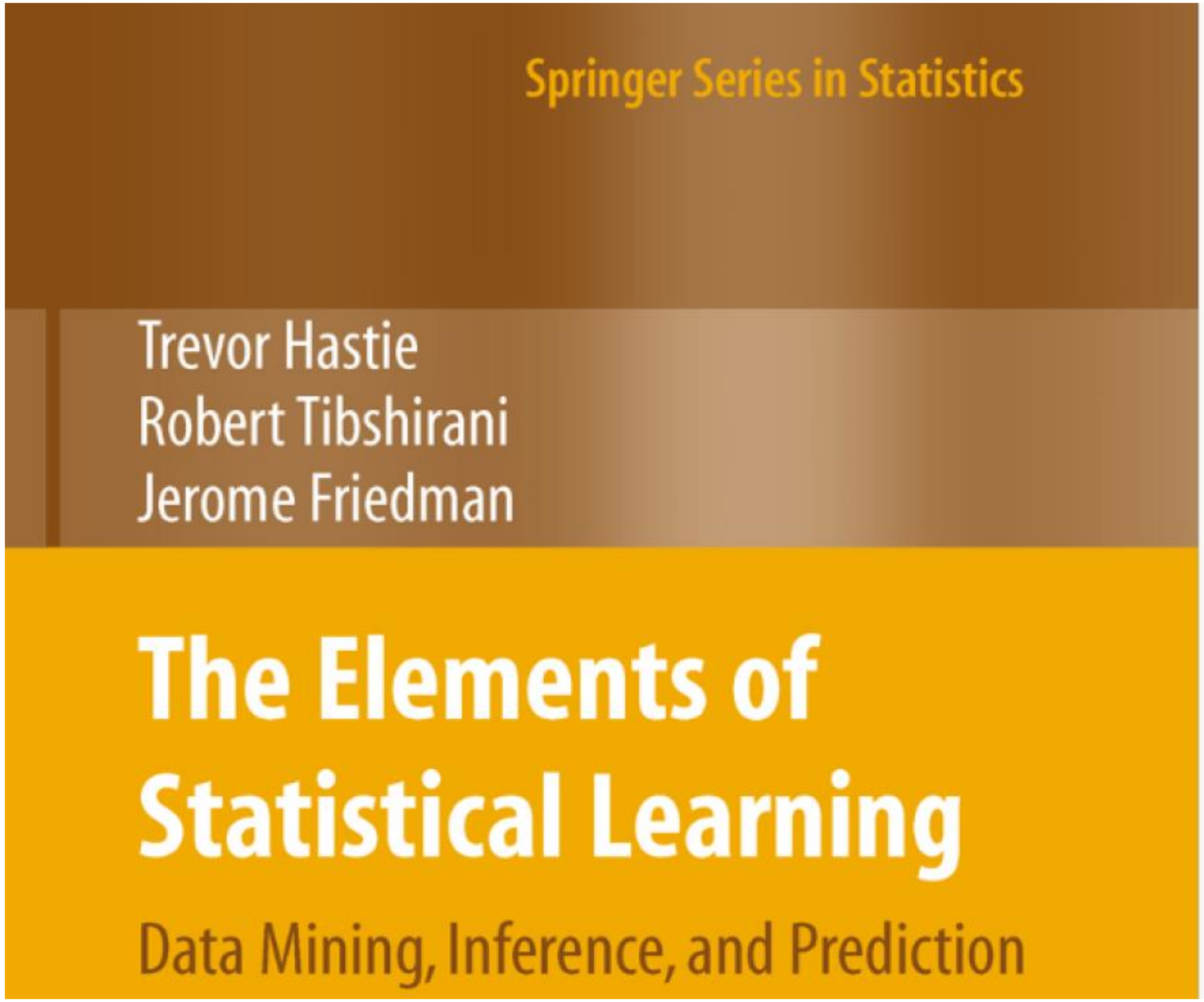
Mathematical foundations of Machine Learning 2024 – lesson 3

