Separating hyperplane, Optimal separating hyperplane

- Classification, we encode the goal class by -1 and 1, respectively.
- separate the space X by a hyperplane
- **Linear Discriminant Analysis** LDA is not necessary optimal.
- **Logistic regression** finds one if it exists.
- **Perceptron** (a neural network with one neuron) finds separating hyperplane if it exists.
  - The exact position depends on initial parameters.

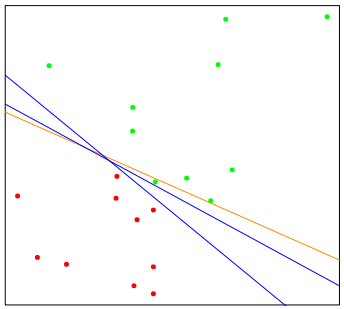


FIGURE 4.14. *A toy example with two classes separable by a hyperplane. The orange line is the least squares solution, which misclassifies one of the training points. Also shown are two blue separating hyperplanes found by the* perceptron learning algorithm *with different random starts.*

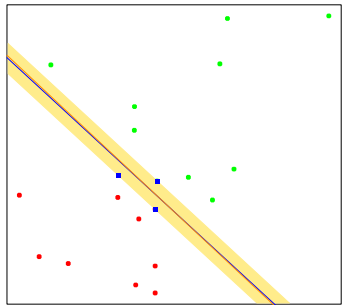


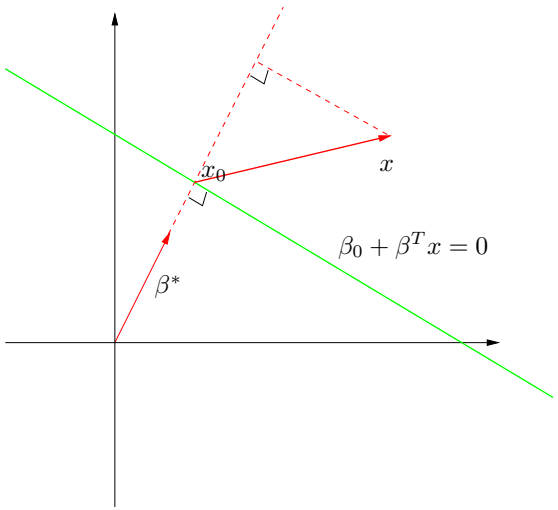
FIGURE 4.16. *The same data as in Figure 4.14. The shaded region delineates the maximum margin separating the two classes. There are three support points indicated, which lie on the boundary of the margin, and the optimal separating hyperplane (blue line) bisects the slab. Included in the figure is the boundary found using logistic regression (red line), which is very close to the optimal separating hyperplane (see Section 12.3.3).*

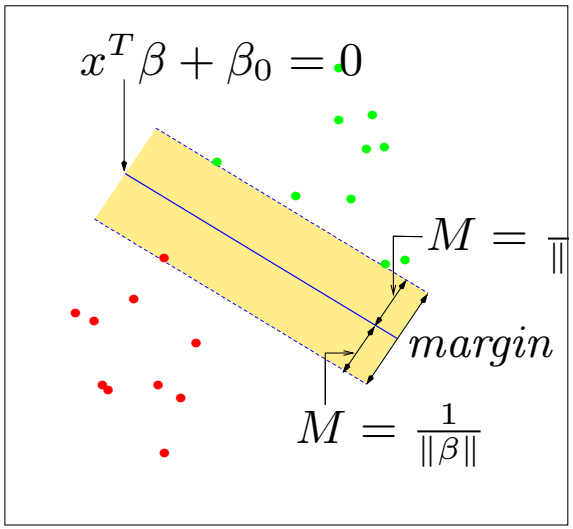
Optimal Separating Hyperplane (separble case)

We define **Optimal Separating Hyperplane** as a separating hyperplane with maximal free space M without any data point around the hyperplane.
Formally:

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

subject to $y_i(x_i^T\beta + \beta_0) \geq M$ for all $i = 1, \ldots, N$.





Formally:

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

subject to $y_i(x_i^T\beta + \beta_0) \geq M$ for all $i = 1, \ldots, N$.

We re-define: $\|\beta\| = 1$ can be moved to the condition (and redefine β_0):

$$\frac{1}{\|\beta\|} y_i(x_i^T\beta + \beta_0) \geq M$$

Since for any β and β_0 satisfying these inequalities, any positively scaled multiple satisfies them too, we can set $\|\beta\| = \frac{1}{M}$ and we get:

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

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

This is a convex optimization problem. The Lagrange function, we look for the saddle point w.r.t. β and β_0:

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

Formally:

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

subject to $y_i(x_i^T\beta + \beta_0) \geq M$ for all $i = 1, \ldots, N$.
We re-define: $\|\beta\| = 1$ can be moved to the condition (and redefine β_0):

$$\frac{1}{\|\beta\|} y_i(x_i^T\beta + \beta_0) \geq M$$

Since for any β and β_0 satisfying these inequalities, any positively scaled multiple satisfies them too, we can set $\|\beta\| = \frac{1}{M}$ and we get:

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

subject to $y_i(x_i^T\beta + \beta_0) \geq 1$ pro $i = 1, \ldots, N$.
This is a convex optimization problem. The Lagrange function, we look for the saddle point w.r.t. β and β_0:

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

Formally:

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

subject to $y_i(x_i^T\beta + \beta_0) \geq M$ for all $i = 1, \ldots, N$.
We re-define: $\|\beta\| = 1$ can be moved to the condition (and redefine β_0):

$$\frac{1}{\|\beta\|} y_i(x_i^T\beta + \beta_0) \geq M$$

Since for any β and β_0 satisfying these inequalities, any positively scaled multiple satisfies them too, we can set $\|\beta\| = \frac{1}{M}$ and we get:

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

subject to $y_i(x_i^T\beta + \beta_0) \geq 1$ pro $i = 1, \ldots, N$.
This is a convex optimization problem. The Lagrange function, we look for the saddle point w.r.t. β and β_0:

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

Formally:

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

subject to $y_i(x_i^T\beta + \beta_0) \geq M$ for all $i = 1, \ldots, N$.
We re-define: $\|\beta\| = 1$ can be moved to the condition (and redefine β_0):

$$\frac{1}{\|\beta\|} y_i(x_i^T\beta + \beta_0) \geq M$$

Since for any β and β_0 satisfying these inequalities, any positively scaled multiple satisfies them too, we can set $\|\beta\| = \frac{1}{M}$ and we get:

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

subject to $y_i(x_i^T\beta + \beta_0) \geq 1$ pro $i = 1, \ldots, N$.
This is a convex optimization problem. The Lagrange function, we look for the saddle point w.r.t. β and β_0:

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

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

Setting the derivatives to zero, we obtain:

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

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

Substituing these in L_P we obtain the so–called Wolfe dual:

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

subject to $\alpha_i \geq 0$

The solution is obtained by maximizing L_D in the positive orthant, for which standard software can be used.

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

Setting the derivatives to zero, we obtain:

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

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

Substituing these in L_P we obtain the so–called Wolfe dual:

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

subject to $\alpha_i \geq 0$

The solution is obtained by maximizing L_D in the positive orthant, for which standard software can be used.

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

Setting the derivatives to zero, we obtain:

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

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

Substituing these in L_P we obtain the so–called Wolfe dual:

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

subject to $\alpha_i \geq 0$

The solution is obtained by maximizing L_D in the positive orthant, for which standard software can be used.

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

Setting the derivatives to zero, we obtain:

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

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

Substituing these in L_P we obtain the so–called Wolfe dual:

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

subject to $\alpha_i \geq 0$
The solution is obtained by maximizing L_D in the positive orthant, for which standard software can be used.

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

subject to $\alpha_i \geq 0$.

In addition the solution must satisfy the Karush–Kuhn–Tucker conditions:

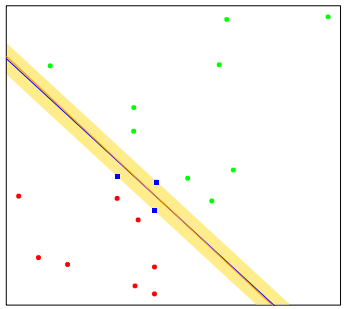
$$\alpha_i [y_i (x_i^T \beta + \beta_0) - 1] = 0$$

for any i, therefore for any $\alpha_i > 0$ must $[y_i (x_i^T \beta + \beta_0) - 1] = 0$, that means x_i is on the boundary and for all x_i outside the boundary is $\alpha_i = 0$.
The boundary is defined by x_i with $\alpha_i > 0$ – so called **support vectors**.

We classify new observations

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

- where $\beta = \sum_{i=1}^{N} \alpha_i y_i x_i$,
- $\beta_0 = y_s - x_s^T \beta$ for any support vector $\alpha_s > 0$.



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

subject to $\alpha_i \geq 0$.
In addition the solution must satisfy the Karush–Kuhn–Tucker conditions:

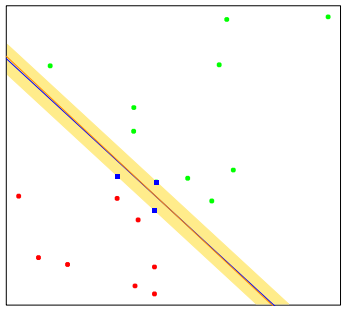
$$\alpha_i [y_i (x_i^T \beta + \beta_0) - 1] = 0$$

for any i, therefore for any $\alpha_i > 0$ must $[y_i (x_i^T \beta + \beta_0) - 1] = 0$, that means x_i is on the boundary and for all x_i outside the boundary is $\alpha_i = 0$.
The boundary is defined by x_i with $\alpha_i > 0$ – so called **support vectors**.

We classify new observations

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

- where $\beta = \sum_{i=1}^{N} \alpha_i y_i x_i$,
- $\beta_0 = y_s - x_s^T \beta$ for any support vector $\alpha_s > 0$.



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

subject to $\alpha_i \geq 0$.
In addition the solution must satisfy the Karush–Kuhn–Tucker conditions:

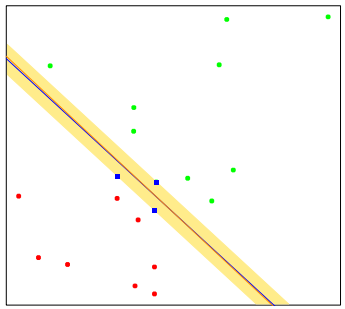
$$\alpha_i [y_i (x_i^T \beta + \beta_0) - 1] = 0$$

for any i, therefore for any $\alpha_i > 0$ must $[y_i (x_i^T \beta + \beta_0) - 1] = 0$, that means x_i is on the boundary and for all x_i outside the boundary is $\alpha_i = 0$.
The boundary is defined by x_i with $\alpha_i > 0$ – so called **support vectors**.

We classify new observations

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

- where $\beta = \sum_{i=1}^{N} \alpha_i y_i x_i$,
- $\beta_0 = y_s - x_s^T \beta$ for any support vector $\alpha_s > 0$.



Optimal Separating Hyperplane (nonseparble case)

- We have to accept incorrectly classified instances in a non–separable case.
- We limit the number of incorrectly classified examples.

We define **slack** ξ for each data point $(\xi_1, \ldots, \xi_N) = \xi$ as follows:

- ξ_i is the distance of x_i from the boundary for x_i at the wrong side of the margin
- and $\xi_i = 0$, for x_i at the correct side.

We require $\sum_{i=1}^{N} \xi_i \leq K$.

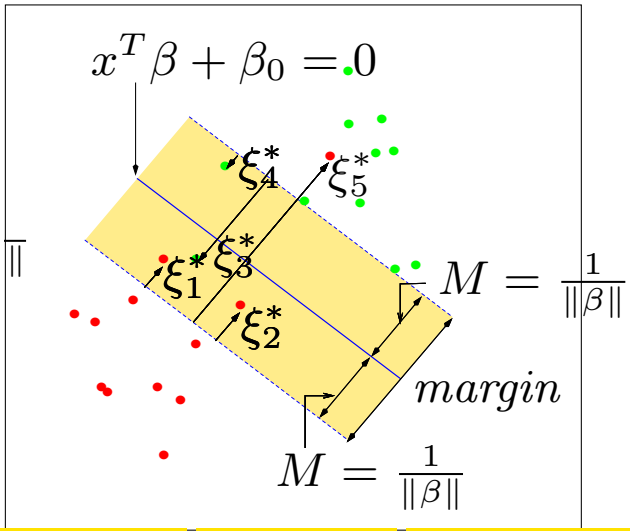
We solve the optimization problem

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

subject to:

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

where $\forall i$ is $\xi_i \geq 0$ a $\sum_{i=1}^{N} \xi_i \leq K$.



Optimal Separating Hyperplane (nonseparble case)

- We have to accept incorrectly classified instances in a non–separable case.
- We limit the number of incorrectly classified examples.

