12. SVM & Flexible Discriminants

오영민

Introduction

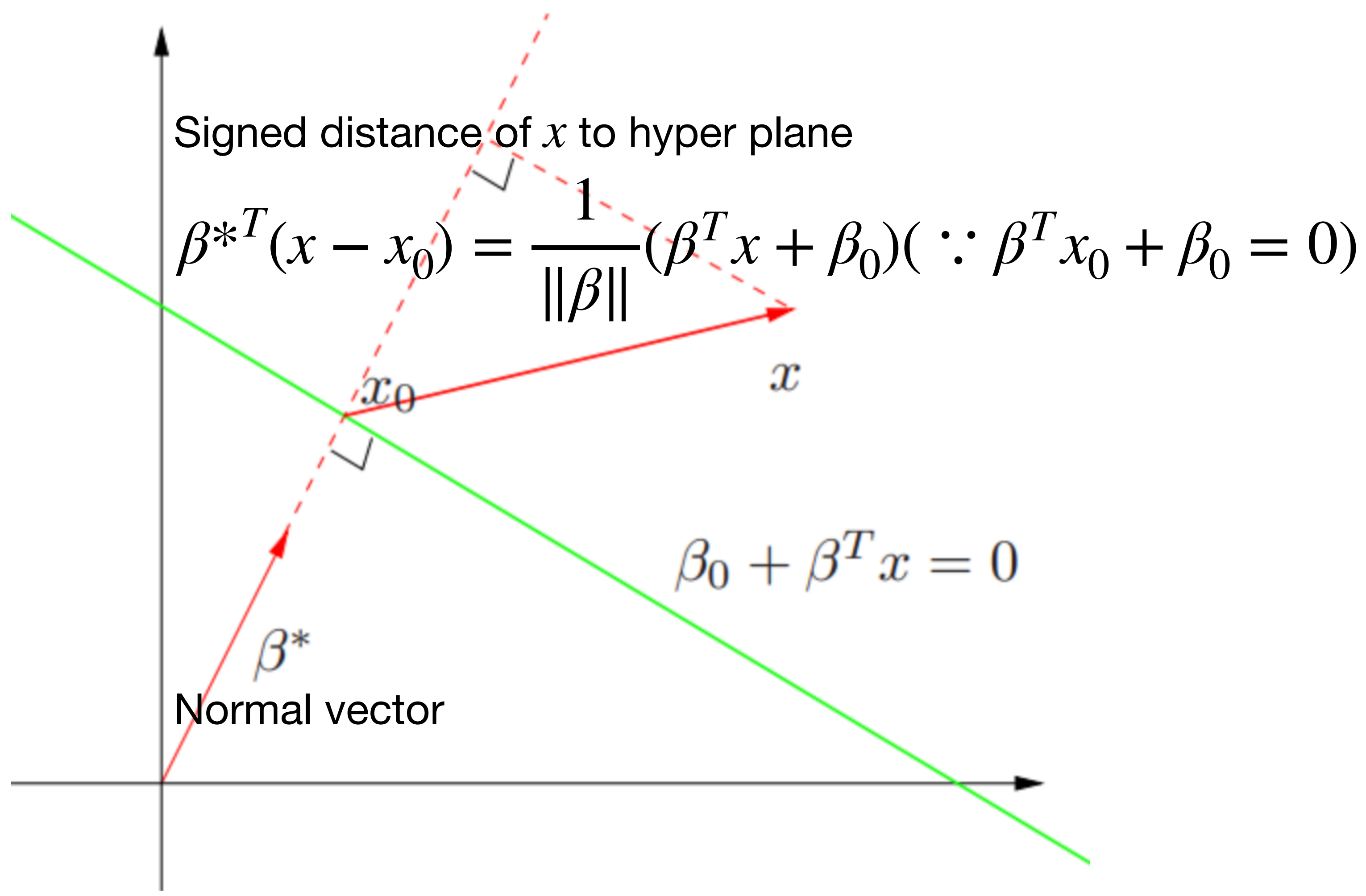
- The Support Vector Classifier
- Support Vector Machines and Kernels
- Curse of dimensionality
- Regularization parameter for SVM
- SVM for Regression