Shai Dekel

TEL AVIV UNIVERSITY

Support Vector Machines (SVM)



Linear SVM for binary classification

We have a dataset $\{(x_i, y_i)\}_{i\in I},\ x_i \in \mathbb{R}^n, y_i \in \{-1,1\}$

Our model is a 'best' separating hyper-plane

$$\theta := \left\{\beta \in \mathbb{R}^n,\ \|\beta\|_{l_2(\mathbb{R}^n)} = 1,\ \beta_0 \in \mathbb{R}\right\},\ \ \mathrm{P} := \left\{x \in \mathbb{R}^n : \sum_{k=1}^{n} \beta_k x_k + \beta_0 = 0\right\}$$

Once found, the inference of a new feature vector $x = (x_1, \ldots, x_n)$ is

$$\operatorname{sgn}\left(\sum_{k=1}^{n} \beta_k x_k + \beta_0\right),\quad \operatorname{dist}(x, P) = \left|\sum_{k=1}^{n} \beta_k x_k + \beta_0\right|$$

Linear SVM – separable case

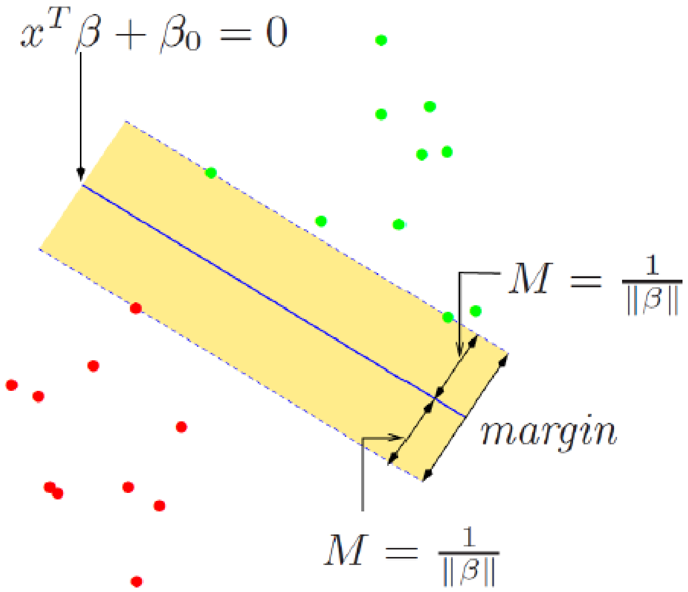
'Best' separating hyper-plane – largest <u>margin</u> on training set

$$\max_{\|\beta\|=1} M \quad \text{s.t.} \quad y_i\left(\langle x_i,\beta\rangle+\beta_0\right)\geq M, \ \ \forall i\in I_{\text{train}} \qquad \Leftrightarrow$$

$$\max_{\|\beta\|=1} M \quad \text{s.t.} \quad y_i\left(\left\langle x_i,\frac{\beta}{M}\right\rangle+\frac{\beta_0}{M}\right)\geq 1, \ \ \forall i\in I_{\text{train}} \qquad \underset{\tilde{\beta}=\beta/M,\,\tilde{\beta}_0=\beta_0/M}{\Leftrightarrow}$$

$$\min_{\tilde{\beta},\tilde{\beta}_0}\|\tilde{\beta}\| \quad \text{s.t.} \quad y_i\left(\langle x_i,\tilde{\beta}\rangle+\tilde{\beta}_0\right)\geq 1, \ \ \forall i\in I_{\text{train}} \qquad \Leftrightarrow$$

$$\min_{\tilde{\beta},\tilde{\beta}_0}\|\tilde{\beta}\|^2 \quad \text{s.t.} \quad y_i\left(\langle x_i,\tilde{\beta}\rangle+\tilde{\beta}_0\right)\geq 1, \ \ \forall i\in I_{\text{train}}.$$