We define **slack** ξ for each data point $(\xi_1, \ldots, \xi_N) = \xi$ as follows:

- ξ_i is the distance of x_i from the boundary for x_i at the wrong side of the margin
- and $\xi_i = 0$, for x_i at the correct side.

We require $\sum_{i=1}^{N} \xi_i \leq K$.

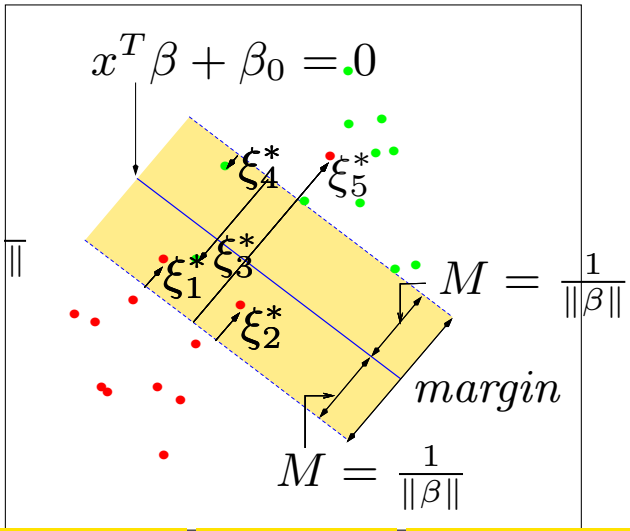
We solve the optimization problem

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

subject to:

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

where $\forall i$ is $\xi_i \geq 0$ a $\sum_{i=1}^{N} \xi_i \leq K$.



Optimal Separating Hyperplane (nonseparble case)

- We have to accept incorrectly classified instances in a non–separable case.
- We limit the number of incorrectly classified examples.

We define **slack** ξ for each data point $(\xi_1, \ldots, \xi_N) = \xi$ as follows:

- ξ_i is the distance of x_i from the boundary for x_i at the wrong side of the margin
- and $\xi_i = 0$, for x_i at the correct side.

We require $\sum_{i=1}^{N} \xi_i \leq K$.

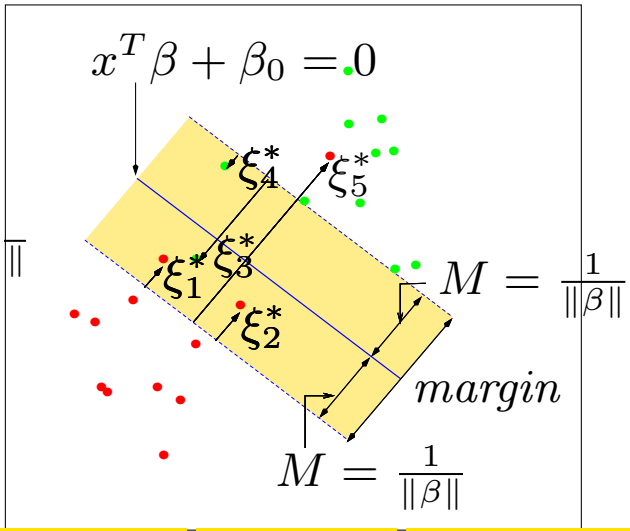
We solve the optimization problem

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

subject to:

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

where $\forall i$ is $\xi_i \geq 0$ a $\sum_{i=1}^{N} \xi_i \leq K$.



Optimal Separating Hyperplane (nonseparble case)

- We have to accept incorrectly classified instances in a non–separable case.
- We limit the number of incorrectly classified examples.

We define **slack** ξ for each data point $(\xi_1, \ldots, \xi_N) = \xi$ as follows:

- ξ_i is the distance of x_i from the boundary for x_i at the wrong side of the margin
- and $\xi_i = 0$, for x_i at the correct side.

We require $\sum_{i=1}^{N} \xi_i \leq K$.

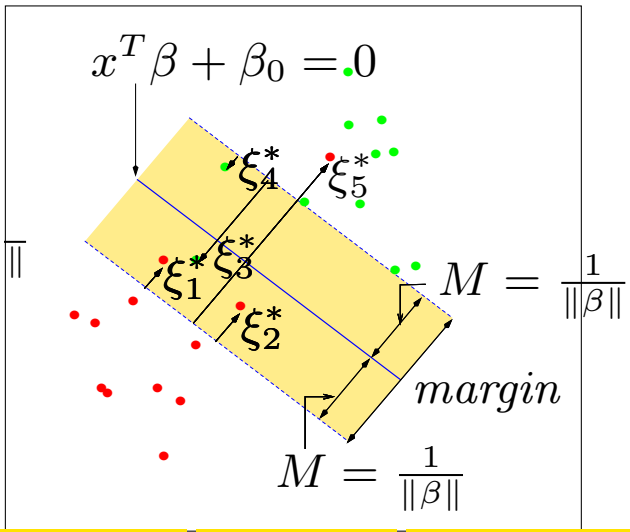
We solve the optimization problem

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

subject to:

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

where $\forall i$ is $\xi_i \geq 0$ a $\sum_{i=1}^{N} \xi_i \leq K$.



Optimal Separating Hyperplane (nonseparble case)

Again, we omit replace the condition $\|\beta\|$ by defining $M = \frac{1}{\|\beta\|}$ and optimize

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

We replace the constant by a multiplicative parameter γ and solve

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

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

- We can set $\gamma = \infty$ for the separable case.
- Large γ: a complex boundary, fewer support vectors.
- Small γ: a smooth boundary, a robust model, many support vectors.
- γ usually set by cross-validation.

Optimal Separating Hyperplane (nonseparble case)

Again, we omit replace the condition $\|\beta\|$ by defining $M = \frac{1}{\|\beta\|}$ and optimize

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

We replace the constant by a multiplicative parameter γ and solve

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

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

- We can set $\gamma = \infty$ for the separable case.
- Large γ: a complex boundary, fewer support vectors.
- Small γ: a smooth boundary, a robust model, many support vectors.
- γ usually set by cross validation.

Optimal Separating Hyperplane (nonseparble case)

Again, we omit replace the condition $\|\beta\|$ by defining $M = \frac{1}{\|\beta\|}$ and optimize

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

We replace the constant by a multiplicative parameter γ and solve

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

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

- We can set $\gamma = \infty$ for the separable case.
- Large γ: a complex boundary, fewer support vectors.
- Small γ: a smooth boundary, a robust model, many support vectors.
- γ usually set by crossvalidation.

Optimal Separating Hyperplane (nonseparble case)

Again, we omit replace the condition $\|\beta\|$ by defining $M = \frac{1}{\|\beta\|}$ and optimize

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

We replace the constant by a multiplicative parameter γ and solve

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

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

- We can set $\gamma = \infty$ for the separable case.
- Large γ: a complex boundary, fewer support vectors.
- Small γ: a smooth boundary, a robust model, many support vectors.
- γ usually set by crossvalidation.

Optimal Separating Hyperplane (nonseparble case)

Again, we omit replace the condition $\|\beta\|$ by defining $M = \frac{1}{\|\beta\|}$ and optimize

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

We replace the constant by a multiplicative parameter γ and solve

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

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

- We can set $\gamma = \infty$ for the separable case.
- Large γ: a complex boundary, fewer support vectors.
- Small γ: a smooth boundary, a robust model, many support vectors.
- γ usually set by crossvalidation.

Optimal Separating Hyperplane (nonseparble case)

Again, we omit replace the condition $\|\beta\|$ by defining $M = \frac{1}{\|\beta\|}$ and optimize

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

We replace the constant by a multiplicative parameter γ and solve

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

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

- We can set $\gamma = \infty$ for the separable case.
- Large γ: a complex boundary, fewer support vectors.
- Small γ: a smooth boundary, a robust model, many support vectors.
- γ usually set by crossvalidation.

We solve

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

subject to $\xi_i \geq 0$ and $y_i(x_i^T\beta + \beta_0) \geq (1 - \xi_i)$.

Lagrange multipliers again for α_i, μ_i:

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

Setting the derivative $= 0$ we get:

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

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

$$\alpha_i = \gamma - \mu_i.$$

We solve

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

subject to $\xi_i \geq 0$ and $y_i(x_i^T\beta + \beta_0) \geq (1 - \xi_i)$.
Lagrange multipliers again for α_i, μ_i:

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

Setting the derivative = 0 we get:

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

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

$$\alpha_i = \gamma - \mu_i.$$

We solve

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

subject to $\xi_i \geq 0$ and $y_i(x_i^T\beta + \beta_0) \geq (1 - \xi_i)$.
Lagrange multipliers again for α_i, μ_i:

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

Setting the derivative $= 0$ we get:

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

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

$$\alpha_i = \gamma - \mu_i.$$

Substitute to get Wolfe dual:

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

and maximize L_D subject to $0 \leq \alpha_i \leq \gamma$ a $\sum_{i=1}^{N} \alpha_i y_i = 0$.
Solution satisfies:

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

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha_i}$ are
  - points at the boundary
    - $\widehat{\xi_i} = 0$ (therefore $0 < \widehat{\alpha_i} < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi_i} > 0$ (and $\widehat{\alpha_i} = \gamma$).
- Any point with $\widehat{\xi_i} = 0$ can be used to calculate $\widehat{\beta_0}$, typically an average.
  - $\widehat{\beta_0}$ for a boundary point $\alpha_i > 0$, $\xi_i = 0$:

$$\alpha_i \left[y_i (x^T \widehat{\beta} + \widehat{\beta_0}) - (1 - 0) \right] = 0$$

- Parameter α settled by tuning (crossvalidation)

Substitute to get Wolfe dual:

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

and maximize L_D subject to $0 \leq \alpha_i \leq \gamma$ a $\sum_{i=1}^{N} \alpha_i y_i = 0$.

Solution satisfies:

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

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha_i}$ are
  - points at the boundary
    - $\widehat{\xi_i} = 0$ (therefore $0 < \widehat{\alpha_i} < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi_i} > 0$ (and $\widehat{\alpha_i} = \gamma$).
- Any point with $\widehat{\xi_i} = 0$ can be used to calculate $\widehat{\beta_0}$, typically an average.
  - $\widehat{\beta_0}$ for a boundary point $\alpha_i > 0$, $\xi_i = 0$:

$$\alpha_i \left[y_i (x^T \widehat{\beta} + \widehat{\beta_0}) - (1 - 0) \right] = 0$$

- Parameter γ settled by tuning (crossvalidation)

Substitute to get Wolfe dual:

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

and maximize L_D subject to $0 \leq \alpha_i \leq \gamma$ a $\sum_{i=1}^{N} \alpha_i y_i = 0$.
Solution satisfies:

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

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha_i}$ are
  - points at the boundary
    - $\widehat{\xi_i} = 0$ (therefore $0 < \widehat{\alpha_i} < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi_i} > 0$ (and $\widehat{\alpha_i} = \gamma$).
- Any point with $\widehat{\xi_i} = 0$ can be used to calculate $\widehat{\beta_0}$, typically an average.
  - $\widehat{\beta_0}$ for a boundary point $\alpha_i > 0$, $\xi_i = 0$:

$$\alpha_i \left[y_i (x^T \widehat{\beta} + \widehat{\beta_0}) - (1 - 0) \right] = 0$$

- Parameter γ settled by tuning (crossvalidation)

Substitute to get Wolfe dual:

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

and maximize L_D subject to $0 \leq \alpha_i \leq \gamma$ a $\sum_{i=1}^{N} \alpha_i y_i = 0$.
Solution satisfies:

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

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha_i}$ are
  - points at the boundary
    - $\widehat{\xi_i} = 0$ (therefore $0 < \widehat{\alpha_i} < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi_i} > 0$ (and $\widehat{\alpha_i} = \gamma$).
- Any point with $\widehat{\xi_i} = 0$ can be used to calculate $\widehat{\beta_0}$, typically an average.
  - $\widehat{\beta_0}$ for a boundary point $\alpha_i > 0$, $\xi_i = 0$:

$$\alpha_i \left[y_i (x^T \widehat{\beta} + \widehat{\beta_0}) - (1 - 0) \right] = 0$$

- Parameter γ settled by tuning (crossvalidation)

Substitute to get Wolfe dual:

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

and maximize L_D subject to $0 \leq \alpha_i \leq \gamma$ a $\sum_{i=1}^{N} \alpha_i y_i = 0$.
Solution satisfies:

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

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha_i}$ are
  - points at the boundary
    - $\widehat{\xi_i} = 0$ (therefore $0 < \widehat{\alpha_i} < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi_i} > 0$ (and $\widehat{\alpha_i} = \gamma$).
- Any point with $\widehat{\xi_i} = 0$ can be used to calculate $\widehat{\beta_0}$, typically an average.
  - $\widehat{\beta_0}$ for a boundary point $\alpha_i > 0$, $\xi_i = 0$:

$$\alpha_i \left[y_i (x^T \widehat{\beta} + \widehat{\beta_0}) - (1 - 0) \right] = 0$$

