Support vector machines

optimal sepatrating hyperplane, kernels (jádra)

Separating Hyperplane

- Hyperplane splits the space in two parts.
- We try to separate -1,1 (red, green).
- Many possibilities:
  - Not necessary optimal
    - linear regression
    - LDA
  - Optimal if exists:
    - logistic regression
    - neural network
      - different for different initialisation, order of examples.
- We want to define unique optimal hyperplane.

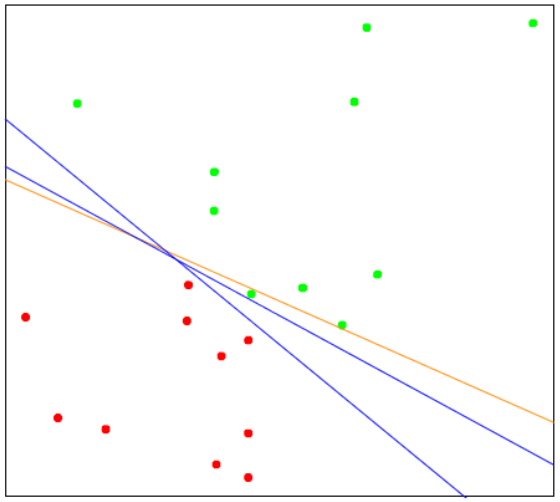


FIGURE 4.14. *A toy example with two classes separable by a hyperplane. The orange line is the least squares solution, which misclassifies one of the training points. Also shown are two blue separating hyperplanes found by the* perceptron learning algorithm *with different random starts.*

Hyperplane L (affine set; nadrovina)

- L defined by an equation:

$$\beta_0 + \beta^T x = 0$$

- For any point on L:

$$\beta^T x = -\beta_0$$

- Vector normal is defined:

$$\beta^* = \frac{\beta}{||\beta||}$$

- The signed distance to L is:

$$\beta^{*T}(x - x_0) = \frac{1}{||\beta||}(\beta^T x + \beta_0)$$

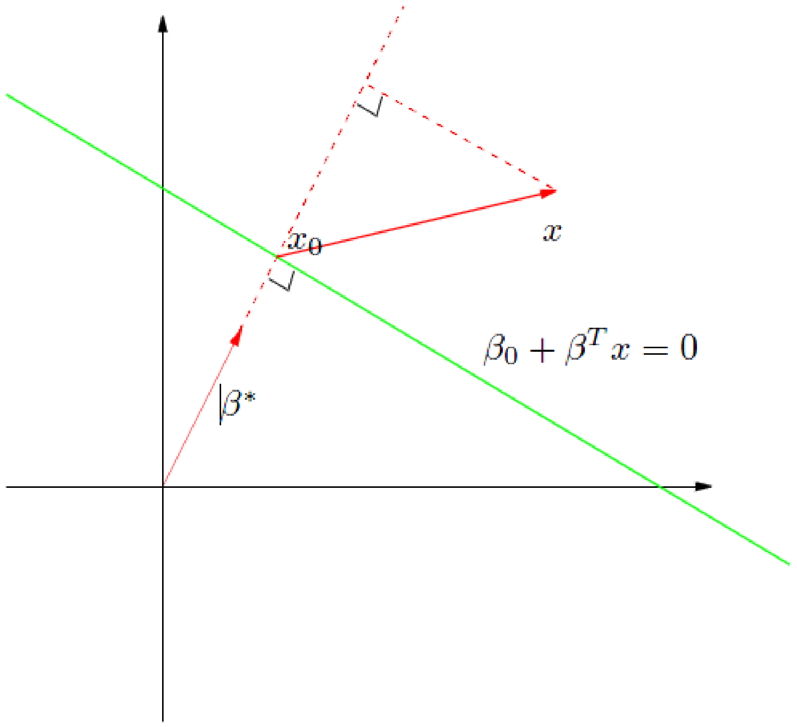


FIGURE 4.15. *The linear algebra of a hyperplane (affine set).*

Optimal Separating Hyperplane
(separable case)

- Separates training cases correctly,
- maximizes the margin between classes

  (that is maximizes the robustness of the split).

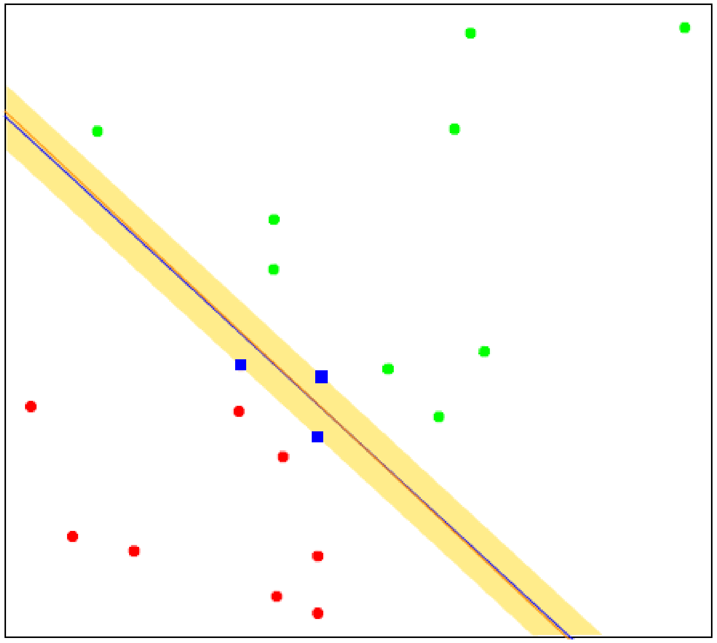


FIGURE 4.16. *The same data as in Figure 4.14. The shaded region deli the maximum margin separating the two classes. There are three support indicated, which lie on the boundary of the margin, and the optimal separating*

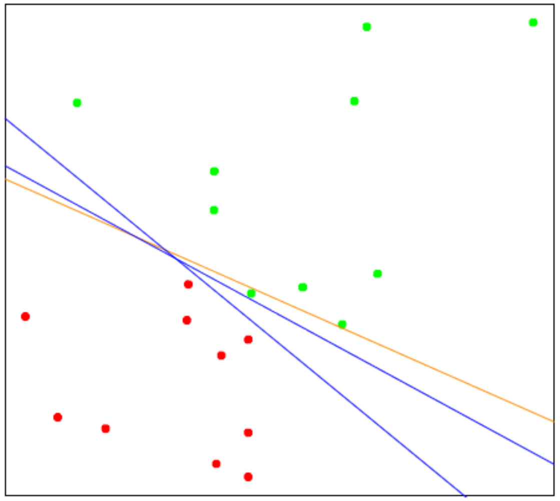
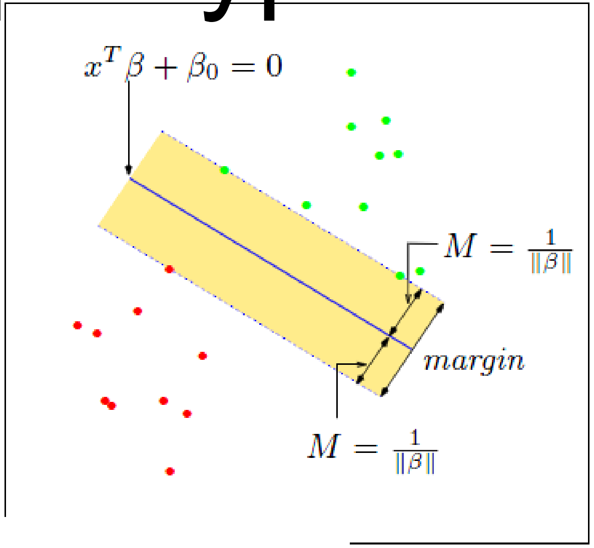