Linear SVM – separable case

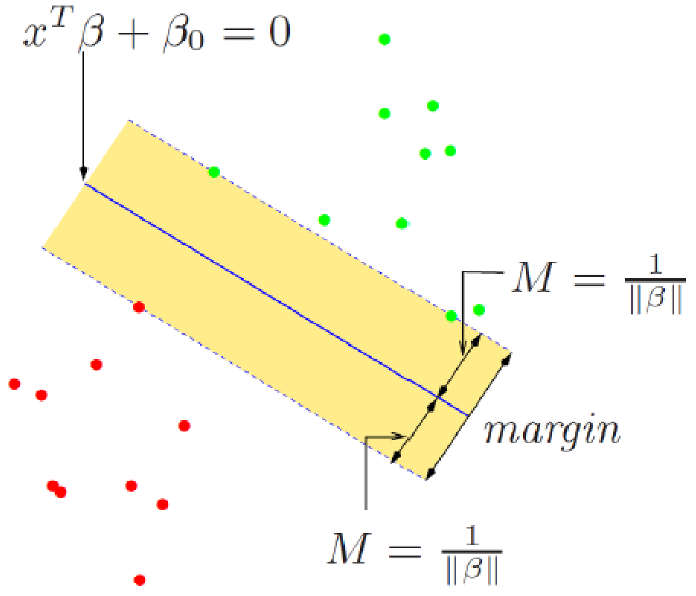
$$\max_{\theta,\|\beta\|=1} M, \quad \text{subject to } y_i(\langle \beta, x_i \rangle + \beta_0) \geq M, \ \forall i \in I_{\text{train}}.$$

Equivalent formulation

$$\min_{\theta} \|\beta\|, \quad \text{subject to } y_i(\langle \beta, x_i \rangle + \beta_0) \geq 1, \ \forall i \in I_{\text{train}}.$$

We may then normalize β for inference

Minimizing $\|\beta\|^2$ gives a convex optimization problem: quadratic criterion, linear constraints



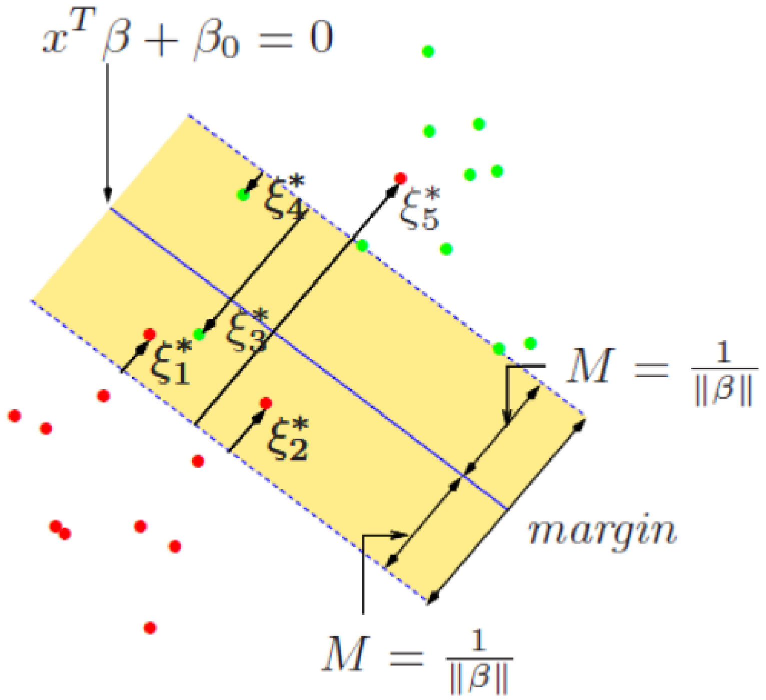
Linear SVM – non separable case

What if a separating hyper-plane does not exist? We add slack variables

$$\min_{\theta,\{\xi_i\}} \frac{1}{2}\|\beta\|^2 + C\sum_{i\in I_{\text{train}}} \xi_i,$$

$$\text{subject to } y_i\left(\langle \beta, x_i\rangle + \beta_0\right) \geq 1-\xi_i,\ \ \xi_i \geq 0, \forall i \in I_{\text{train}}$$

C serves as a 'weight' term. $C=\infty$: separable case.