- Parameter γ settled by tuning (crossvalidation)

Substitute to get Wolfe dual:

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

and maximize L_D subject to $0 \leq \alpha_i \leq \gamma$ a $\sum_{i=1}^{N} \alpha_i y_i = 0$.
Solution satisfies:

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

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha_i}$ are
  - points at the boundary
    - $\widehat{\xi_i} = 0$ (therefore $0 < \widehat{\alpha_i} < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi_i} > 0$ (and $\widehat{\alpha_i} = \gamma$).
- Any point with $\widehat{\xi_i} = 0$ can be used to calculate $\widehat{\beta_0}$, typically an average.
  - $\widehat{\beta_0}$ for a boundary point $\alpha_i > 0$, $\xi_i = 0$:

$$\alpha_i \left[y_i (x^T \widehat{\beta} + \widehat{\beta_0}) - (1 - 0) \right] = 0$$

- Parameter γ settled by tuning (crossvalidation)

Substitute to get Wolfe dual:

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

and maximize L_D subject to $0 \leq \alpha_i \leq \gamma$ a $\sum_{i=1}^{N} \alpha_i y_i = 0$.
Solution satisfies:

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

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha_i}$ are
  - points at the boundary
    - $\widehat{\xi_i} = 0$ (therefore $0 < \widehat{\alpha_i} < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi_i} > 0$ (and $\widehat{\alpha_i} = \gamma$).
- Any point with $\widehat{\xi_i} = 0$ can be used to calculate $\widehat{\beta_0}$, typically an average.
  - $\widehat{\beta_0}$ for a boundary point $\alpha_i > 0$, $\xi_i = 0$:

$$\alpha_i \left[y_i (x^T \widehat{\beta} + \widehat{\beta_0}) - (1 - 0) \right] = 0$$

- Parameter γ settled by tuning (crossvalidation)

Substitute to get Wolfe dual:

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

and maximize L_D subject to $0 \leq \alpha_i \leq \gamma$ a $\sum_{i=1}^{N} \alpha_i y_i = 0$.
Solution satisfies:

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

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha_i}$ are
  - points at the boundary
    - $\widehat{\xi_i} = 0$ (therefore $0 < \widehat{\alpha_i} < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi_i} > 0$ (and $\widehat{\alpha_i} = \gamma$).
- Any point with $\widehat{\xi_i} = 0$ can be used to calculate $\widehat{\beta_0}$, typically an average.
  - $\widehat{\beta_0}$ for a boundary point $\alpha_i > 0$, $\xi_i = 0$:

$$\alpha_i \left[y_i (x^T \widehat{\beta} + \widehat{\beta_0}) - (1 - 0) \right] = 0$$

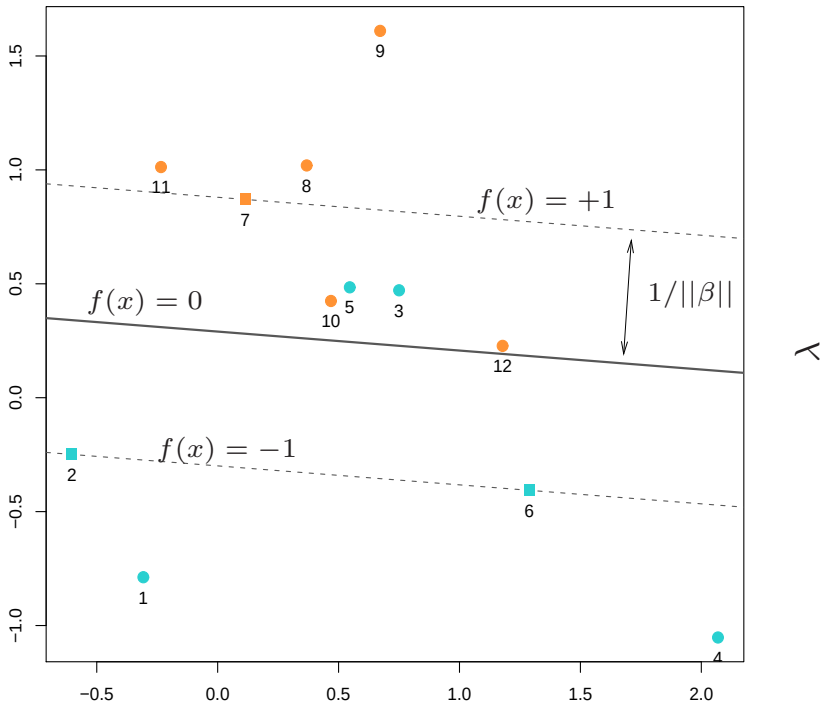
- Parameter γ settled by tuning (crossvalidation)

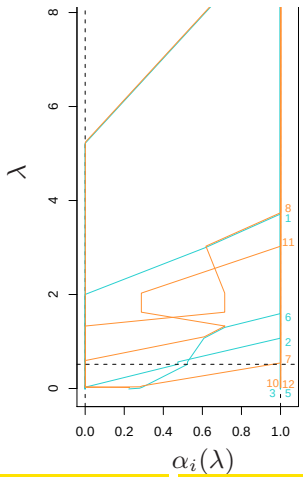
SVM Solution

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha}_i$ are
  - points at the boundary
  - and points on the wrong side of the margin
- Any point with $\widehat{\xi}_i = 0$ can be used to calculate $\widehat{\beta}_0$, typically an average.
  - $\widehat{\beta}_0$ for a boundary point ($\xi_i = 0$):

$$\alpha_i \left[y_i(x^T \widehat{\beta} + \widehat{\beta}_0) - (1 - 0) \right] = 0$$

- $\alpha = \xi = 0$ for points 1,4,8,9,11
- $\alpha > 0, \xi = 0$ for points 2,6,8
- missclassified points 3,5

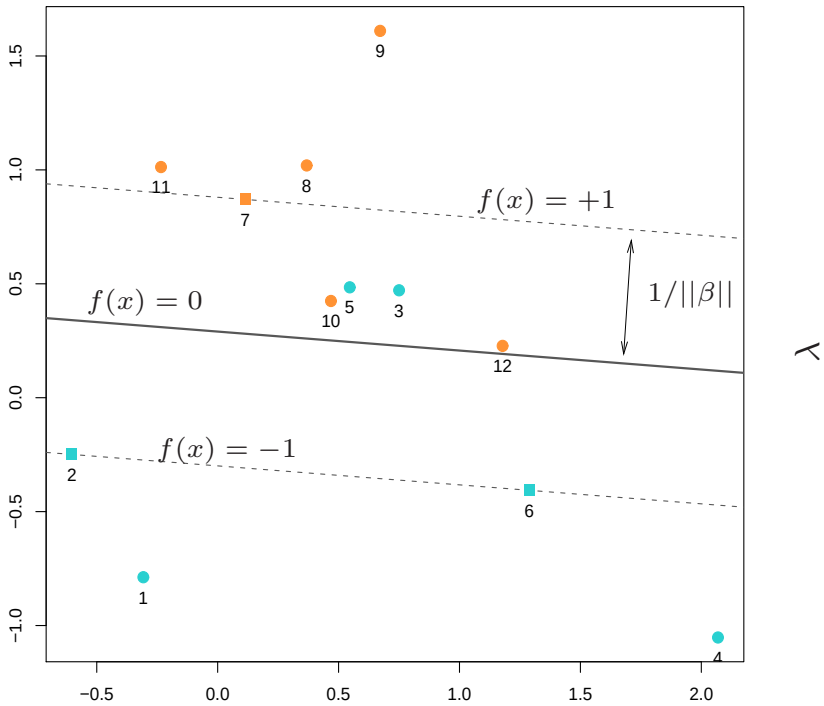


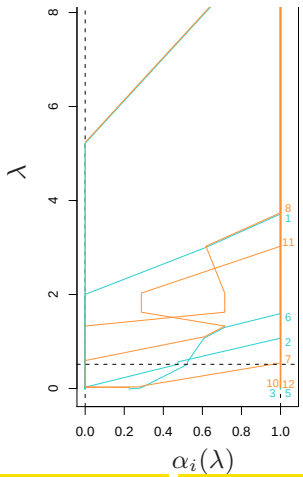


SVM Solution

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha}_i$ are
  - points at the boundary
    - $\widehat{\xi}_i = 0$ (therefore $0 < \widehat{\alpha}_i < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi}_i > 0$ (and $\widehat{\alpha}_i = \gamma$).
- Any point with $\widehat{\xi}_i = 0$ can be used to calculate $\widehat{\beta}_0$, typically an average.
  - $\widehat{\beta}_0$ for a boundary point ($\xi_i = 0$):

$$\alpha_i \left[y_i (x^T \widehat{\beta} + \widehat{\beta}_0) - (1 - 0) \right] = 0$$





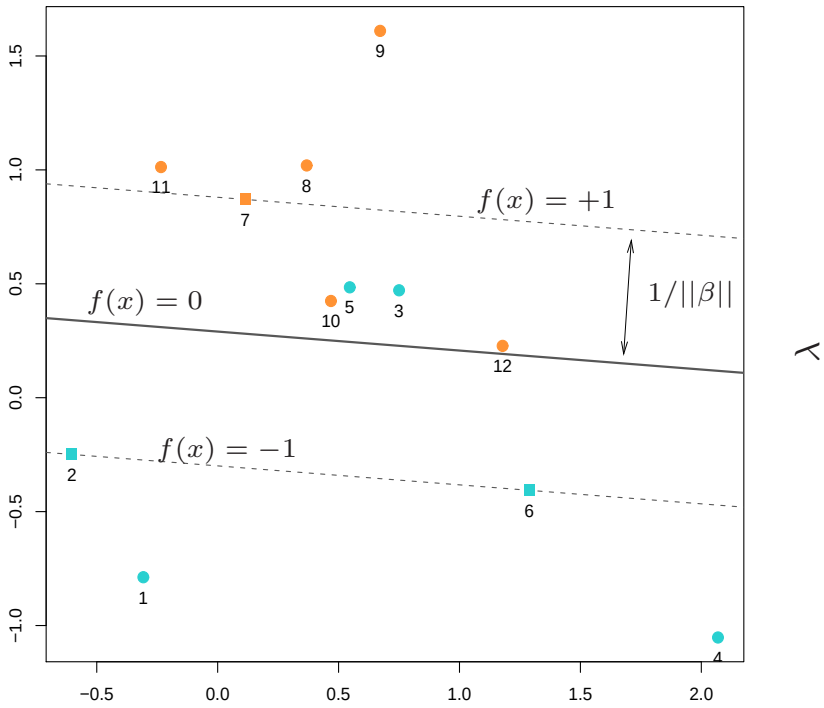
- $\alpha = \xi = 0$ for points 1,4,8,9,11
- $\alpha > 0, \xi = 0$ for points 2,6,8
- missclassified points 3,5,

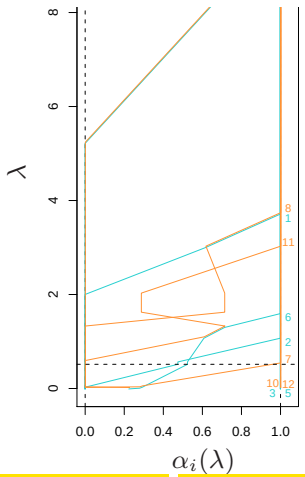
SVM Solution

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha}_i$ are
  - points at the boundary
    - $\widehat{\xi}_i = 0$ (therefore $0 < \widehat{\alpha}_i < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi}_i > 0$ (and $\widehat{\alpha}_i = \gamma$).
- Any point with $\widehat{\xi}_i = 0$ can be used to calculate $\widehat{\beta}_0$, typically an average.
  - $\widehat{\beta}_0$ for a boundary point ($\xi_i = 0$):

$$\alpha_i \left[y_i (x^T \widehat{\beta} + \widehat{\beta}_0) - (1 - 0) \right] = 0$$

- $\alpha = \xi = 0$ for points 1,4,8,9,11
- $\alpha > 0, \xi = 0$ for points 2,6,8
- missclassified points 3,5