FIGURE 4.14. *A toy example with two classes separable by a hyperplane. T orange line is the least squares solution, which misclassifies one of the traini points. Also shown are two blue separating hyperplanes found by the* perceptr learning algorithm *with different random starts.*

Search for the Optimal Sep. Hyper.



- We search:

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

  under conditions:

$$y_i(x_i^T\beta + \beta_0) \geq M \text{ for all examples } i = 1, \dots, N.$$

  (that is all examples are on correct side).

- Technical staff:

$\|\beta\| = 1$ we move to the condition (and change β_0):

$$\max_{\beta,\beta_0} M$$

under conditions $\frac{1}{\|\beta\|} y_i(x_i^T\beta + \beta_0) \geq M$ for all examples $i = 1, \dots, N$.

Lagrange Multipliers

$$\max_{\beta,\beta_0} M$$

under conditions $\frac{1}{\|\beta\|} y_i(x_i^T \beta + \beta_0) \geq M$ for all examples $i = 1, \dots, N$.

If it holds for one β, β_0, it holds for all positive multiplications. We can choose $\|\beta\| = \frac{1}{M}$ and we get:

$$\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2$$

under conditions $y_i(x_i^T \beta + \beta_0) \geq 1$ pro $i = 1, \dots, N$.
It is convex optimization problem, that can be solved by Lagrange functions:

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

Wolfe Dual Form

$$L_P = min_{\beta,\beta_0} \left(\frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(x_i^T \beta + \beta_0) - 1] \right)$$

We set the derivative $= 0$ and we get:

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

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

Replacing L_P we get Wolfe dual form:

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k x_i^T x_k$$

that we maximize under conditions $\alpha_i \geq 0$

Support Vectors - $\alpha_i > 0$

It is convex optimization problem with known numeric algorithms.

The solution fulfills Karush–Kuhn–Tucker condition:

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

for all i.

For any $\alpha_i > 0$ must $[y_i(x_i^T\beta + \beta_0) - 1] = 0$ therefore

- either x_i is on the border,
- for all x_i outside the border is $\alpha_i = 0$.

The resulting border depends only on vectors on the border that are called **support vectors**.

Support Vector Classification !

- For each example, we get $\hat{\alpha}_i$
- We calculate

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

- for any support point on the boundary

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

$$\widehat{\beta_0} = y_i - x_s^T \hat{\beta}$$

- The resulting classification is:

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

High Sensitivity on Input (Overfitting?)

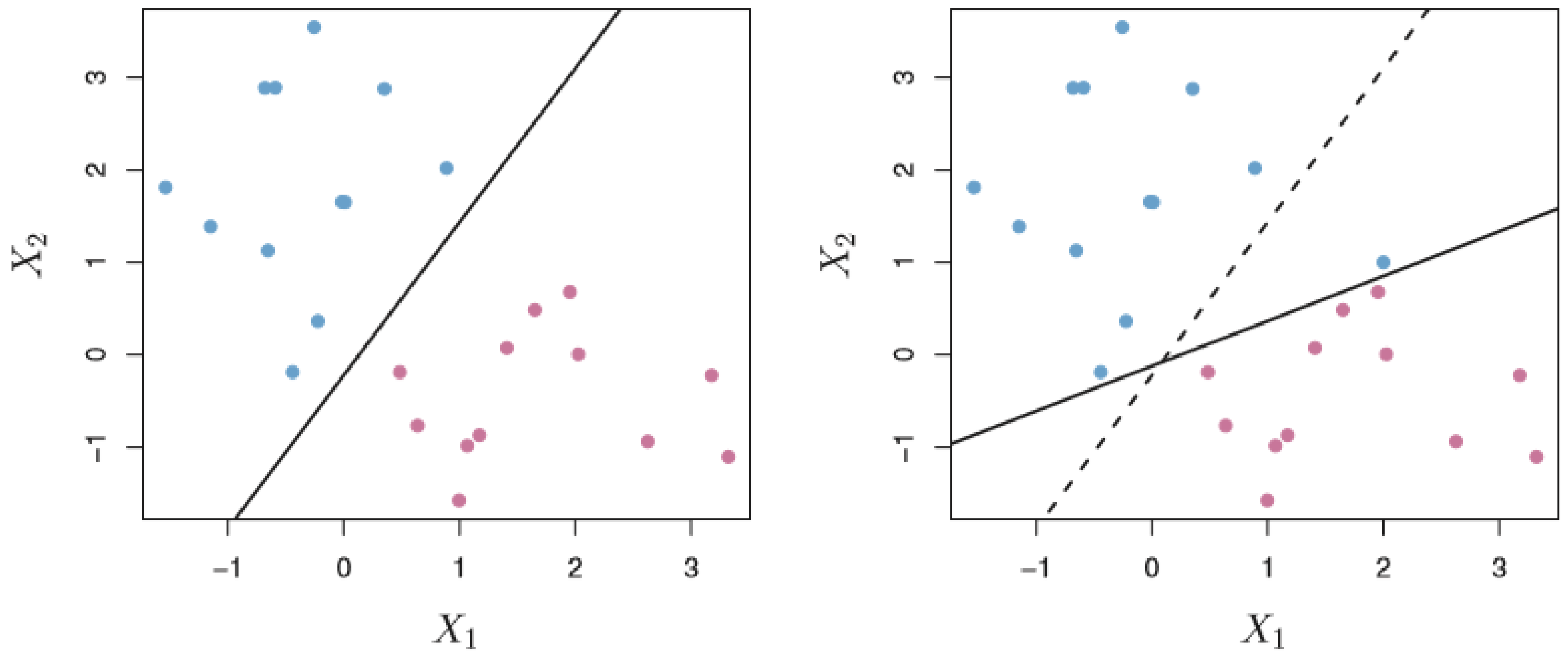


FIGURE 9.5. Left: *Two classes of observations are shown in blue and in purple, along with the maximal margin hyperplane.* Right: *An additional blue observation has been added, leading to a dramatic shift in the maximal margin hyperplane shown as a solid line. The dashed line indicates the maximal margin hyperplane that was obtained in the absence of this additional point.*

Non-Separable Case: Slacks

- Classes may not be separable.
- For each example, we introduce SLACK variable:
  - on proper side of the margin: $\xi_i = 0$
  - example is on the wrong side: $\xi_i = 1 - \frac{1}{M} y_i (x^T \beta + \beta_0)$
- Incorrectly classified examples have slack at least 1.
- We restrict the sum of slacks or introduce a penalty coefficient on the sum of slacks.