Linear SVM – non separable case

The Lagrange (primal) function is

$$L_P = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[y_i(x_i^T\beta + \beta_0) - (1 - \xi_i)] - \sum_{i=1}^{N}\mu_i\xi_i, \quad (12.9)$$

which we minimize w.r.t β, β_0 and ξ_i. Setting the respective derivatives to zero, we get

$$\beta = \sum_{i=1}^{N}\alpha_i y_i x_i, \quad (12.10)$$

$$0 = \sum_{i=1}^{N}\alpha_i y_i, \quad (12.11)$$

$$\alpha_i = C - \mu_i, \ \forall i, \quad (12.12)$$

Linear SVM – non separable case

The Lagrange (primal) function is

$$L_P = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[y_i(x_i^T\beta + \beta_0) - (1-\xi_i)] - \sum_{i=1}^{N}\mu_i\xi_i, \quad (12.9)$$

which we minimize w.r.t β, β_0 and ξ_i. Setting the respective derivatives to zero, we get

$$\beta = \sum_{i=1}^{N}\alpha_i y_i x_i, \quad (12.10)$$

$$0 = \sum_{i=1}^{N}\alpha_i y_i, \quad (12.11)$$

$$\alpha_i = C - \mu_i, \ \forall i, \quad (12.12)$$

as well as the positivity constraints $\alpha_i, \ \mu_i, \ \xi_i \geq 0 \ \forall i$. By substituting (12.10)–(12.12) into (12.9), we obtain the Lagrangian (Wolfe) dual objective function

$$L_D = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{i'=1}^{N}\alpha_i\alpha_{i'}y_i y_{i'}x_i^T x_{i'}, \quad (12.13)$$

Linear SVM – non separable case

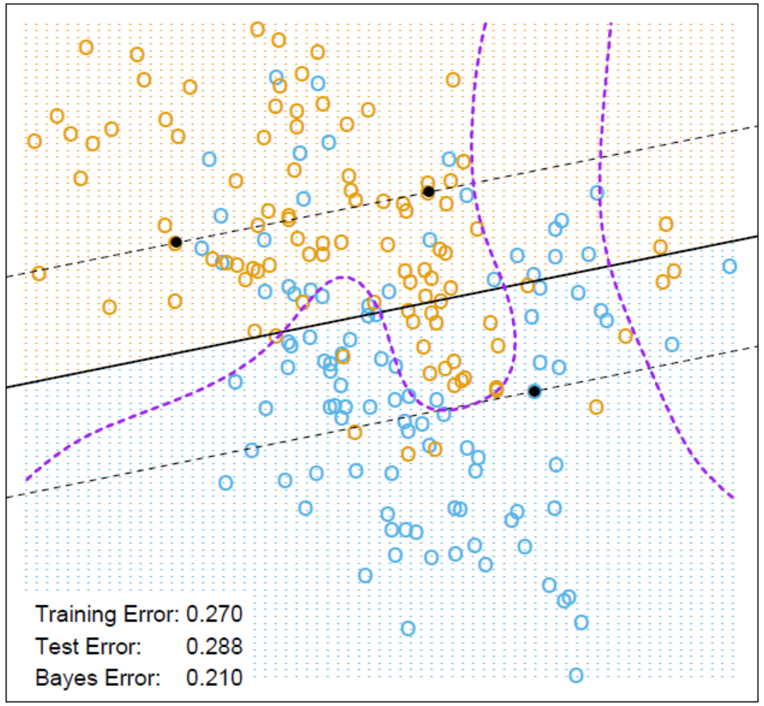
$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \alpha_i \alpha_{i'} y_i y_{i'} x_i^T x_{i'}, \qquad (12.13)$$

which gives a lower bound on the objective function (12.8) for any feasible point. We maximize L_D subject to $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$. In