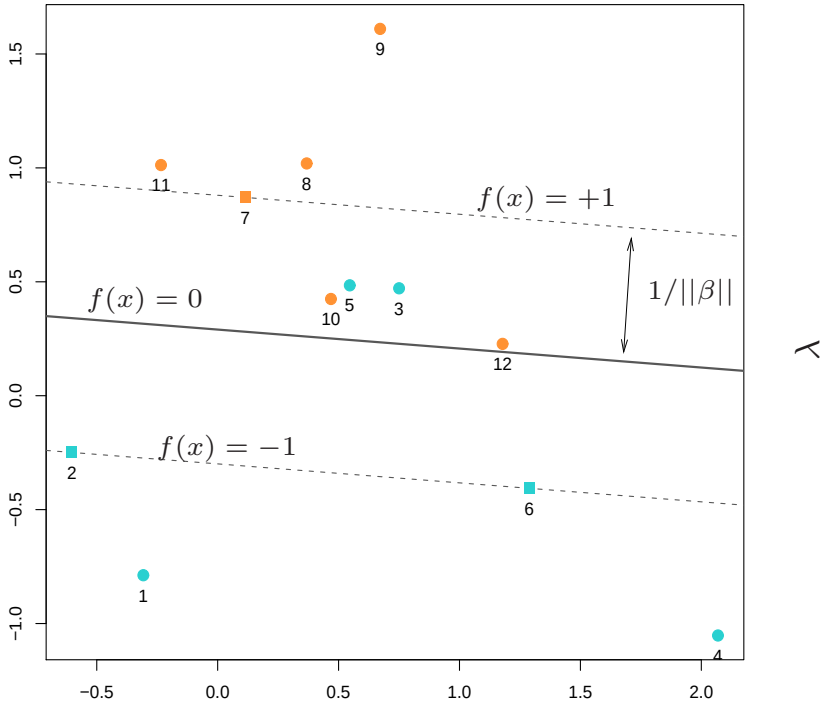


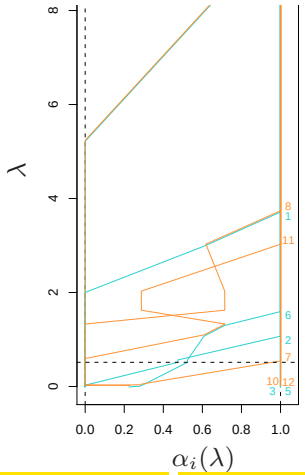
SVM Solution

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha}_i$ are
  - points at the boundary
    - $\widehat{\xi}_i = 0$ (therefore $0 < \widehat{\alpha}_i < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi}_i > 0$ (and $\widehat{\alpha}_i = \gamma$).
- Any point with $\widehat{\xi}_i = 0$ can be used to calculate $\widehat{\beta}_0$, typically an average.
  - $\widehat{\beta}_0$ for a boundary point ($\xi_i = 0$):

$$\alpha_i \left[y_i (x^T \widehat{\beta} + \widehat{\beta}_0) - (1 - 0) \right] = 0$$

- $\alpha = \xi = 0$ for points 1,4,8,9,11
- $\alpha > 0, \xi = 0$ for points 2,6,8
- missclassified points 3,5.



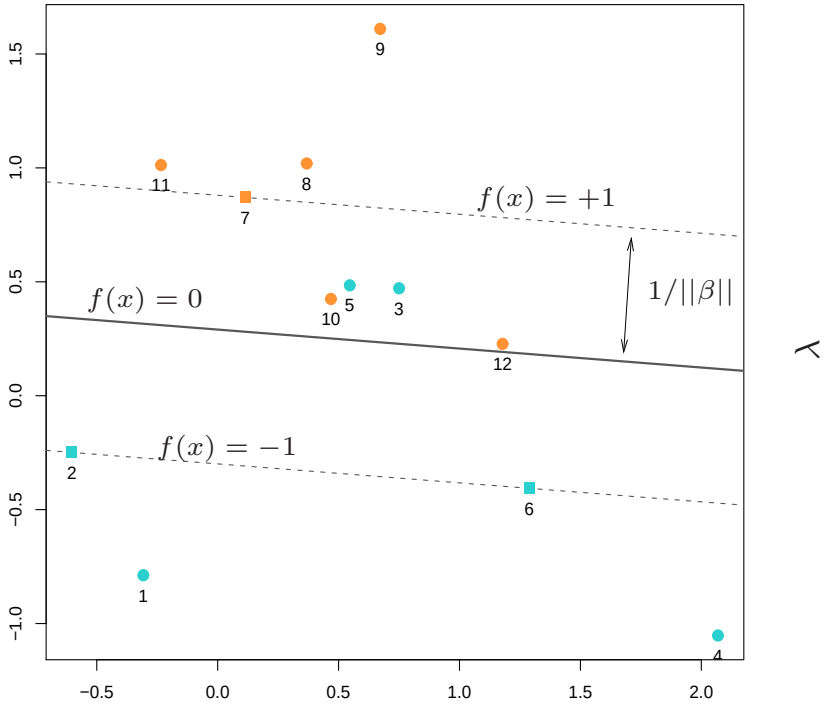


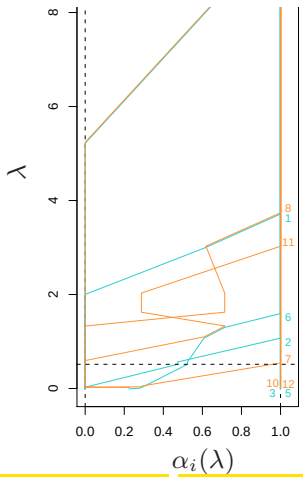
SVM Solution

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha}_i$ are
  - points at the boundary
    - $\widehat{\xi}_i = 0$ (therefore $0 < \widehat{\alpha}_i < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi}_i > 0$ (and $\widehat{\alpha}_i = \gamma$).
- Any point with $\widehat{\xi}_i = 0$ can be used to calculate $\widehat{\beta}_0$, typically an average.
- $\widehat{\beta}_0$ for a boundary point i results in

$$\alpha_i \left[y_i(x^T \widehat{\beta} + \widehat{\beta}_0) - (1 - 0) \right] = 0$$

- $\alpha = \xi = 0$ for points 1,4,8,9,11
- $\alpha > 0, \xi = 0$ for points 2,6,8
- missclassified points 3,5



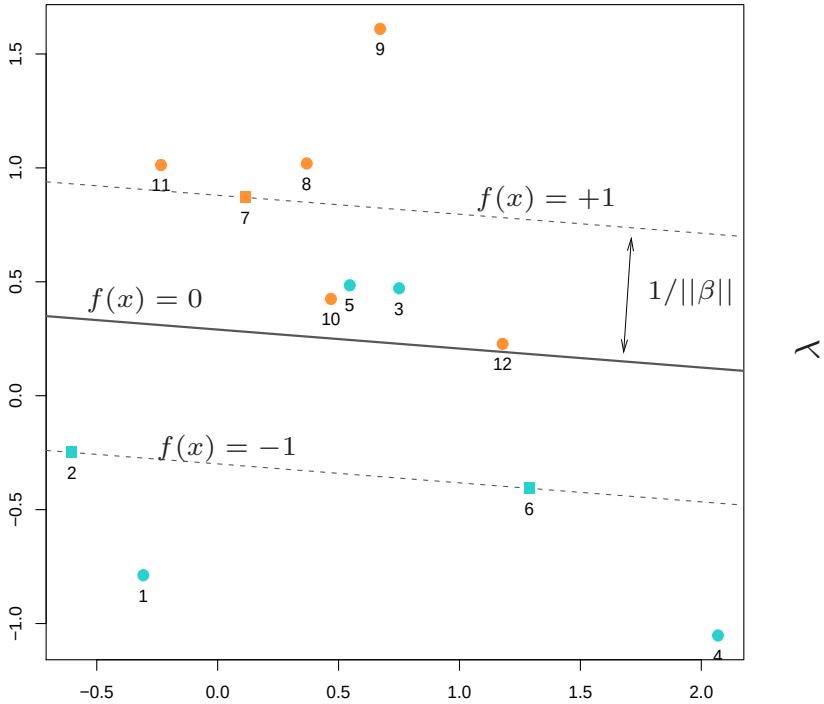


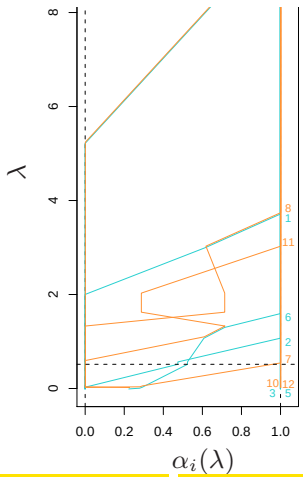
SVM Solution

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha}_i$ are
  - points at the boundary
    - $\widehat{\xi}_i = 0$ (therefore $0 < \widehat{\alpha}_i < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi}_i > 0$ (and $\widehat{\alpha}_i = \gamma$).
- Any point with $\widehat{\xi}_i = 0$ can be used to calculate $\widehat{\beta}_0$, typically an average.
  - $\widehat{\beta}_0$ for a boundary point $\xi_i = 0$:

$$\alpha_i \left[y_i (x^T \widehat{\beta} + \widehat{\beta}_0) - (1 - 0) \right] = 0$$

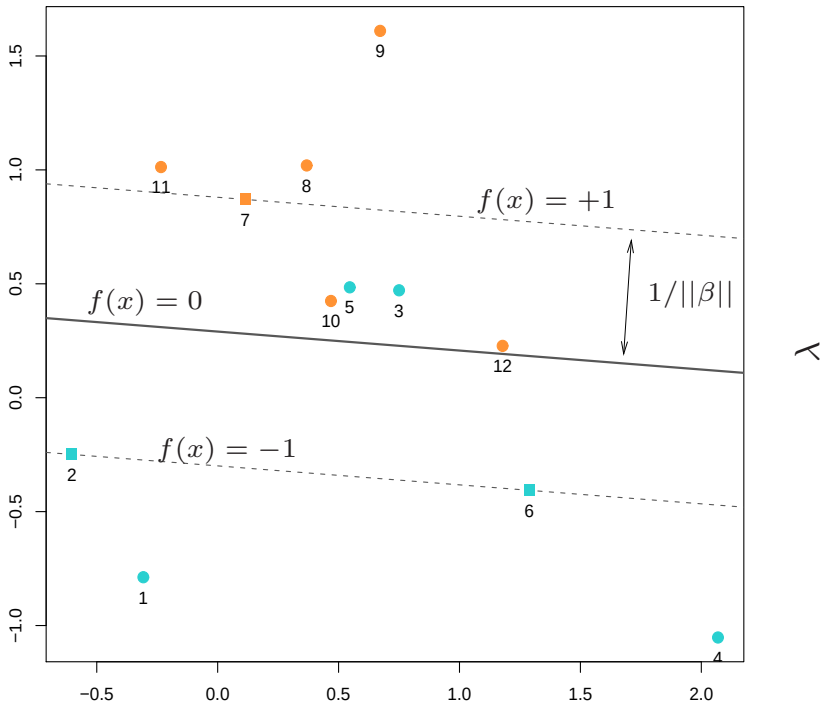
- $\alpha = \xi = 0$ for points 1,4,8,9,11
- $\alpha > 0, \xi = 0$ for points 2,6,8
- missclassified points 3,5.





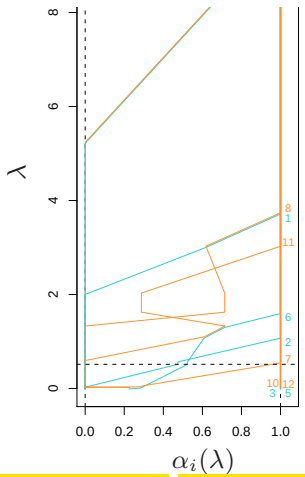
SVM Solution

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha}_i$ are
  - points at the boundary
    - $\widehat{\xi}_i = 0$ (therefore $0 < \widehat{\alpha}_i < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi}_i > 0$ (and $\widehat{\alpha}_i = \gamma$).
- Any point with $\widehat{\xi}_i = 0$ can be used to calculate $\widehat{\beta}_0$, typically an average.