General Optimal Separating Hypp.

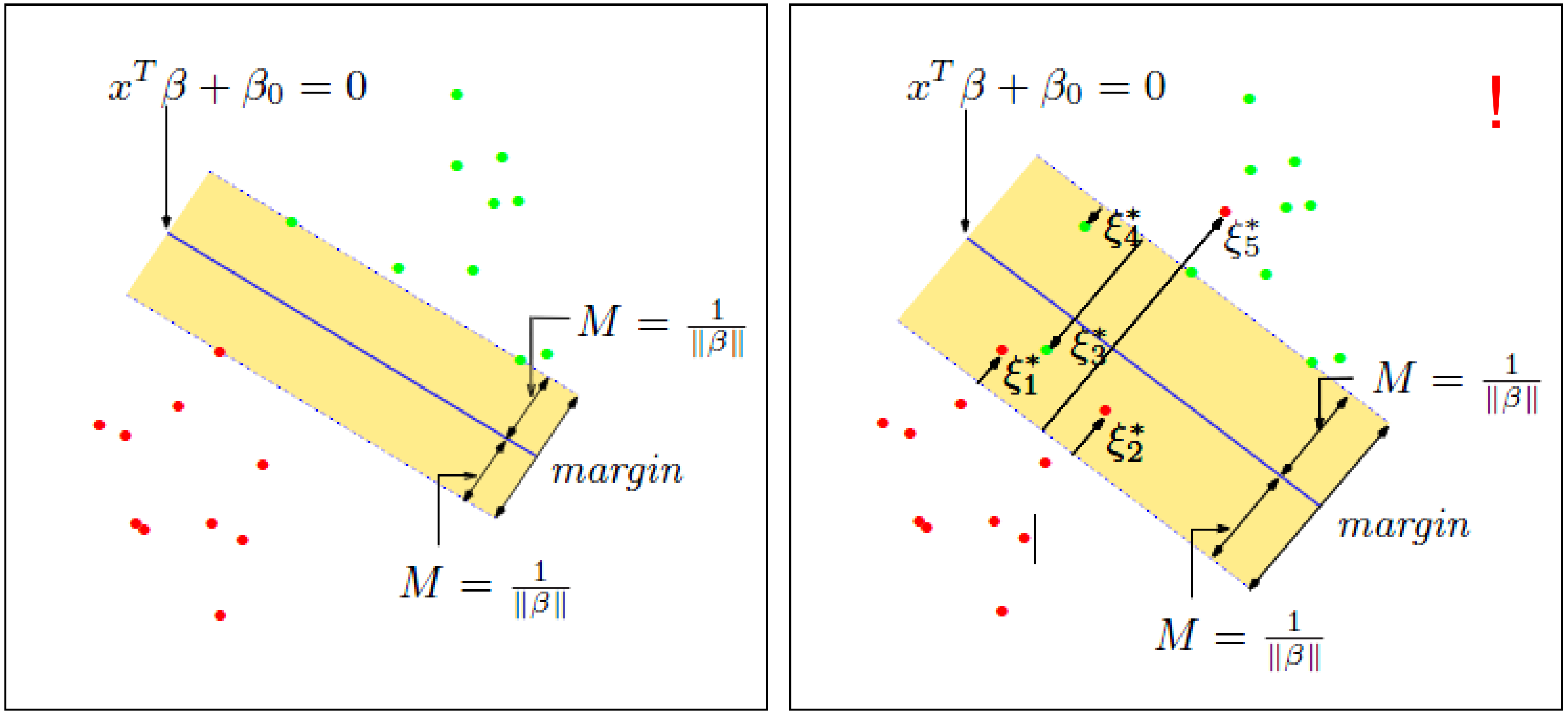


FIGURE 12.1. *Support vector classifiers. The left panel shows the separable case. The decision boundary is the solid line, while broken lines bound the shaded maximal margin of width $2M = 2/\|\beta\|$. The right panel shows the nonseparable (overlap) case. The points labeled ξ_j^* are on the wrong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

Optimal Separating Hyperplane

We solve optimization

$$\max_{\beta,\beta_0,\|\beta\|=1} M - \gamma_0 \sum_{i=1}^{N} \xi_i$$

under (new $2 \cdot N$) conditions:

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

and

$$\xi_i \geq 0.$$

- It is equivalent to maximize M $= \frac{1}{\|\beta\|}$
- γ=infinity for separable case.
- y_i ϵ {-1, 1} encodes the goal class.

!

Lagrange Functions ...

$$\max_{\beta,\beta_0,\|\beta\|=1} M - \gamma_0 \sum_{i=1}^{N} \xi_i$$

M replaced by $\frac{1}{\|\beta\|}$:

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

constrained $\xi_i \geq 0$ and $y_i(x^T\beta + \beta_0) \geq (1 - \xi_i)$.

Lagrange functions again: $\alpha_i, \mu_i \geq 0$:

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

Derivatives 0, Dual

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

Derivatives set to zero:

$$\begin{aligned}\beta &= \sum_{i=1}^{N}\alpha_i y_i x_i\\ 0 &= \sum_{i=1}^{N}\alpha_i y_i\\ \alpha_i &= \gamma-\mu_i\end{aligned}$$

Wolfe dual:

$$max_{\alpha_i,\mu_i}\left(L_D = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{N}\alpha_i\alpha_k y_i y_k x_i^T x_k\right)$$

constrained by $0 \le \alpha_i \le \gamma$ and $\sum_{i=1}^{N}\alpha_i y_i = 0$.

- Convex optimization problem with linear constraints,
- solution exists,
- we get α_i for each data point.

Support vectors $\widehat{\alpha}_i > 0$ are either

- on the border: $\widehat{\xi}_i = 0$ (and $0 < \widehat{\alpha}_i < \gamma$),
- on wrong side of the margin: $\widehat{\xi}_i > 0$ (and $\widehat{\alpha}_i = \gamma$).

Solution has following properties:

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

Separating Hyperplane - General

- For each data point, get α_i.