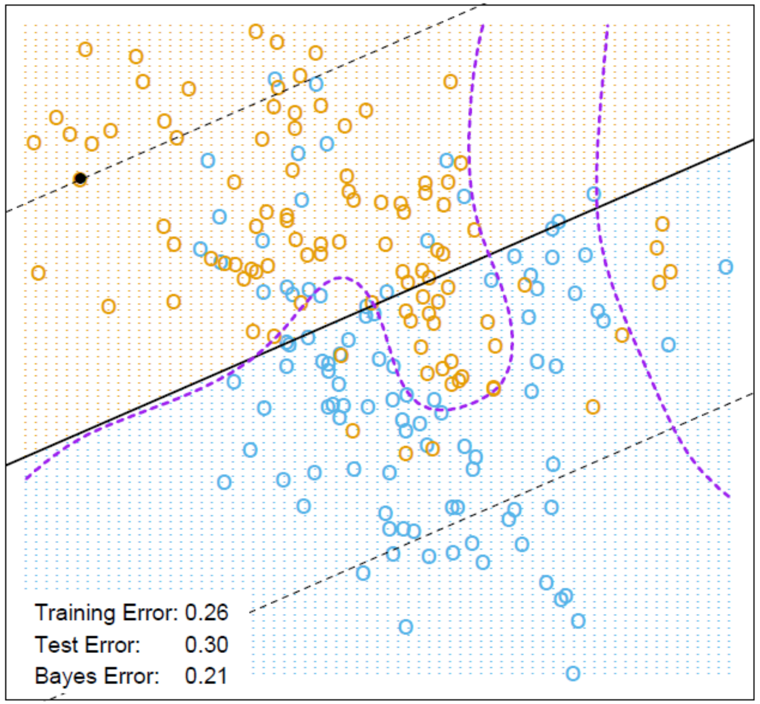
$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^{N} \xi_i \qquad (12.8)$$

$$\text{subject to} \quad \xi_i \geq 0, \; y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i \; \forall i,$$

Linear SVM – non separable case



$C = 10000$



$C = 0.01$

```python
from sklearn.svm import LinearSVC
linSVM = LinearSVC()
linSVM.fit(X_train,Y_train)
```

```
C:\Users\user\anaconda3\Lib\site-pack
om `True` to `'auto'` in 1.5. Set the
  warnings.warn(
```

```
▾ LinearSVC

LinearSVC()
```

kernel SVM

The idea is to transform the features using a nonlinear transformation to a higher-dimensional space and find a separating hyper-plane there.

$$h:\mathbb{R}^n \to \mathbb{R}^M,\ h(x) = \left(h_1(x),\ldots,h_M(x)\right)$$

The hyper-plane in defined by $\hat{\theta} := \left\{\hat{\beta} \in \mathbb{R}^M, \hat{\beta}_0 \in \mathbb{R}\right\}$

$$\left\langle h(x), \hat{\beta} \right\rangle + \hat{\beta}_0 = 0$$

Inference is by

$$\operatorname{sgn}\left(\left\langle h(x), \hat{\beta} \right\rangle + \hat{\beta}_0\right)$$

kernel SVM

- We can represent the problem in a special way, replacing the explicit mapping h by a kernel.
- First observe that the dual maximization is determined by dot-products of the mappings

$$L_D = \sum_{i \in I_{\text{train}}} \hat{\alpha}_i - \frac{1}{2} \sum_{i \in I_{\text{train}}} \sum_{i \in I_{\text{train}}} \hat{\alpha}_i \hat{\alpha}_j y_i y_j \left\langle h(x_i), h(x_j) \right\rangle \tag{12.19}$$