- $\widehat{\beta}_0$ for a boundary point $\xi_i = 0$:

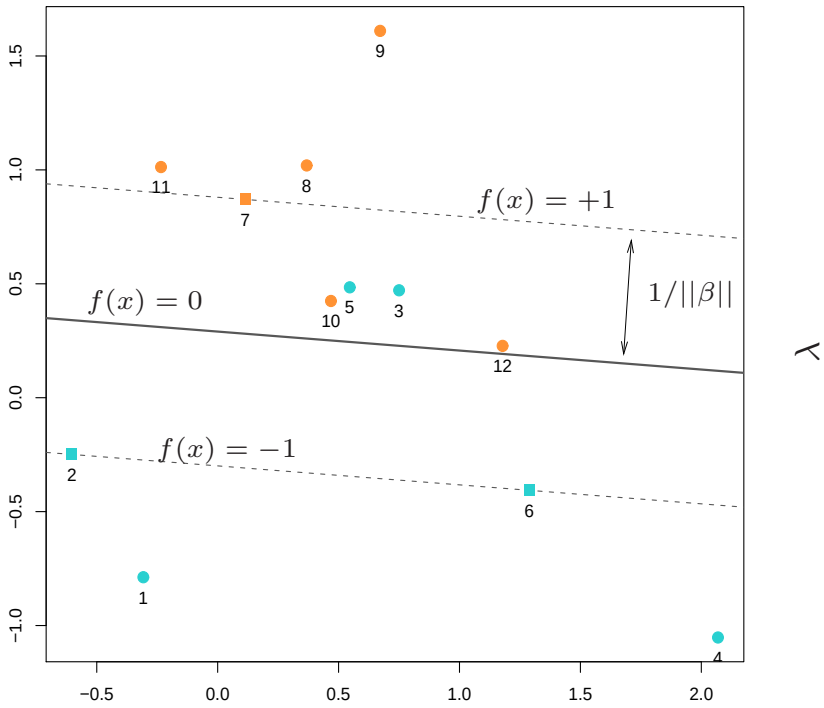
$$\alpha_i \left[y_i(x^T \widehat{\beta} + \widehat{\beta}_0) - (1 - 0) \right] = 0$$



- $\alpha = \xi = 0$ for points 1,4,8,9,11
- $\alpha > 0, \xi = 0$ for points 2,6,8
- missclassified points 3,5

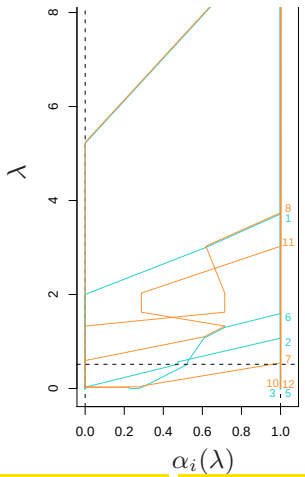
SVM Solution

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha}_i$ are
  - points at the boundary
    - $\widehat{\xi}_i = 0$ (therefore $0 < \widehat{\alpha}_i < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi}_i > 0$ (and $\widehat{\alpha}_i = \gamma$).
- Any point with $\widehat{\xi}_i = 0$ can be used to calculate $\widehat{\beta}_0$, typically an average.



  - $\widehat{\beta}_0$ for a boundary point $\xi_i = 0$:

$$\alpha_i \left[y_i(x^T \widehat{\beta} + \widehat{\beta}_0) - (1 - 0) \right] = 0$$

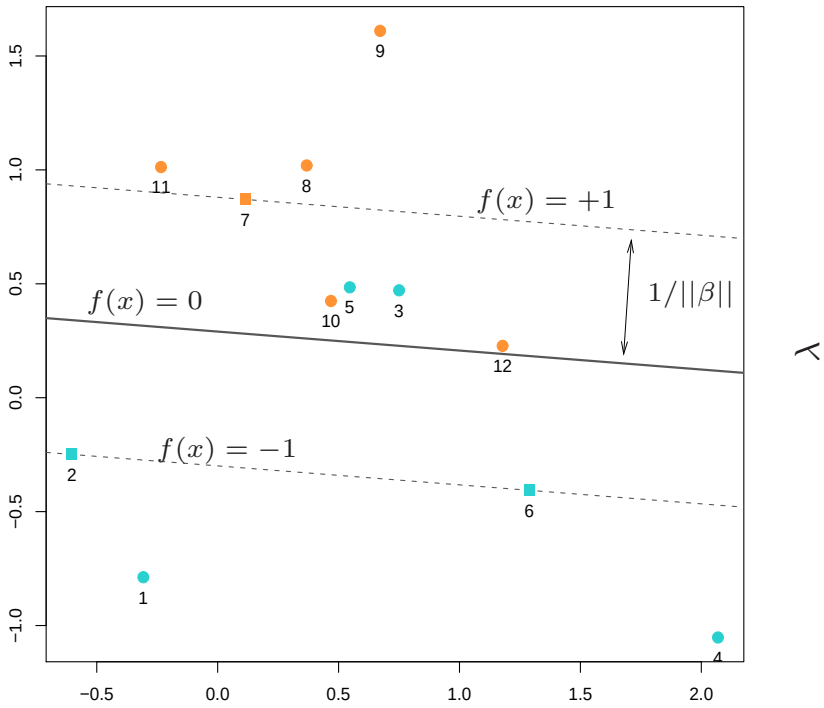


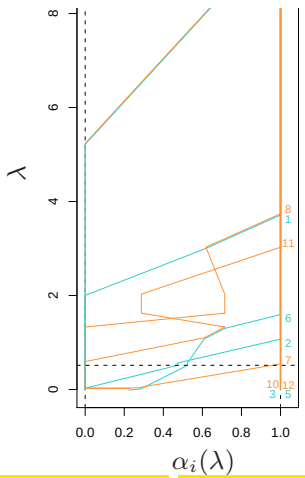
- $\alpha = \xi = 0$ for points 1,4,8,9,11
- $\alpha > 0, \xi = 0$ for points 2,6,8
- missclassified points 3,5

SVM Solution

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha}_i$ are
  - points at the boundary
    - $\widehat{\xi}_i = 0$ (therefore $0 < \widehat{\alpha}_i < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi}_i > 0$ (and $\widehat{\alpha}_i = \gamma$).
- Any point with $\widehat{\xi}_i = 0$ can be used to calculate $\widehat{\beta}_0$, typically an average.
  - $\widehat{\beta}_0$ for a boundary point $\xi_i = 0$:

$$\alpha_i \left[y_i (x^T \widehat{\beta} + \widehat{\beta}_0) - (1 - 0) \right] = 0$$



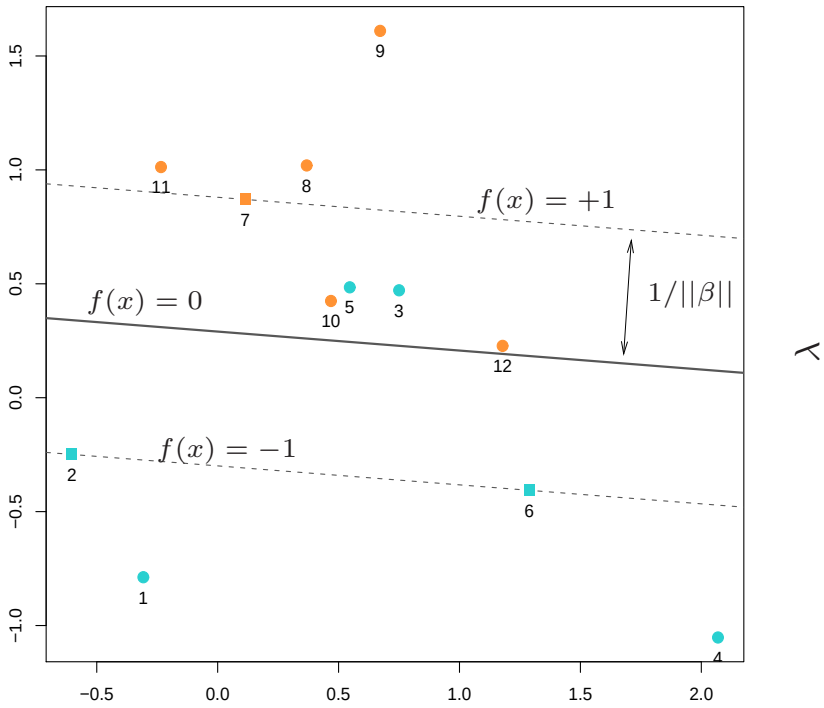


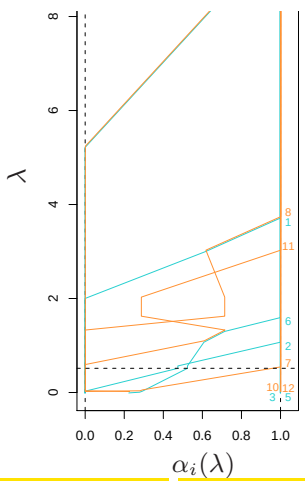
- $\alpha = \xi = 0$ for points 1,4,8,9,11
- $\alpha > 0, \xi = 0$ for points 2,6,8
- missclassified points 3,5,

SVM Solution

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha}_i$ are
  - points at the boundary
    - $\widehat{\xi}_i = 0$ (therefore $0 < \widehat{\alpha}_i < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi}_i > 0$ (and $\widehat{\alpha}_i = \gamma$).
- Any point with $\widehat{\xi}_i = 0$ can be used to calculate $\widehat{\beta}_0$, typically an average.
  - $\widehat{\beta}_0$ for a boundary point $\xi_i = 0$:

$$\alpha_i \left[y_i (x^T \widehat{\beta} + \widehat{\beta}_0) - (1 - 0) \right] = 0$$



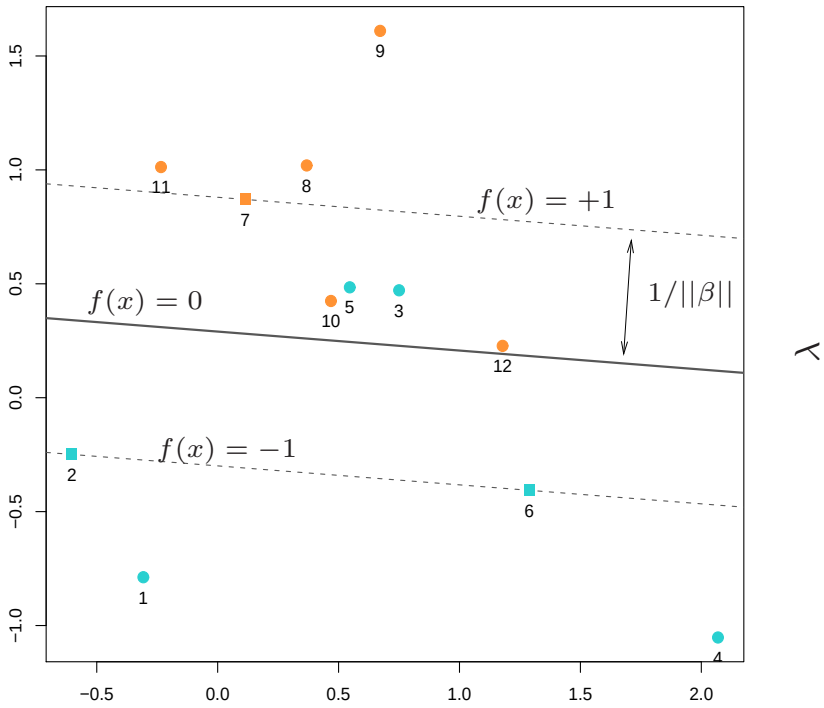


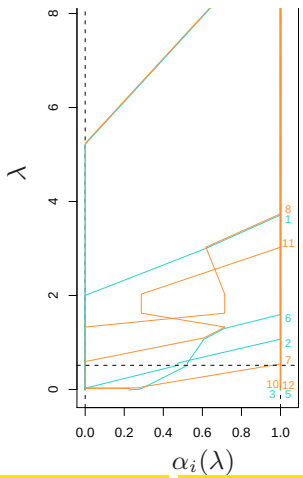
- $\alpha = \xi = 0$ for points 1,4,8,9,11
- $\alpha > 0, \xi = 0$ for points 2,6,8
- missclassified points 3,5,

SVM Solution

- The solution is $\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$.
- **support points** with nonzero coefficients $\widehat{\alpha}_i$ are
  - points at the boundary
    - $\widehat{\xi}_i = 0$ (therefore $0 < \widehat{\alpha}_i < \gamma$),
  - and points on the wrong side of the margin
    - $\widehat{\xi}_i > 0$ (and $\widehat{\alpha}_i = \gamma$).
- Any point with $\widehat{\xi}_i = 0$ can be used to calculate $\widehat{\beta}_0$, typically an average.
  - $\widehat{\beta}_0$ for a boundary point $\xi_i = 0$:

$$\alpha_i \left[y_i (x^T \widehat{\beta} + \widehat{\beta}_0) - (1 - 0) \right] = 0$$





- $\alpha = \xi = 0$ for points 1,4,8,9,11
- $\alpha > 0, \xi = 0$ for points 2,6,8
- missclassified points 3,5.

Support Vector Machines

Let us have the training data $(x_i, y_i)_{i=1}^N$, $x_i \in \mathbb{R}^p$, y_i in $\{-1, 1\}$. We define a hyperplane

$$\{x : f(x) = x^T \beta + \beta_0 = 0\} \tag{10}$$

where $\|\beta\| = 1$.
We classify according to

$$G(x) = sign\left[x^T \beta + \beta_0\right]$$

where $f(x)$ is a signed distance of x from the hyperplane.
Support vector machines replace the scalar product $\langle x_i, x \rangle$ by a kernel function.

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

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

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

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

Support Vector Machines

Let us have the training data $(x_i, y_i)_{i=1}^N$, $x_i \in \mathbb{R}^p$, y_i in $\{-1, 1\}$. We define a hyperplane

$$\{x : f(x) = x^T \beta + \beta_0 = 0\} \tag{10}$$

where $\|\beta\| = 1$.
We classify according to

$$G(x) = sign\left[x^T \beta + \beta_0\right]$$

where $f(x)$ is a signed distance of x from the hyperplane.
Support vector machines replace the scalar product $\langle x_i, x \rangle$ by a kernel function.