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

Calculate $\widehat{\beta_0} = y_m - x_m^T \hat{\beta}$ from any example on the margin with $\xi_m = 0$:

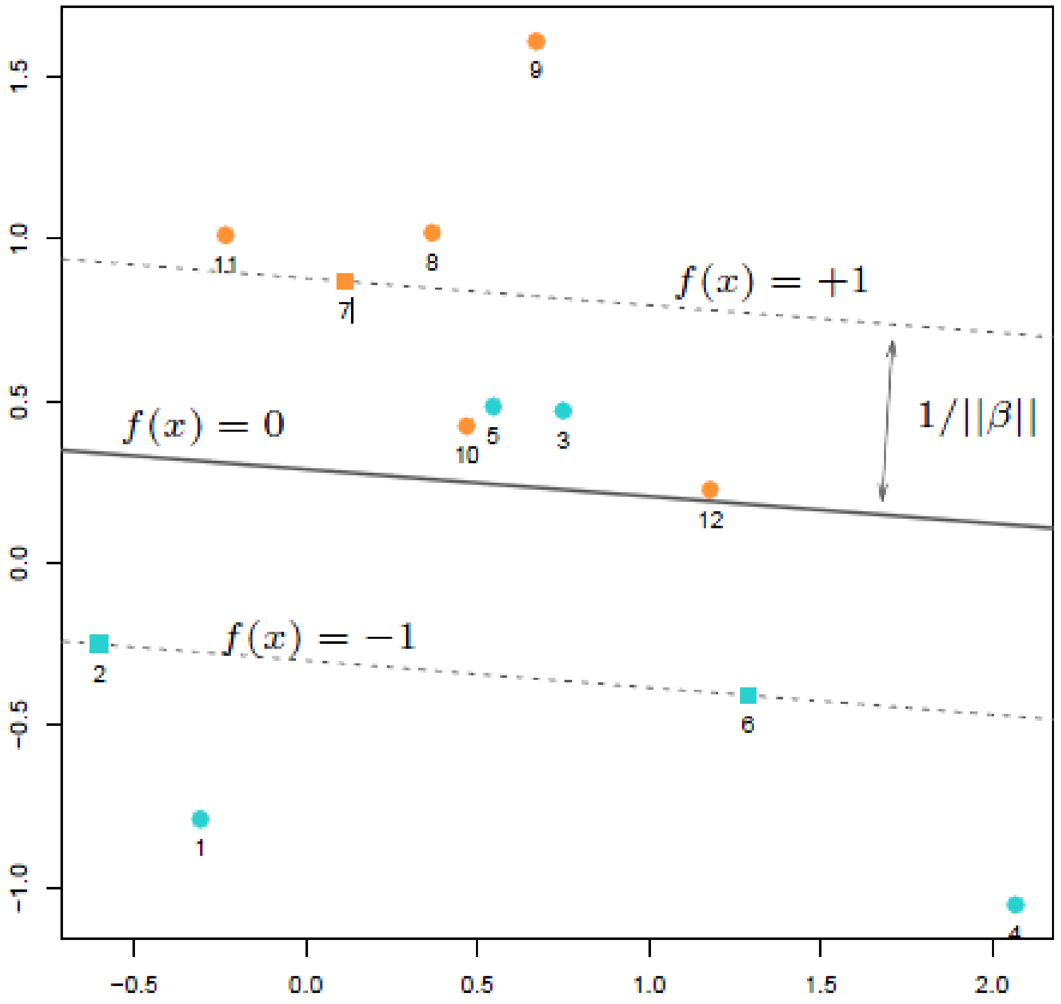
$$\alpha_m \left[y_m (x_m^T \widehat{\beta} + \widehat{\beta_0}) - (1 - 0) \right] = 0$$

Parameter γ set by crossvalidation.

Example

Which vectors have

- non-zero alpha?
- non-zero slack?
- Support vectors?
- Incorrectly classified vectors?



Optimal Separating Hyperplane

- Goal class is coded by -1,1.
- We use optimization procedure, for each example we get α_i
- Select any example with $0 < \hat{\alpha}_i < \gamma$, denote it s.
- New example x is classified:

$$\hat{G}(x) = \sum_{i=1}^{N} \alpha_i y_i x^T x_i + (y_s - x_s^T \beta).$$

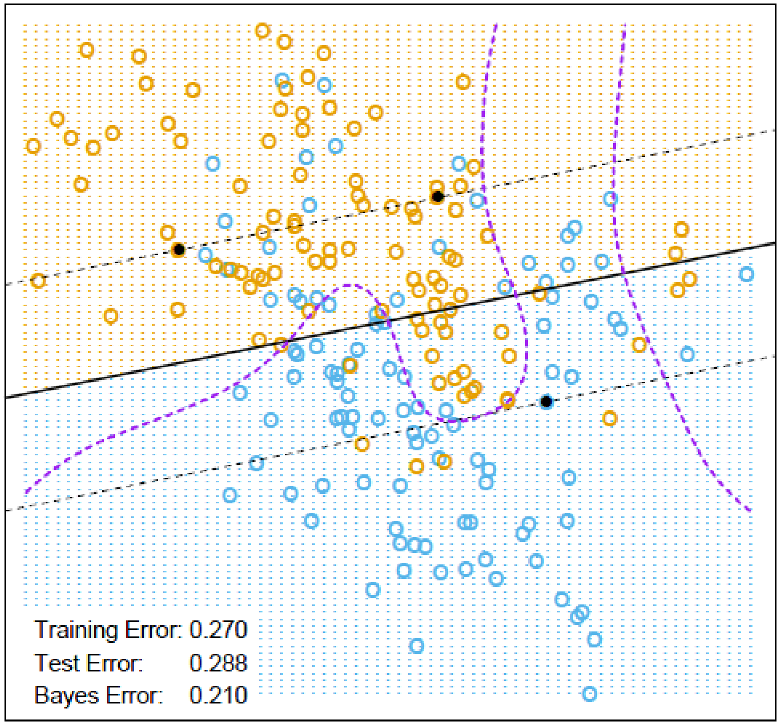
that is

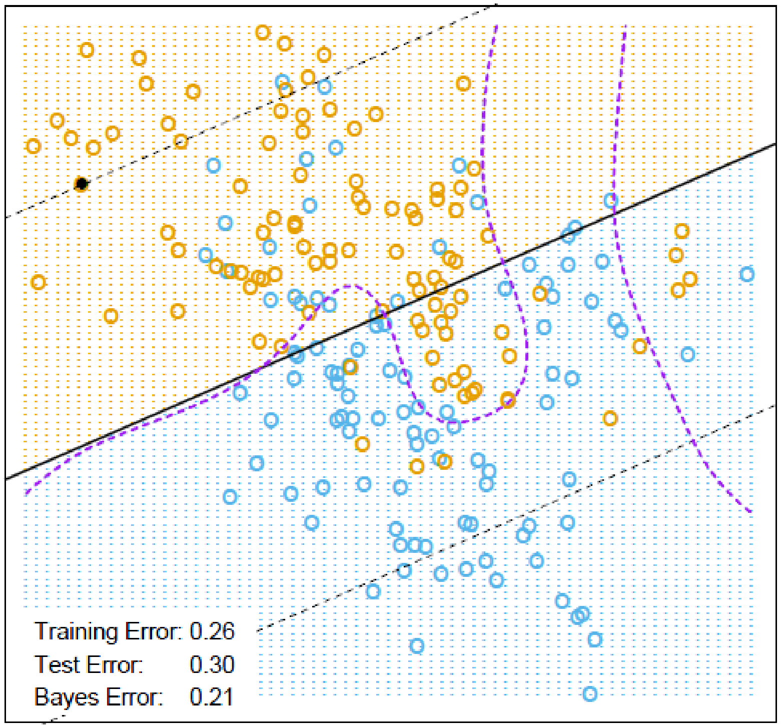
$$\hat{G}(x) = sign(x^T \beta + \beta_0),$$

!

- Parameter gamma is set by tunning (CV).

Different Cost Penalty





$\gamma = 10000$ $\gamma = 0.01$

FIGURE 12.2. *The linear support vector boundary for the mixture data example with two overlapping classes, for two different values of γ. The broken lines indicate the margins, where $f(x) = \pm 1$. The support points ($\alpha_i > 0$) are all the points on the wrong side of their margin. The black solid dots are those support points falling exactly on the margin ($\xi_i = 0$, $\alpha_i > 0$). In the upper panel 62% of the observations are support points, while in the lower panel 85% are. The broken purple curve in the background is the Bayes decision boundary.*

Linear Cut may be sub-optimal

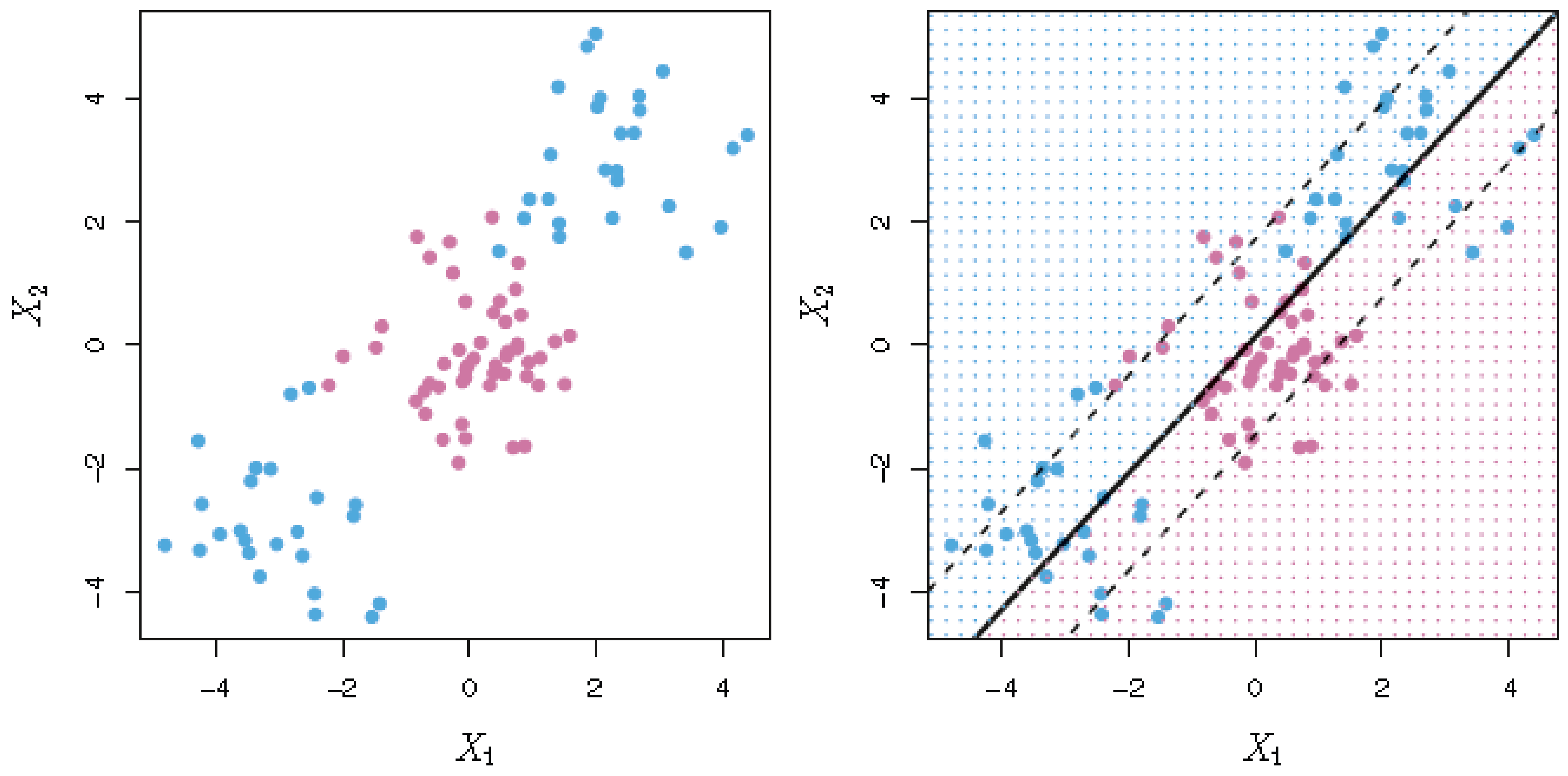


FIGURE 9.8. Left: *The observations fall into two classes, with a non-linear boundary between them.* Right: *The support vector classifier seeks a linear boundary, and consequently performs very poorly.*

Support vector machines

- Support vector machines look for linear separating hyperplane in nonlinear transformation of the input space
- Find the difference:

$$\hat{G}(x) = \sum_{i=1}^{N} \alpha_i y_i x^T x_i + (y_s - x_s^T \beta).$$

$$\hat{G}(x) = \sum_{i=1}^{N} \alpha_i y_i K(x,x_i) + (y_s - x_s^T \beta).$$

!