- Using (12.10)

$$\left\langle h(x), \hat{\beta} \right\rangle + \hat{\beta}_0 = \sum_{i \in I_{\text{train}}} \hat{\alpha}_i y_i \left\langle h(x), h(x_i) \right\rangle + \hat{\beta}_0 \tag{12.20}$$

kernel SVM

So both (12.19) and (12.20) involve $h(x)$ only through inner products. In fact, we need not specify the transformation $h(x)$ at all, but require only knowledge of the kernel function

$$K(x, x') = \langle h(x), h(x') \rangle \tag{12.21}$$

that computes inner products in the transformed space. K should be a symmetric positive (semi-) definite function; see Section 5.8.1.

Three popular choices for K in the SVM literature are

$$\begin{aligned} d\text{th-Degree polynomial: } K(x, x') &= (1 + \langle x, x' \rangle)^d, \\ \text{Radial basis: } K(x, x') &= \exp(-\gamma \|x - x'\|^2), \\ \text{Neural network: } K(x, x') &= \tanh(\kappa_1 \langle x, x' \rangle + \kappa_2). \end{aligned} \tag{12.22}$$

kernel SVM…is feature engineering?

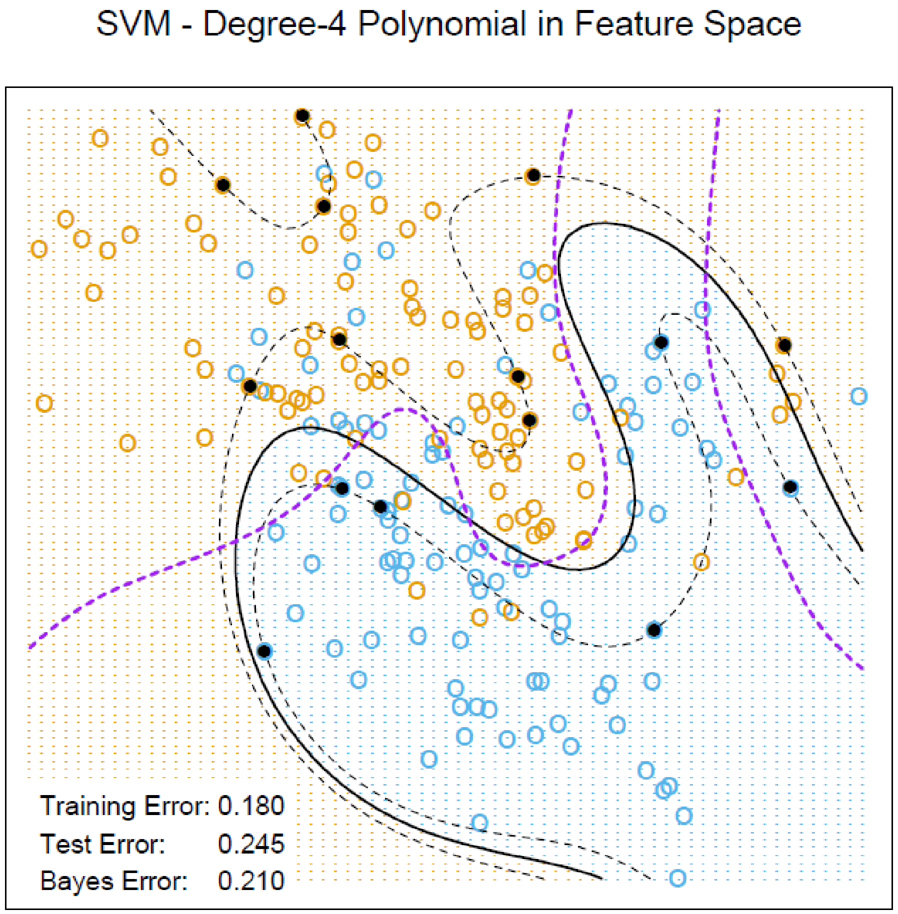
Consider for example a feature space with two inputs X_1 and X_2, and a polynomial kernel of degree 2. Then