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

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

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

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

Support Vector Machines

Let us have the training data $(x_i, y_i)_{i=1}^N$, $x_i \in \mathbb{R}^p$, y_i in $\{-1, 1\}$. We define a hyperplane

$$\{x : f(x) = x^T \beta + \beta_0 = 0\} \tag{10}$$

where $\|\beta\| = 1$.
We classify according to

$$G(x) = sign\left[x^T \beta + \beta_0\right]$$

where $f(x)$ is a signed distance of x from the hyperplane.
Support vector machines replace the scalar product $\langle x_i, x \rangle$ by a kernel function.

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

Support Vector Machines

Let us have the training data $(x_i, y_i)_{i=1}^N$, $x_i \in \mathbb{R}^p$, y_i in $\{-1, 1\}$. We define a hyperplane

$$\{x : f(x) = x^T \beta + \beta_0 = 0\} \tag{10}$$

where $\|\beta\| = 1$.
We classify according to

$$G(x) = sign\left[x^T \beta + \beta_0\right]$$

where $f(x)$ is a signed distance of x from the hyperplane.
Support vector machines replace the scalar product $\langle x_i, x \rangle$ by a kernel function.

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

Support Vector Machines

Let us have the training data $(x_i, y_i)_{i=1}^N$, $x_i \in \mathbb{R}^p$, y_i in $\{-1, 1\}$. We define a hyperplane

$$\{x : f(x) = x^T \beta + \beta_0 = 0\} \tag{10}$$

where $\|\beta\| = 1$.
We classify according to

$$G(x) = sign\left[x^T \beta + \beta_0\right]$$

where $f(x)$ is a signed distance of x from the hyperplane.
Support vector machines replace the scalar product $\langle x_i, x \rangle$ by a kernel function.

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

Support Vector Machines

Let us have the training data $(x_i, y_i)_{i=1}^N$, $x_i \in \mathbb{R}^p$, y_i in $\{-1, 1\}$. We define a hyperplane

$$\{x : f(x) = x^T \beta + \beta_0 = 0\} \tag{10}$$

where $\|\beta\| = 1$.
We classify according to

$$G(x) = sign\left[x^T \beta + \beta_0\right]$$

where $f(x)$ is a signed distance of x from the hyperplane.
Support vector machines replace the scalar product $\langle x_i, x \rangle$ by a kernel function.

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

SVM Example

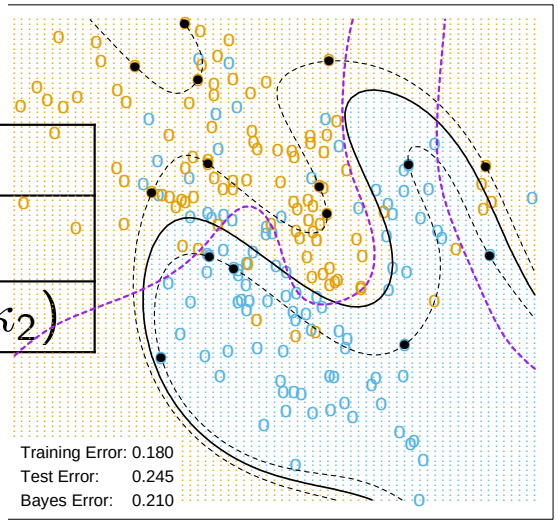
- **kernel functions** are function to replace scalar product with a scalar product in a transformed space.

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

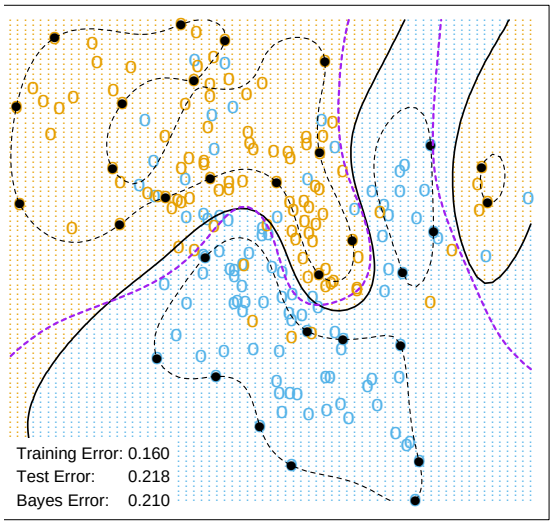
- For example a degree 2 with two dimensional input: $K(x, x') = (1 + \langle x, x' \rangle)^2 = (1 + 2x_1x_1' + 2x_2x_2' + (x_1x_1')^2 + (x_2x_2')^2 + 2x_1x_1'x_2x_2')$
- that is $M = 6$, $h_1(x) = 1$, $h_2(x) = \sqrt{2}x_1$, $h_3(x) = \sqrt{2}x_2$, $h_4(x) = x_1^2$, $h_5(x) = x_2^2$, $h_6(x) = \sqrt{2}x_1x_2$.

The classification function $\hat{f}(x) = h(x)^T\beta + \beta_0 = \sum_{i=1}^{N} \alpha_i y_i \langle h(x), h(x_i) \rangle + \beta_0$ does not need evaluation of $h(i)$, only the scalar product $\langle h(x), h(x_i) \rangle$.

SVM - Degree-4 Polynomial in Feature Space



SVM - Radial Kernel in Feature Space



SVM Example

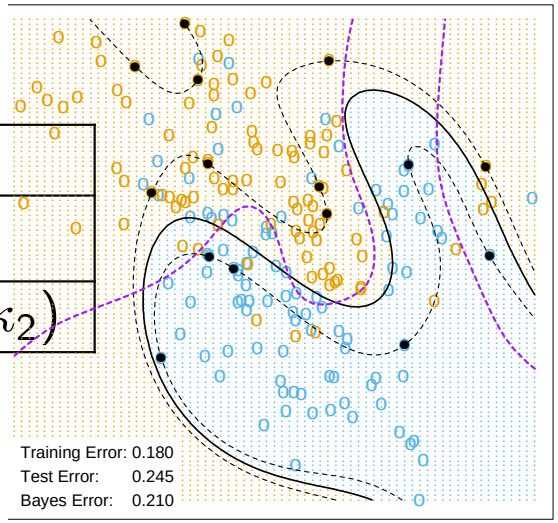
- **kernel functions** are function to replace scalar product with a scalar product in a transformed space.

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

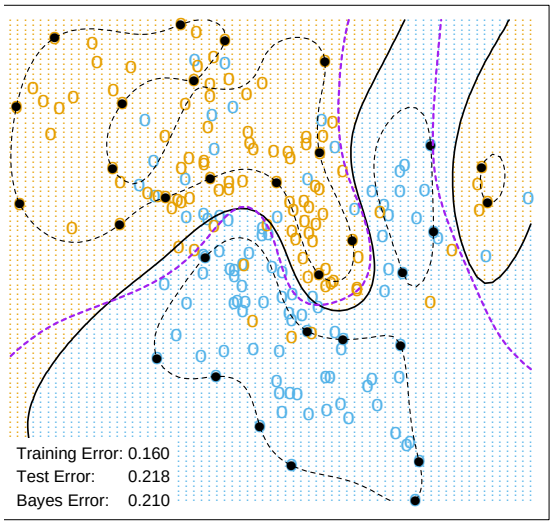
- For example a degree 2 with two dimensional input: $K(x, x') = (1 + \langle x, x' \rangle)^2 = (1 + 2x_1x_1' + 2x_2x_2' + (x_1x_1')^2 + (x_2x_2')^2 + 2x_1x_1'x_2x_2')$
- that is $M = 6$, $h_1(x) = 1$, $h_2(x) = \sqrt{2}x_1$, $h_3(x) = \sqrt{2}x_2$, $h_4(x) = x_1^2$, $h_5(x) = x_2^2$, $h_6(x) = \sqrt{2}x_1x_2$.

The classification function $\hat{f}(x) = h(x)^T\beta + \beta_0 = \sum_{i=1}^{N} \alpha_i y_i \langle h(x), h(x_i) \rangle + \beta_0$ does not need evaluation of $h(i)$, only the scalar product $\langle h(x), h(x_i) \rangle$.

SVM - Degree-4 Polynomial in Feature Space



SVM - Radial Kernel in Feature Space



SVM Example

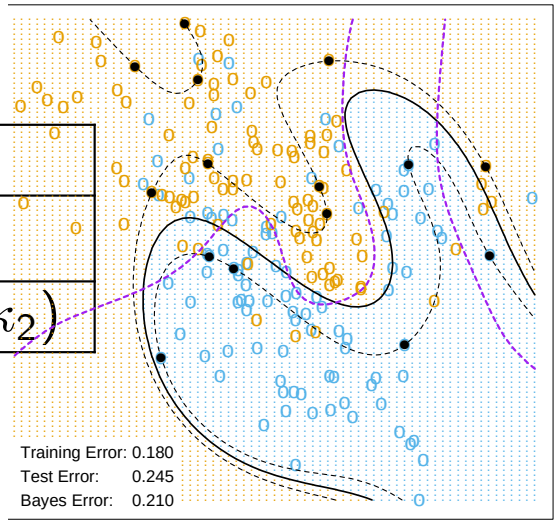
- **kernel functions** are function to replace scalar product with a scalar product in a transformed space.

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

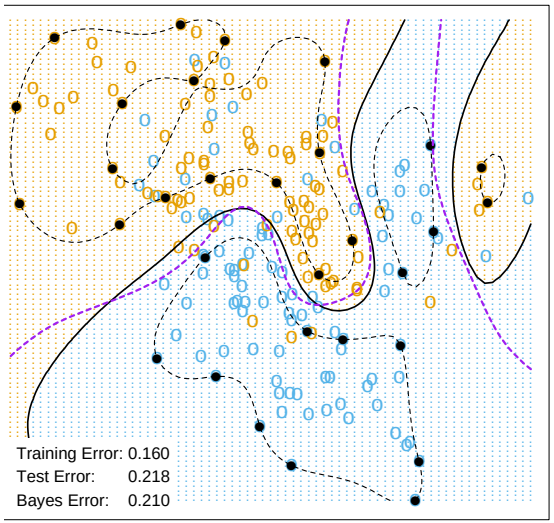
- For example a degree 2 with two dimensional input:
  $K(x, x') = (1 + \langle x, x' \rangle)^2 =$
  $(1 + 2x_1x_1' + 2x_2x_2' + (x_1x_1')^2 + (x_2x_2')^2 + 2x_1x_1'x_2x_2')$
- that is $M = 6$, $h_1(x) = 1$, $h_2(x) = \sqrt{2}x_1$, $h_3(x) = \sqrt{2}x_2$, $h_4(x) = x_1^2$, $h_5(x) = x_2^2$, $h_6(x) = \sqrt{2}x_1x_2$.

The classification function $\hat{f}(x) = h(x)^T\beta + \beta_0 = \sum_{i=1}^{N} \alpha_i y_i \langle h(x), h(x_i) \rangle + \beta_0$ does not need evaluation of $h(i)$, only the scalar product $\langle h(x), h(x_i) \rangle$.

SVM - Degree-4 Polynomial in Feature Space



SVM - Radial Kernel in Feature Space



SVM Example

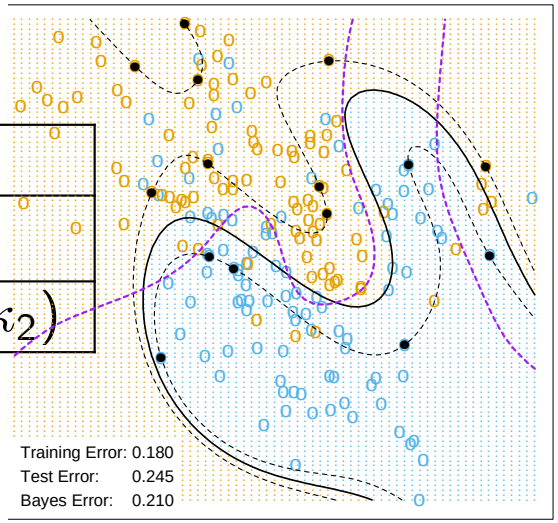
- **kernel functions** are function to replace scalar product with a scalar product in a transformed space.