SVM -Example

- Original feature space X_1, X_2 transform to $H_M(X_1, X_2) = \{1, \sqrt(2) \cdot X_1, \sqrt(2) \cdot X_2, \sqrt(2) \cdot X_1 \cdot X_2, X_1^2, X_2^2\}$.
- Find optimal separating hyperplane in the feature space H_M, $\widehat{f}([X_1, X_2]) = \left[1, \sqrt(2) \cdot X_1, \sqrt(2) \cdot X_2, \sqrt(2) \cdot X_1 \cdot X_2, X_1^2, X_2^2\right] \widehat{\beta} + \widehat{\beta}_0$.
- This is equivalent to find SVM classifier in the original feature space with the polynomial kernel degree 2: $K(x, x^|) = (1 + \langle x, x^| \rangle)^2$, that is:

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

Kernel Functions

- The most common kernel functions are:

dth Degree polynomial:	$K(x, x^{\vert}) = (1 + \langle x, x^{\vert} \rangle)^d$
Radial basis	$K(x, x^{\vert}) = exp(\frac{-\Vert x - x^{\vert} \Vert^2}{c})$
Neural network	$K(x, x^{\vert}) = tanh(\kappa_1 \langle x, x^{\vert} \rangle + \kappa_2)$

Consider for example a feature space with two inputs X_1 and X_2, and a polynomial kernel of degree 2. Then

$$\begin{aligned} K(X, X') &= (1 + \langle X, X' \rangle)^2 \\ &= (1 + X_1 X_1' + X_2 X_2')^2 \\ &= 1 + 2X_1 X_1' + 2X_2 X_2' + (X_1 X_1')^2 + (X_2 X_2')^2 + 2X_1 X_1' X_2 X_2'. \end{aligned} \tag{12.23}$$

Then $M = 6$, and if we choose $h_1(X) = 1$, $h_2(X) = \sqrt{2}X_1$, $h_3(X) = \sqrt{2}X_2$, $h_4(X) = X_1^2$, $h_5(X) = X_2^2$, and $h_6(X) = \sqrt{2}X_1X_2$, then $K(X, X') = \langle h(X), h(X') \rangle$. From (12.20) we see that the solution can be written

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

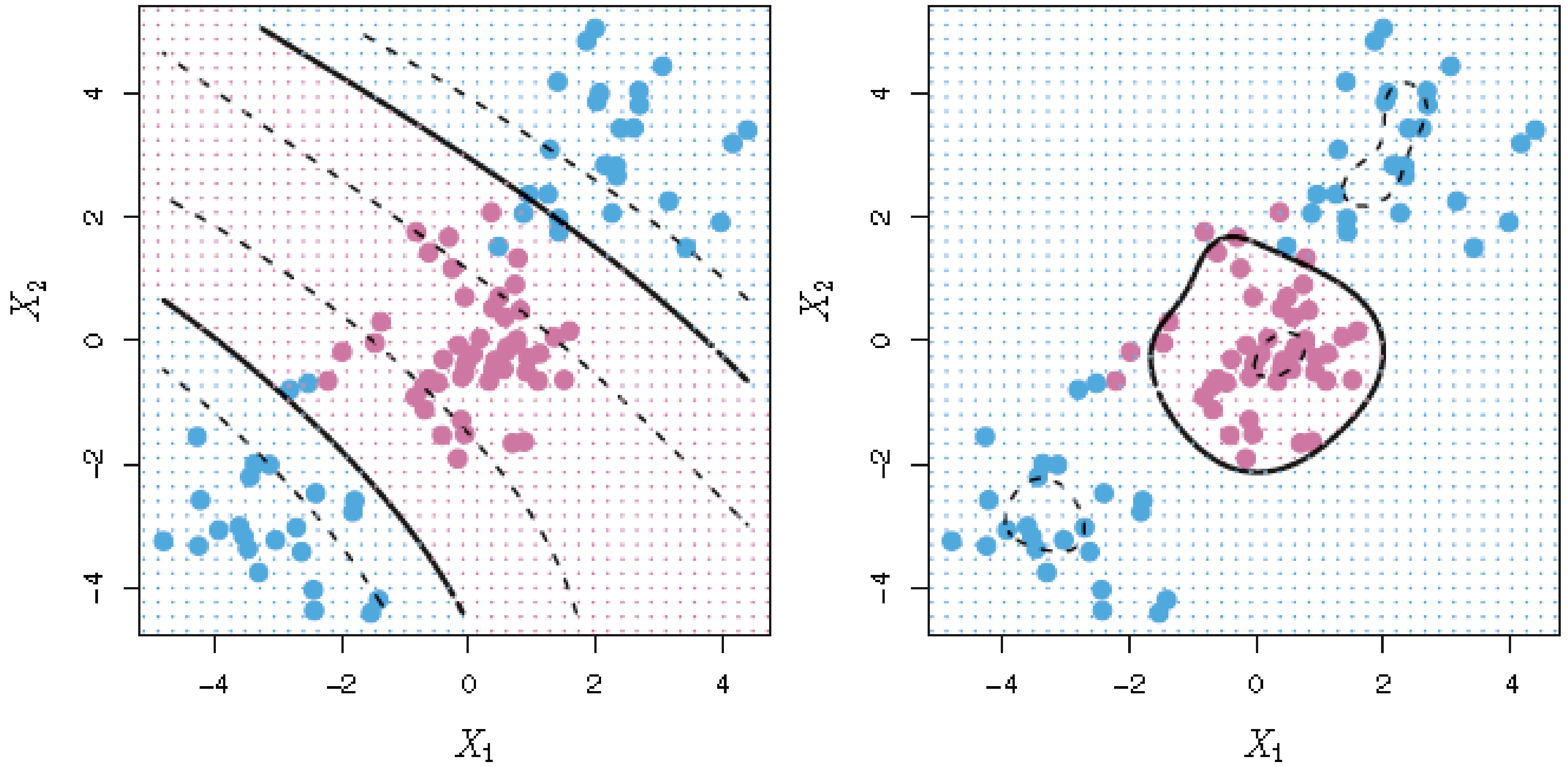


FIGURE 9.9. Left: *An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule.* Right: *An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.*

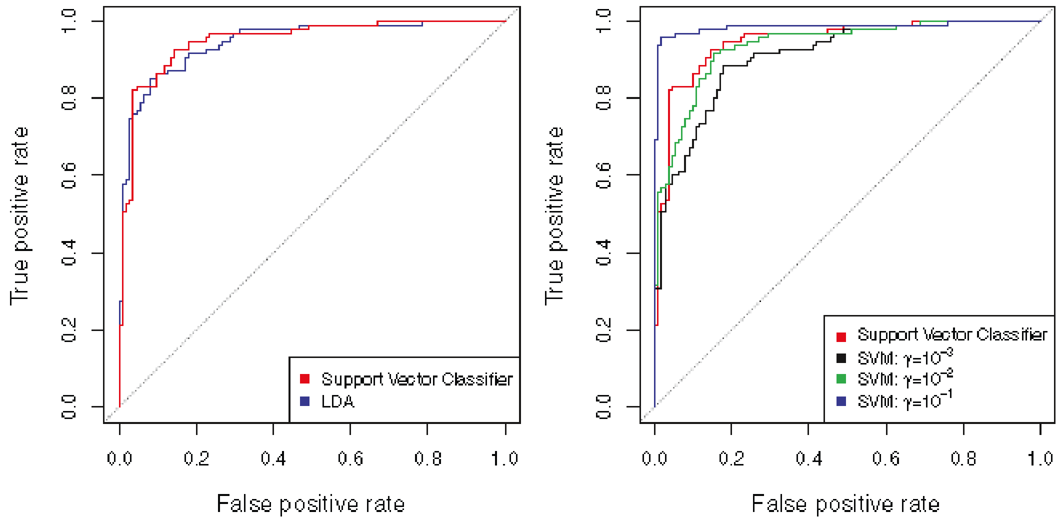


FIGURE 9.10. *ROC curves for the* Heart *data training set.* Left: *The support vector classifier and LDA are compared.* Right: *The support vector classifier is compared to an SVM using a radial basis kernel with* $\gamma = 10^{-3}$, 10^{-2}, *and* 10^{-1}.

