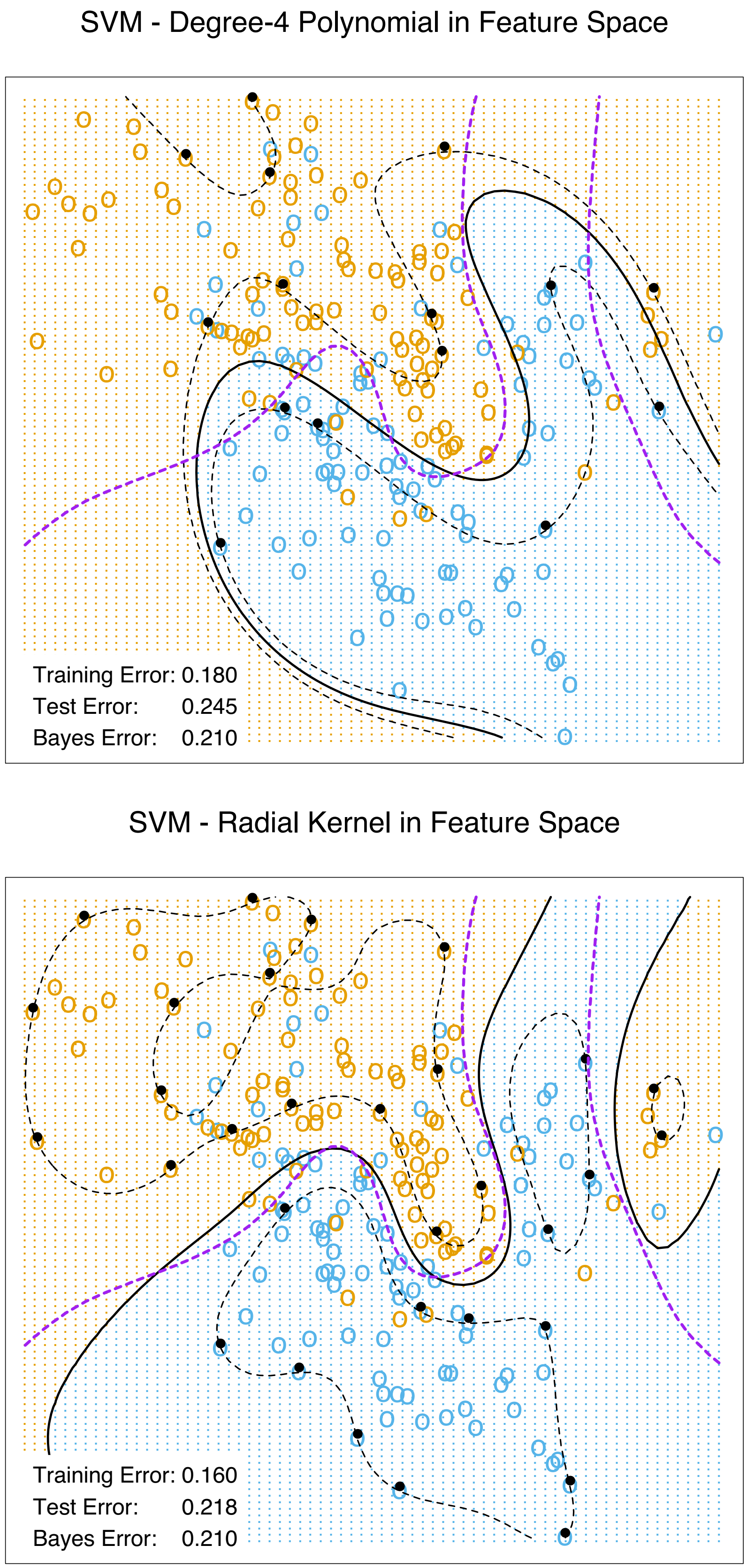


FIGURE 12.3. *Two nonlinear SVMs for the mixture data. The upper plot uses a 4th degree polynomial*

RKHS

spaces of functions $\ni f$ classifier

infinite dim, with scalar product, complete

kernel

reproducing property (coming next)