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

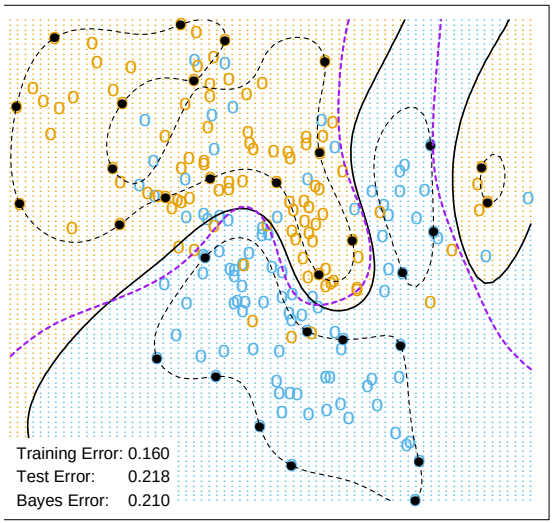
- For example a degree 2 with two dimensional input:
  $K(x, x') = (1 + \langle x, x' \rangle)^2 =$
  $(1 + 2x_1x_1' + 2x_2x_2' + (x_1x_1')^2 + (x_2x_2')^2 + 2x_1x_1'x_2x_2')$
- that is $M = 6$, $h_1(x) = 1$, $h_2(x) = \sqrt{2}x_1$,
  $h_3(x) = \sqrt{2}x_2$, $h_4(x) = x_1^2$, $h_5(x) = x_2^2$,
  $h_6(x) = \sqrt{2}x_1x_2$.

The classification function
$\hat{f}(x) = h(x)^T\beta + \beta_0 = \sum_{i=1}^{N} \alpha_i y_i \langle h(x), h(x_i) \rangle + \beta_0$
does not need evaluation of $h(i)$, only the scalar product $\langle h(x), h(x_i) \rangle$.

SVM - Degree-4 Polynomial in Feature Space



SVM - Radial Kernel in Feature Space



String Kernels and Protein Classification

IPTSALVKETLALLSTHRTLLIANETLRIPVPVHKNHQLCTEEIFQGIGTLESQTVQGGTV
ERLFKNLSLIKKYIDGQKKKCGEERRRVNQFLDY**LQE**FLGVMNTEWI

PHRRDLCSRSIWLARKIRSDLTALTESYVKHQGLWSELTEAER**LQE**NLQAYRTFHVLLA
RLLEDQQVHFTPTEGDFHQAIHTLLLQVAAFAYQIEELMILLEYKIPRNEADGMLFEKK

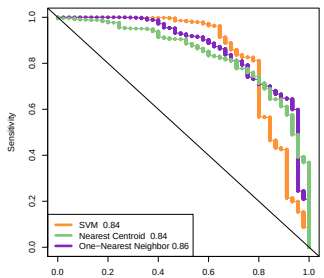
- Consider all possible sequences $a \in \mathcal{A}_m$ of length m.
- We define a feature map

$$\Phi_m(x) = \{\phi_a(x)\}_{a \in \mathcal{A}_m}$$

- The kernel function is the inner product:

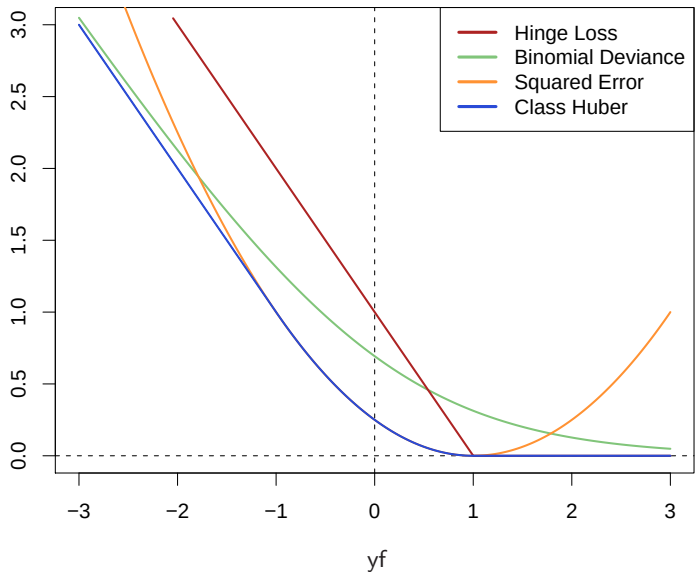
$$K_m(x_1, x_2) = \langle \Phi_m(x_1), \Phi_m(x_2) \rangle.$$

- where $\phi_a(x)$ is the number of occurence a in x.



SVM as a Penalization Method

- We fit a linear function wrt. basis $\{h_i(x)\}$: $f(x) = h^T\beta + \beta_0$.
- Consider the loss function $L(y, f) = [1 - yf]_+$
  - The optimization problem $min_{\beta_0,\beta} \sum_{i=1}^{N}[1 - yf]_+ + \lambda\|\beta\|^2$
- is equivalent to SVM
  - $\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 + \gamma\sum_{i=1}^{N}\xi_i$
  - subject to $\xi_i \geq 0$ and $y_i(x^T\beta + \beta_0) \geq (1 - \xi_i)$.
- is similar to smoothing splines penalty:
  - $\min_{\alpha,\alpha_0} \sum_{i=1}^{N}[1 - yf]_+ + \lambda\alpha^T\mathbf{K}\alpha$
  - where $\alpha^T\mathbf{K}\alpha = J(f)$ is the smoothing penalty.



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	$-4yf(x)$, $yf(x) < -1$ $[1 - yf(x)]_+^2$ otherwise	$f(x) = 2\Pr(Y=+1\|x) - 1$

SVM and Kernel Dimsension

- The first Simulated example
  - 100 observations of each class
  - First class: four standard normal independent features X_1, X_2, X_3, X_4.
  - Second class conditioned on $9 \leq \sum X_j^2 \leq 16$.
- Second example
  - The first one augmented with an additional six standard Gaussian noise features.
- BRUTTO: Additive spline model.
- BRUTTO and MARS has the ability to ignore noisy features.
- We can see the overfitting of SVM. The degree 2 polynomial kernel is the best since the decision boundary is quadratic.

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

SVM and Kernel Dimsension

- The first Simulated example
  - 100 observations of each class
  - First class: four standard normal independent features X_1, X_2, X_3, X_4.
  - Second class conditioned on $9 \leq \sum X_j^2 \leq 16$.
- Second example
  - The first one augmented with an additional six standard Gaussian noise features.
- BRUTTO: Additive spline model.
- BRUTTO and MARS has the ability to ignore noisy features.
- We can see the overfitting of SVM. The degree 2 polynomial kernel is the best since the decision boundary is quadratic.

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

SVM and Kernel Dimsension

- The first Simulated example
  - 100 observations of each class
  - First class: four standard normal independent features X_1, X_2, X_3, X_4.
  - Second class conditioned on $9 \leq \sum X_j^2 \leq 16$.
- Second example
  - The first one augmented with an additional six standard Gaussian noise features.
- BRUTTO: Additive spline model.
- BRUTTO and MARS has the ability to ignore noisy features.
- We can see the overfitting of SVM. The degree 2 polynomial kernel is the best since the decision boundary is quadratic.

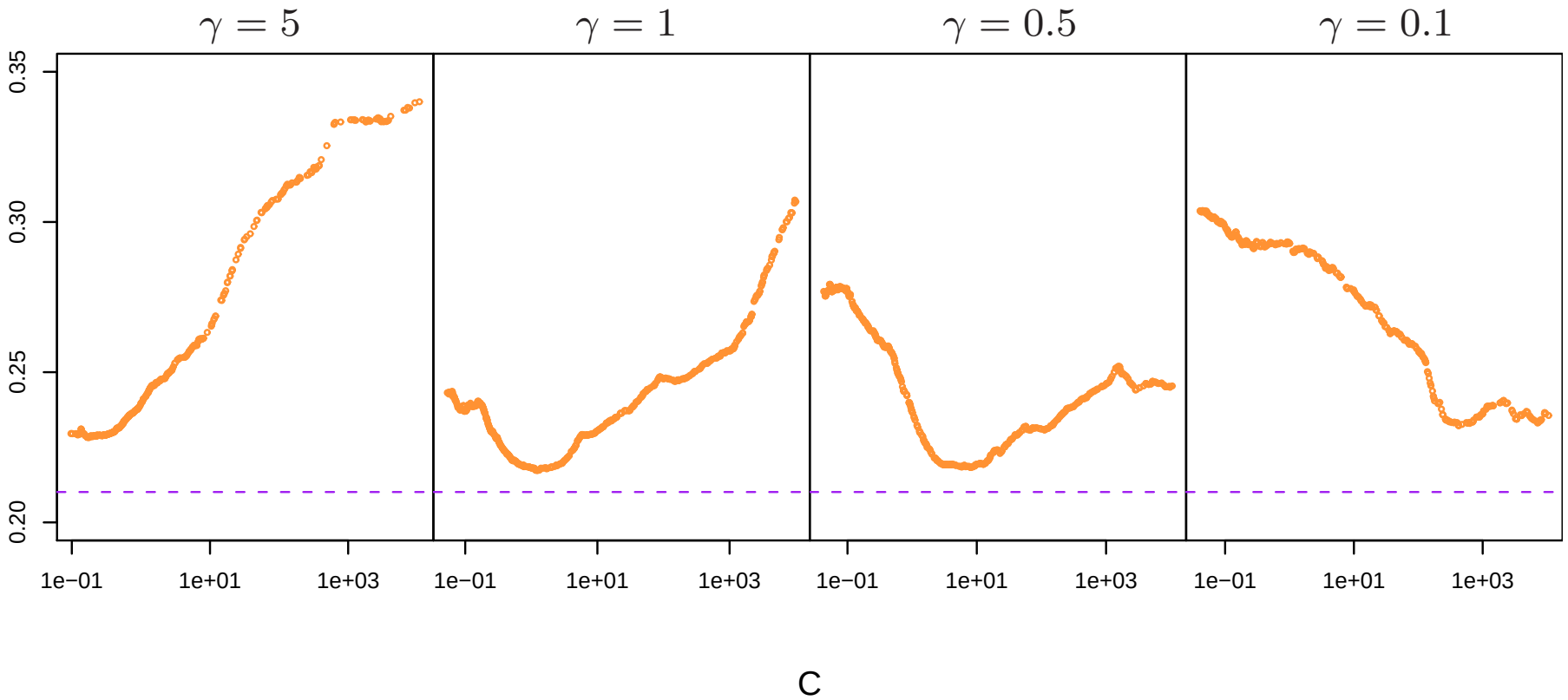
	Test Error (SE)	
Method	No Noise Features	Six Noise Features
SV Classifier	0.450 (0.003)	0.472 (0.003)
SVM/poly 2	0.078 (0.003)	0.152 (0.004)
SVM/poly 5	0.180 (0.004)	0.370 (0.004)
SVM/poly 10	0.230 (0.003)	0.434 (0.002)
BRUTO	0.084 (0.003)	0.090 (0.003)
MARS	0.156 (0.004)	0.173 (0.005)
Bayes	0.029	0.029

SVM Complexity, Kernel Parameter Tuning

SVM Complexity

The SVM complexity is $m^3 + mN + mpN$, where m is the number of support vectors.

Parameter tuning for different radial basis lengthscale γ, C is the cost penalty γ.



SVM for Regression

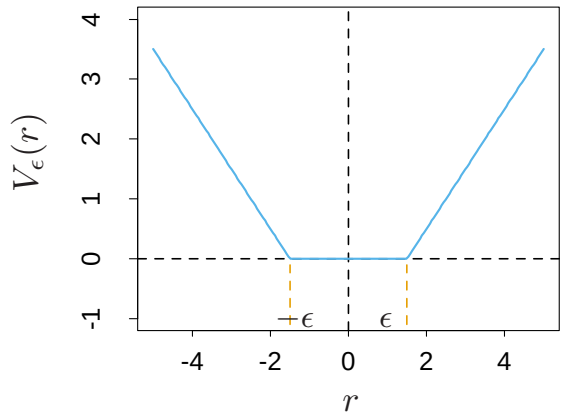
- In regression, we fit a function: $f(x) = x^T\beta + \beta_0$
- We consider error function V_ϵ (left figure)

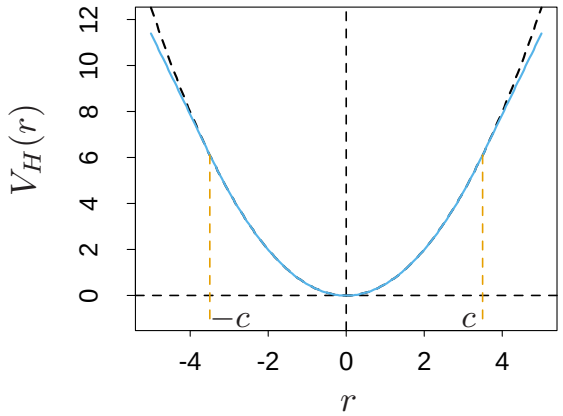
$$V_\epsilon(r) = \begin{cases} 0 & \text{if } |r| < \epsilon, \\ |r| - \epsilon, & \text{otherwise.} \end{cases}$$

- and minimize:

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

-





SVM for Regression 2

- The solution has the form: $\hat{\alpha}_i, \hat{\alpha}_i^* \geq 0$

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

- and solve the quadratic programming problem

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

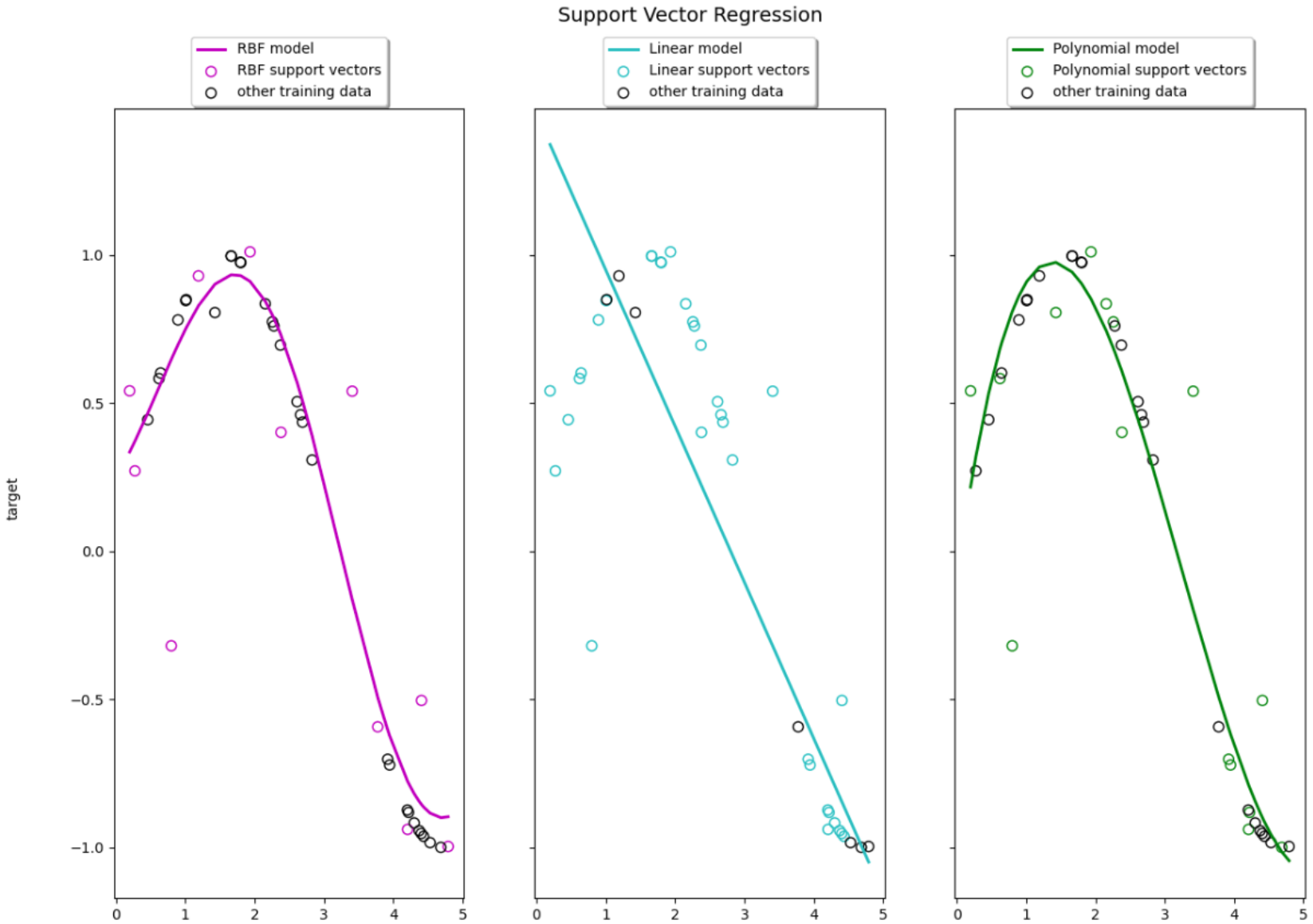
- subject to the constraints

$$\begin{aligned} 0 \leq \alpha_i,\ \alpha_i^* \leq 1/\lambda, \\ \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) = 0, \\ \alpha_i \alpha_i^* = 0. \end{aligned}$$

- Support vectors are those with nonzero $(\hat{\alpha}_i^* - \hat{\alpha}_i)$.
- With scaled response y, you may use the default ϵ.
- λ is tuned by cross-validation.

SVR

sklearn.svm.SVR



List of topics

1. Linear, ridge, lasso regression, k-nearest neighbors, overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, (generalized additive models)
4. Train/test error and data split, square error, 0-1, cross-entropy, AIC, BIC, cross-validation, one-leave-out CV, biased estimate example
5. decision trees, information gain/entropy/gini, CART α pruning
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost and gradient boosting), stacking, (MARS),
7. Bayesian learning: MAP, ML hypothesis, Bayesian optimal prediction !formulas!, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, (Cliques, MRF)
12. Gaussian processes: estimation of the function and its variance !figures, ideas!
13. Support Vector Machines (optimal separating hyperplane, slacks, kernels).

List of topics

1. Linear, ridge, lasso regression, k-nearest neighbors, overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, (generalized additive models)
4. Train/test error and data split, square error, 0-1, cross-entropy, AIC, BIC, cross-validation, one-leave-out CV, biased estimate example
5. decision trees, information gain/entropy/gini, CART α pruning
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost and gradient boosting), stacking, (MARS),
7. Bayesian learning: MAP, ML hypothesis, Bayesian optimal prediction !formulas!, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, (Ising, MRF)
12. Gaussian processes: estimation of the function and its variance !figures, ideas!
13. Support Vector Machines (optimal separating hyperplane, slacks, kernels).

List of topics

1. Linear, ridge, lasso regression, k-nearest neighbors, overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, (generalized additive models)
4. Train/test error and data split, square error, 0-1, cross-entropy, AIC, BIC, cross-validation, one-leave-out CV, biased estimate example
5. decision trees, information gain/entropy/gini, CART α pruning
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost and gradient boosting), stacking, (MARS),
7. Bayesian learning: MAP, ML hypothesis, Bayesian optimal prediction !formulas!, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, (Ising, MRF)
12. Gaussian processes: estimation of the function and its variance !figures, ideas!
13. Support Vector Machines (optimal separating hyperplane, slacks, kernels).

List of topics

1. Linear, ridge, lasso regression, k-nearest neighbors, overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, (generalized additive models)
4. Train/test error and data split, square error, 0-1, cross-entropy, AIC, BIC, cross-validation, one-leave-out CV, biased estimate example
5. decision trees, information gain/entropy/gini, CART α pruning
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost and gradient boosting), stacking, (MARS),
7. Bayesian learning: MAP, ML hypothesis, Bayesian optimal prediction !formulas!, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, (Ising, MRF)
12. Gaussian processes: estimation of the function and its variance !figures, ideas!
13. Support Vector Machines (optimal separating hyperplane, slacks, kernels).

List of topics

1. Linear, ridge, lasso regression, k-nearest neighbors, overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, (generalized additive models)
4. Train/test error and data split, square error, 0-1, cross-entropy, AIC, BIC, cross-validation, one-leave-out CV, biased estimate example
5. decision trees, information gain/entropy/gini, CART α pruning
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost and gradient boosting), stacking, (MARS),
7. Bayesian learning: MAP, ML hypothesis, Bayesian optimal prediction !formulas!, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, (Ising, MRF)
12. Gaussian processes: estimation of the function and its variance !figures, ideas!
13. Support Vector Machines (optimal separating hyperplane, slacks, kernels).

List of topics

1. Linear, ridge, lasso regression, k-nearest neighbors, overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, (generalized additive models)
4. Train/test error and data split, square error, 0-1, cross-entropy, AIC, BIC, cross-validation, one-leave-out CV, biased estimate example
5. decision trees, information gain/entropy/gini, CART α pruning
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost and gradient boosting), stacking, (MARS),
7. Bayesian learning: MAP, ML hypothesis, Bayesian optimal prediction !formulas!, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, (Potts, MRF)
12. Gaussian processes: estimation of the function and its variance !figures, ideas!
13. Support Vector Machines (optimal separating hyperplane, slacks, kernels).

List of topics

1. Linear, ridge, lasso regression, k-nearest neighbors, overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, (generalized additive models)
4. Train/test error and data split, square error, 0-1, cross-entropy, AIC, BIC, cross-validation, one-leave-out CV, biased estimate example
5. decision trees, information gain/entropy/gini, CART α pruning
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost and gradient boosting), stacking, (MARS),
7. Bayesian learning: MAP, ML hypothesis, Bayesian optimal prediction !formulas!, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, (Cliques, MRF)
12. Gaussian processes: estimation of the function and its variance !figures, ideas!
13. Support Vector Machines (optimal separating hyperplane, slacks, kernels).

List of topics

1. Linear, ridge, lasso regression, k-nearest neighbors, overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, (generalized additive models)
4. Train/test error and data split, square error, 0-1, cross-entropy, AIC, BIC, cross-validation, one-leave-out CV, biased estimate example
5. decision trees, information gain/entropy/gini, CART α pruning
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost and gradient boosting), stacking, (MARS),
7. Bayesian learning: MAP, ML hypothesis, Bayesian optimal prediction !formulas!, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, (Potts, GRF)
12. Gaussian processes: estimation of the function and its variance !figures, ideas!
13. Support Vector Machines (optimal separating hyperplane, slacks, kernels).

List of topics

1. Linear, ridge, lasso regression, k-nearest neighbors, overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, (generalized additive models)
4. Train/test error and data split, square error, 0-1, cross-entropy, AIC, BIC, cross-validation, one-leave-out CV, biased estimate example
5. decision trees, information gain/entropy/gini, CART α pruning
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost and gradient boosting), stacking, (MARS),
7. Bayesian learning: MAP, ML hypothesis, Bayesian optimal prediction !formulas!, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, (deviance, MRF)
12. Gaussian processes: estimation of the function and its variance !figures, ideas!
13. Support Vector Machines (optimal separating hyperplane, slacks, kernels).

List of topics

1. Linear, ridge, lasso regression, k-nearest neighbors, overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, (generalized additive models)
4. Train/test error and data split, square error, 0-1, cross-entropy, AIC, BIC, cross-validation, one-leave-out CV, biased estimate example
5. decision trees, information gain/entropy/gini, CART α pruning
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost and gradient boosting), stacking, (MARS),
7. Bayesian learning: MAP, ML hypothesis, Bayesian optimal prediction !formulas!, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, (deviance, MRF)
12. Gaussian processes: estimation of the function and its variance !figures, ideas!
13. Support Vector Machines (optimal separating hyperplane, slacks, kernels).

List of topics

1. Linear, ridge, lasso regression, k-nearest neighbors, overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, (generalized additive models)
4. Train/test error and data split, square error, 0-1, cross-entropy, AIC, BIC, cross-validation, one-leave-out CV, biased estimate example
5. decision trees, information gain/entropy/gini, CART α pruning
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost and gradient boosting), stacking, (MARS),
7. Bayesian learning: MAP, ML hypothesis, Bayesian optimal prediction !formulas!, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, (deviance, MRF)
12. Gaussian processes: estimation of the function and its variance !figures, ideas!
13. Support Vector Machines (optimal separating hyperplane, slacks, kernels).

List of topics

1. Linear, ridge, lasso regression, k-nearest neighbors, overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, (generalized additive models)
4. Train/test error and data split, square error, 0-1, cross-entropy, AIC, BIC, cross-validation, one-leave-out CV, biased estimate example
5. decision trees, information gain/entropy/gini, CART α pruning
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost and gradient boosting), stacking, (MARS),
7. Bayesian learning: MAP, ML hypothesis, Bayesian optimal prediction !formulas!, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, (deviance, MRF)
12. Gaussian processes: estimation of the function and its variance !figures, ideas!
13. Support Vector Machines (optimal separating hyperplane, slacks, kernels).

List of topics

1. Linear, ridge, lasso regression, k-nearest neighbors, overfitting, curse of dimensionality, (LARS)
2. Splines - the base, natural splines, smoothing splines; kernel smoothing: kernel average, Epanechnikov kernel.
3. Logistic regression, Linear discriminant analysis, (generalized additive models)
4. Train/test error and data split, square error, 0-1, cross-entropy, AIC, BIC, cross-validation, one-leave-out CV, biased estimate example
5. decision trees, information gain/entropy/gini, CART α pruning
6. random forest (+bagging), OOB error, Variable importance, boosting (Adaboost and gradient boosting), stacking, (MARS),
7. Bayesian learning: MAP, ML hypothesis, Bayesian optimal prediction !formulas!, EM algorithm
8. Clustering: k-means, Silhouette, k-medoids, hierarchical
9. Apriori algorithm, Association rules, support, confidence, lift
10. Inductive logic programming basic: hypothesis space search, background knowledge, necessity, sufficiency and consistency of a hypothesis, Aleph
11. Undirected graphical models, Graphical Lasso procedure, (deviance, MRF)
12. Gaussian processes: estimation of the function and its variance !figures, ideas!
13. Support Vector Machines (optimal separating hyperplane, slacks, kernels).

Table of Contens