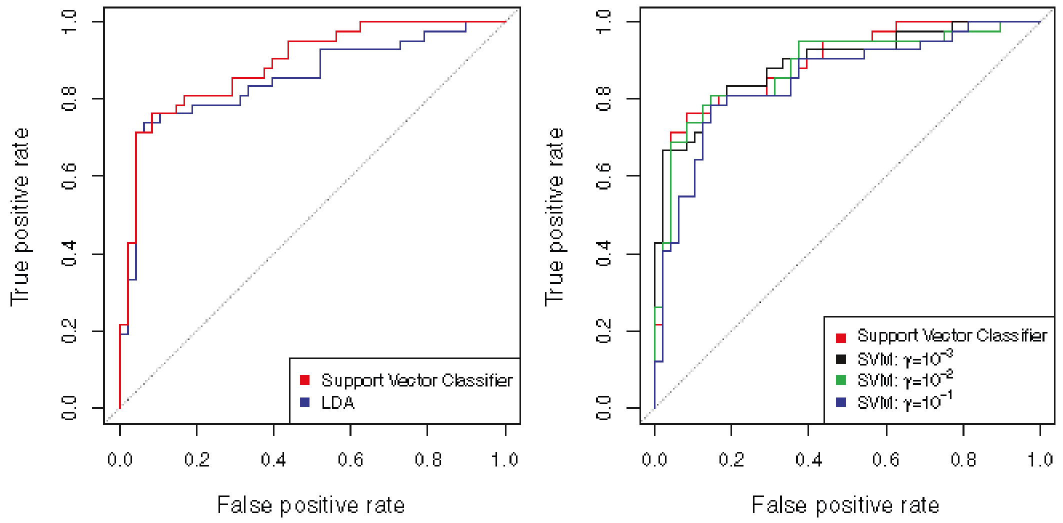


FIGURE 9.11. *ROC curves for the test set of the* Heart *data.* Left: *The support vector classifier and LDA are compared.* Right: *The support vector classifier is compared to an SVM using a radial basis kernel with* $\gamma = 10^{-3}$, 10^{-2}, *and* 10^{-1}.

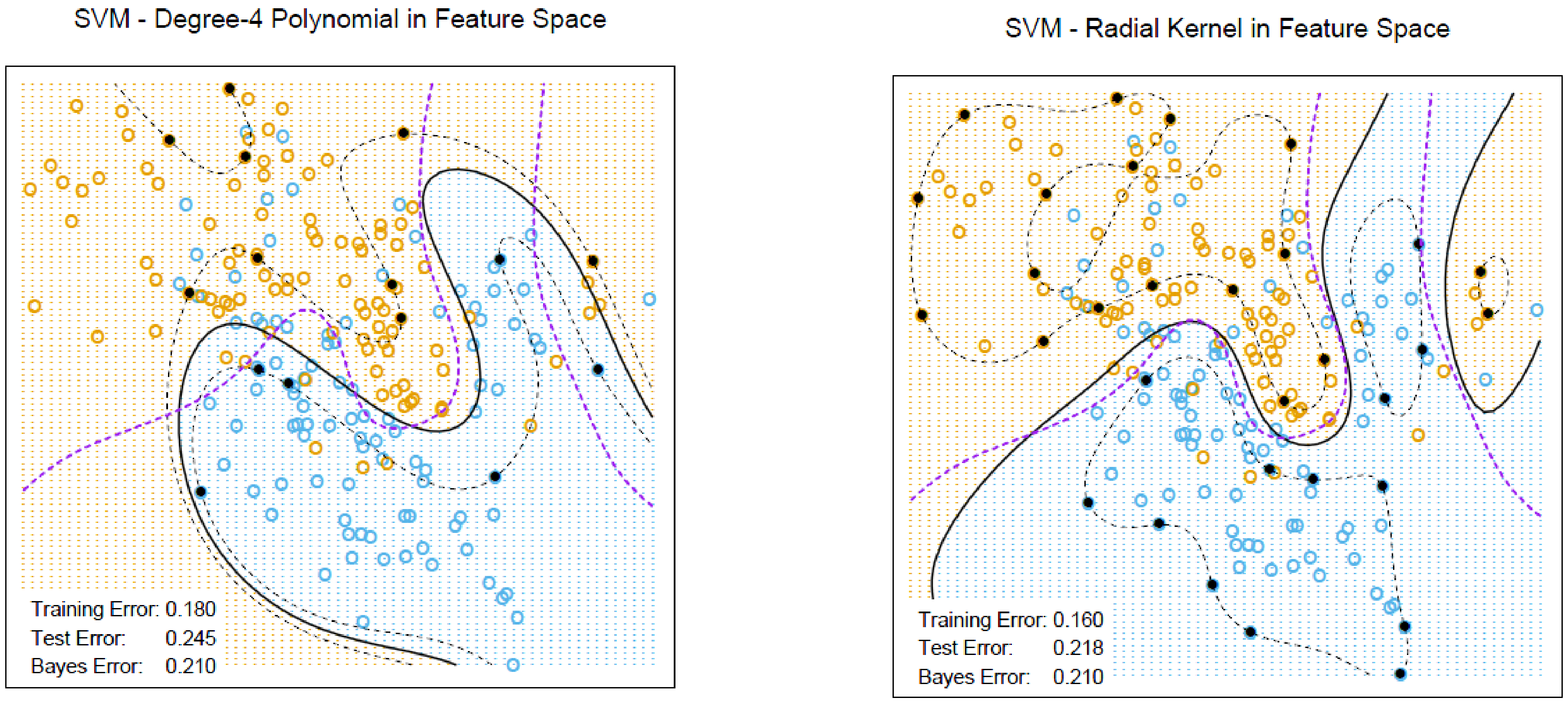


FIGURE 12.3. *Two nonlinear SVMs for the mixture data. The upper plot uses a 4th degree polynomial kernel, the lower a radial basis kernel (with $\gamma = 1$). In each case (γ was tuned to approximately achieve the best test error performance, and $\gamma = 1$ worked well in both cases. The radial basis kernel performs the best (close to Bayes optimal), as might be expected given the data arise from mixtures of Gaussians. The broken purple curve in the background is the Bayes decision boundary.*

SVM as penalization method

Both tasks lead to the same solution:

- Solve:

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

constrained $\xi_i \geq 0$ and $y_i(x^T\beta + \beta_0) \geq (1 - \xi_i)$.