The Support Vector Classifier

- **Binary case:** N-pairs training data $\{x_i, y_i\}_1^N$, $x_i \in \mathbb{R}^p$, $y_i \in \{-1,1\}$

- **Hyper plane.** $\{x : f(x) = x^T\beta + \beta_0 = 0,\ \|\beta\| = 1\}$

- **Classification rule induced by** $f(x)$. $G(x) = sign[f(x)]$.



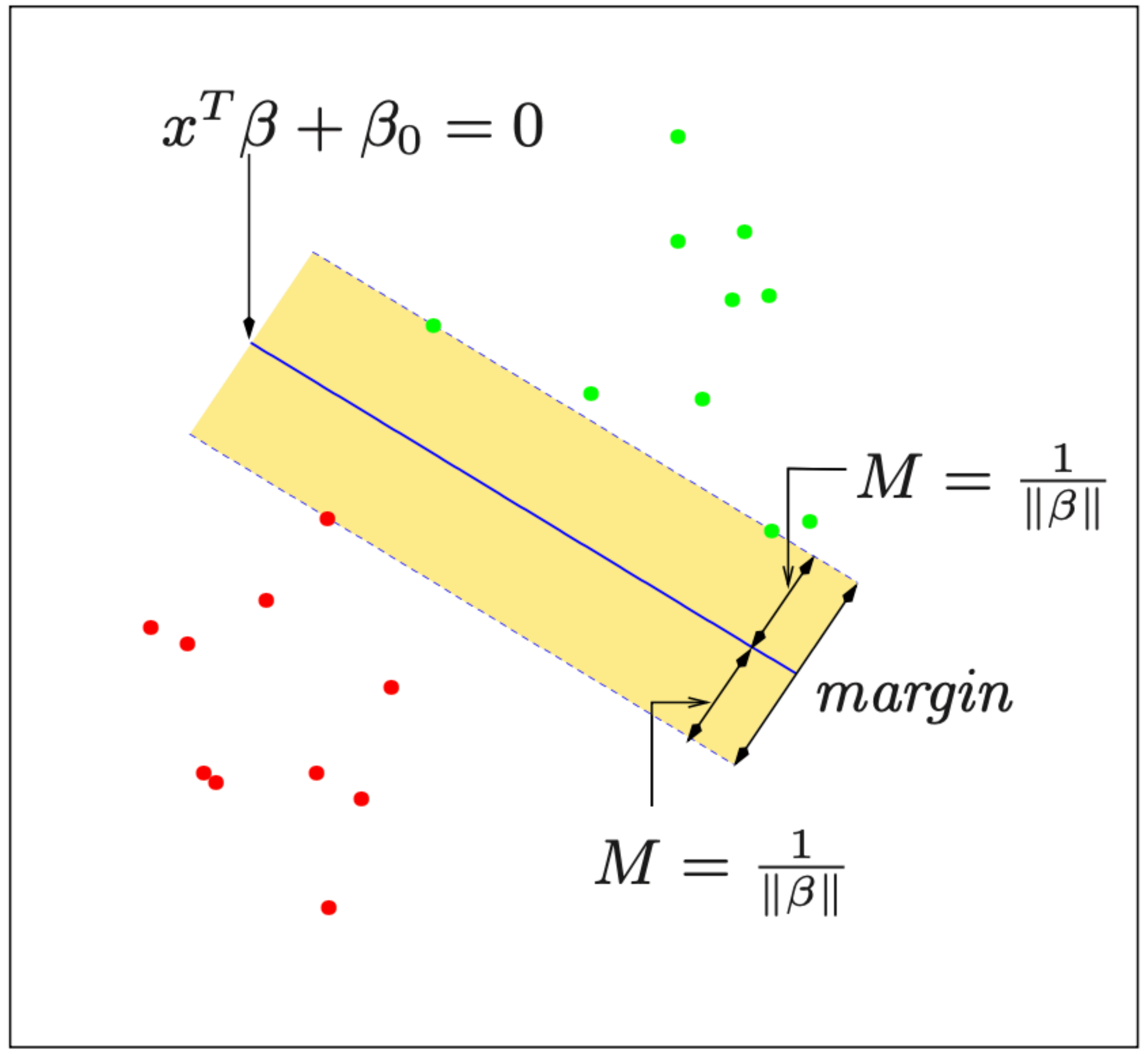
- **For linearly separable case**, $y_i f(x_i) > 0$ for all i. Then the optimization problem:

$$\max_{\beta, \beta_0, \|\beta\|=1} M \text{ subject to } y_i(x_i^T\beta + \beta_0) \geq M,\ \forall i$$

Equivalently,

$$\min_{\beta, \beta_0} \|\beta\| \text{ subject to } y_i(x_i^T\beta + \beta_0) \geq 1,\ \forall i \text{ where } M = 1/\|\beta\|$$

Which is a **convex** optimization problem
(quadratic criterion, linear equality constraints).



Linearly separable case

The Support Vector Classifier

- **For linearly non-separable case**: Allow for some points to be on the wrong side of the margin. Define the **slack variables** $\xi = (\xi_1, \ldots, \xi_N)$.

- Modify the constraints.

    - Option 1. $y_i f(x_i) \geq M - \xi_i,\ \xi_i \geq 0\ , \forall i$, and $\sum_i \xi_i \leq$ constant (measures overlap in actual distance from margin)

    - Option2. $y_i f(x_i) \geq \underline{M(1 - \xi_i)},\ \xi_i \geq 0\ , \forall i$, and $\underline{\sum_i \xi_i \leq \text{constant}}$ (measures overlap in relative distance from margin)

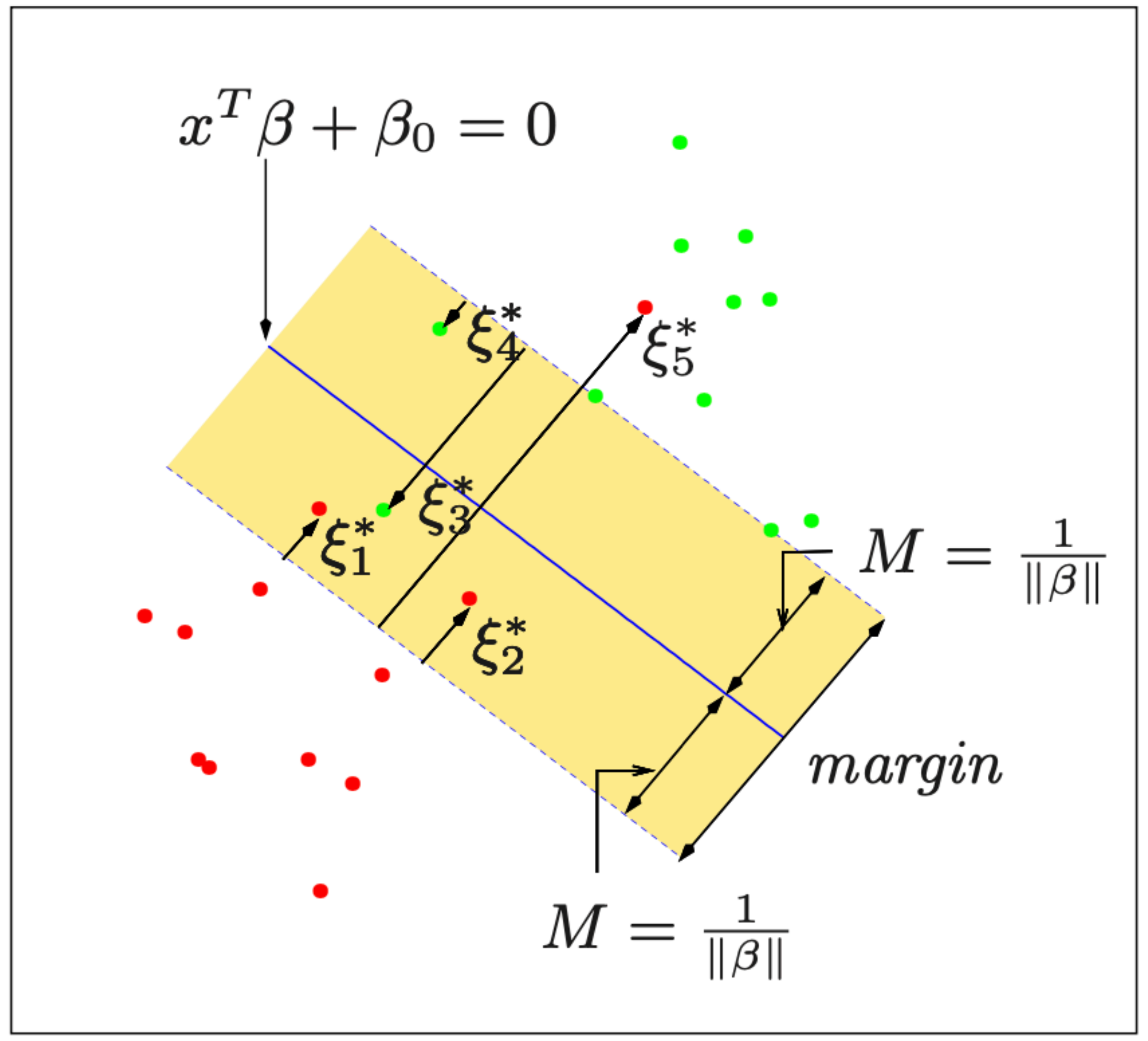
      Misclassification occurs at $\xi_i > 1$