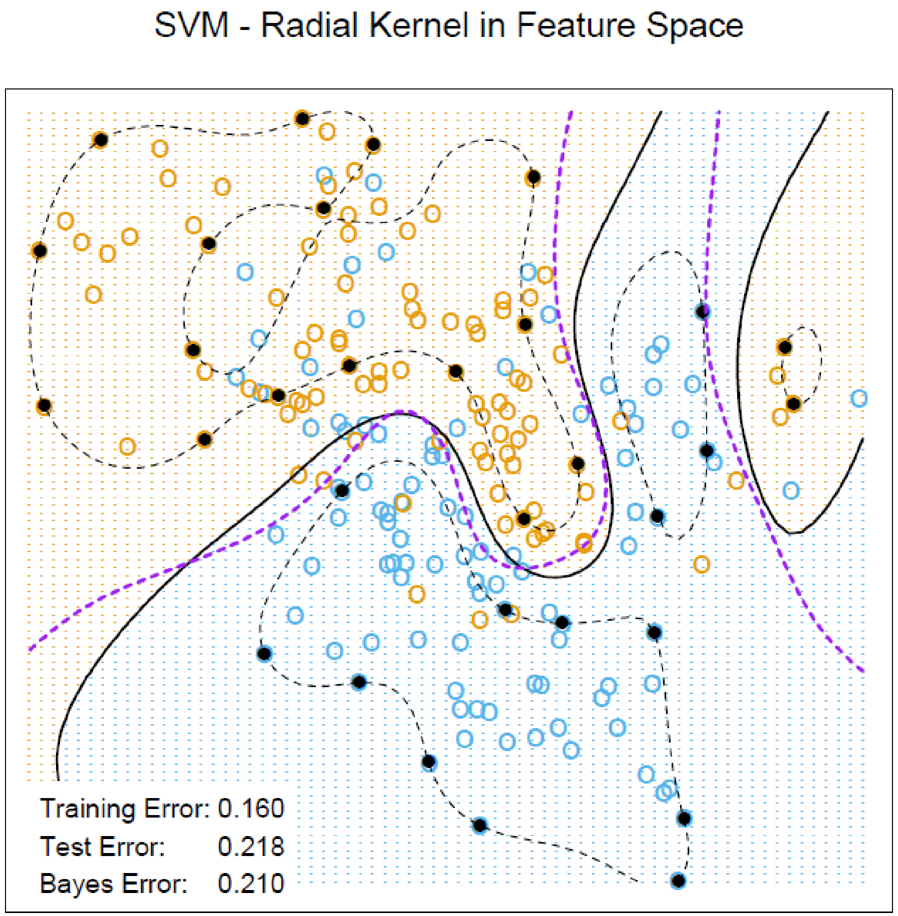
$$\begin{aligned} K(X, X') &= (1 + \langle X, X' \rangle)^2 \\ &= (1 + X_1 X_1' + X_2 X_2')^2 \\ &= 1 + 2X_1X_1' + 2X_2X_2' + (X_1X_1')^2 + (X_2X_2')^2 + 2X_1X_1'X_2X_2'. \end{aligned} \tag{12.23}$$

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

$$\hat{f}(x) = \sum_{i=1}^{N} \hat{\alpha}_i y_i K(x, x_i) + \hat{\beta}_0. \tag{12.24}$$

kernel SVM






```
from sklearn import svm
kernelSVM = svm.SVC(kernel='rbf')
kernelSVM.fit(X_train,Y_train)
```

```
▾ SVC
SVC()
```