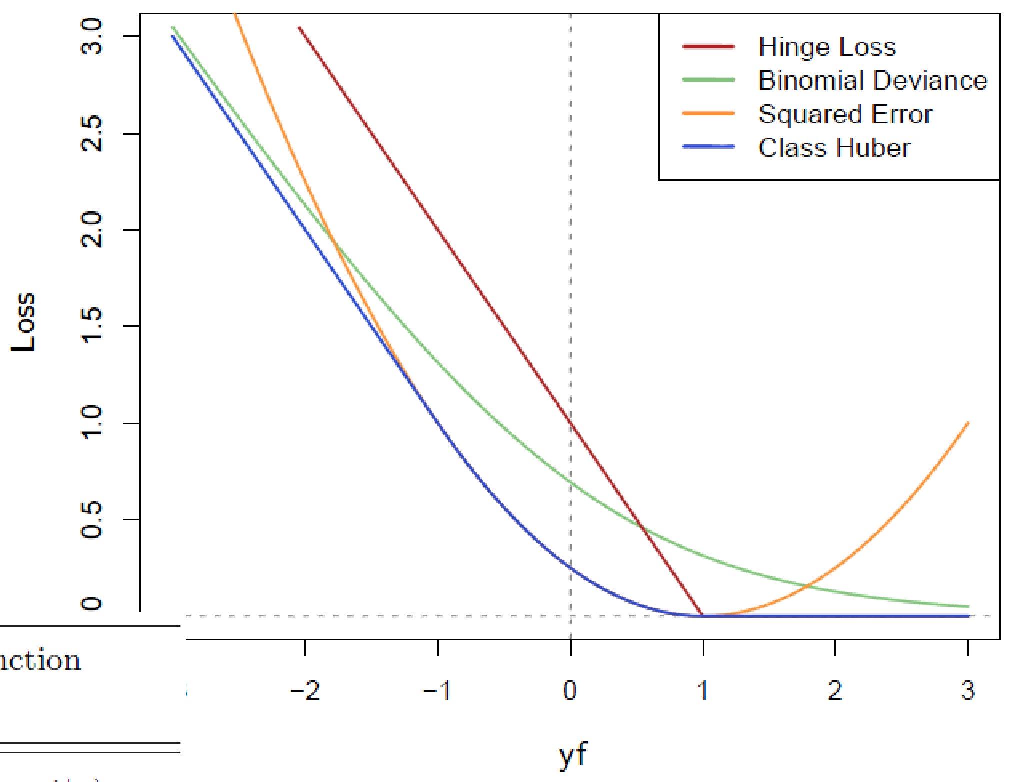
- Or: With $f(x) = h(x)^T\beta + \beta_0$, consider the optimization problem

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

where the subscript "+" indicates positive part.

Comparison of Loss Functions

FIGURE 12.4. *The support vector loss function (hinge loss), co negative log-likelihood loss (binomial deviance) for logistic regressio ror loss, and a "Huberized" version of the squared hinge loss. All a function of yf rather than f, because of the symmetry between the $y = -1$ case. The deviance and Huber have the same asymptotes loss, but are rounded in the interior. All are scaled to have the lin slope of -1.*



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

Skin of Orange Example

- Skin of Orange:
  - a ball in 4 dimensions
    - one class inside
    - second on the skin.
  - 6 additionall noisy features.

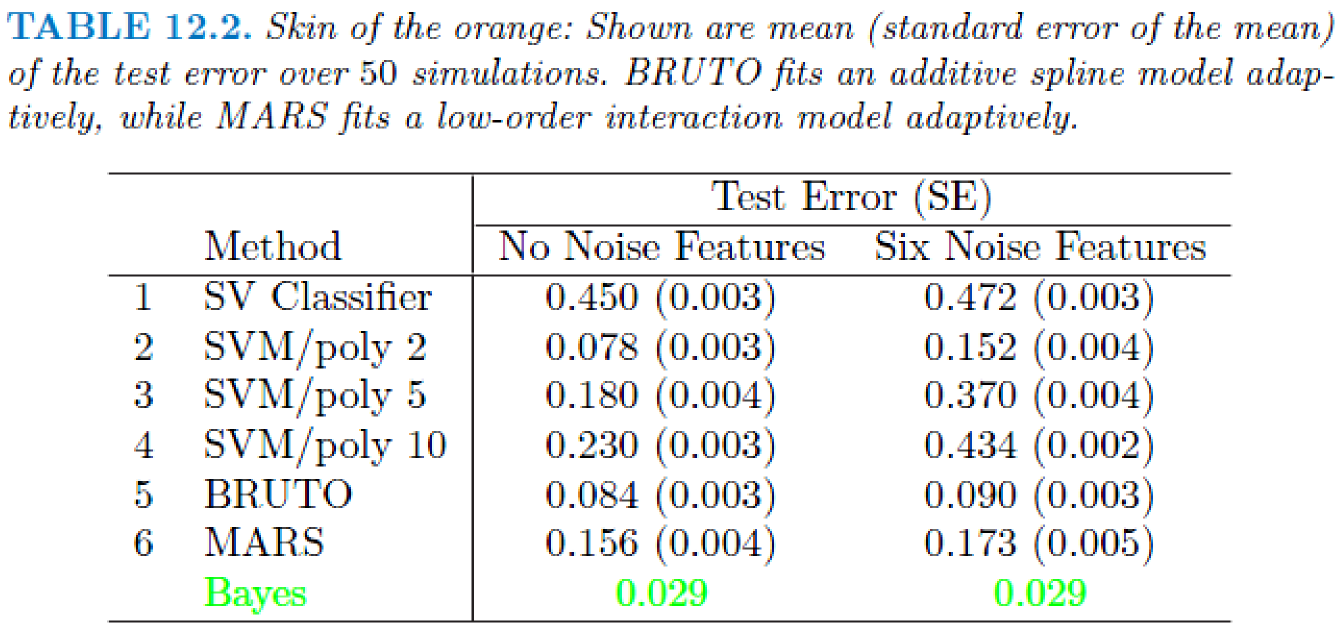


TABLE 12.2. *Skin of the orange: Shown are mean (standard error of the mean) of the test error over 50 simulations. BRUTO fits an additive spline model adaptively, while MARS fits a low-order interaction model adaptively.*

		Test Error (SE)	
	Method	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