      e.g. constant K says, bounds # training misclassification.

- First constraint results in a non-convex optimization, while the second is convex.

$$\min_{\beta, \beta_0} \|\beta\| \text{ subject to } y_i f(x_i) \geq 1 - \xi_i,\ \xi_i \geq 0, \forall i,\ \sum_i \xi_i \leq \text{constant}$$

- Points well inside their class boundary **do not play a big role** in shaping the boundary.



Linearly non-separable case

Computing the SVC

- **Optimization:** $\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 + C \sum_i \xi_i$ subject to $y_i f(x_i) \geq 1 - \xi_i,\ \xi_i \geq 0, \forall i$

  For linearly separable case, $C \to \infty$

- **Primal:** $L_p = \frac{1}{2}\|\beta\|^2 + C\sum_i \xi_i - \sum_i \alpha_i[y_i f(x_i) - (1-\xi_i))] - \sum_i \mu_i \xi_i$, which maximize w.r.t. β, β_0, ξ_i's.

- **Dual:** $L_D = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_{i'} \alpha_i \alpha_{i'} y_i y_{i'} x_i^T x_{i'}$, which is $L_p(\beta^*, \beta_0^*, \xi_i^*)$ where each are derivatives to 0.

$$\beta = \sum_i \alpha_i y_i x_i$$

$$0 = \sum_i \alpha_i y_i$$

$$\alpha_i = C - \mu_i, \forall i$$

- **Our goal:** maximize L_D subject to $0 \leq \alpha_i \leq C$, and $\sum_i \alpha_i y_i = 0$ (due to derivation), and KKT condition include the constraints

  $1.\ \alpha_i[f(x_i) - (1-\xi_i)] = 0,\ \ 2.\ \mu_i \xi_i = 0,\ \ 3.\ y_i f(x_i) - (1-\xi_i) \geq 0,\ \forall i\,.$ (1 is complementary slackness and 2&3 are primal feasibility.)

  These equations uniquely characterize the solution to the primal and dual problem. The solution for β has the form

  $\hat{\beta} = \sum_i \hat{\alpha}_i y_i x_i$ (x_i is called **support vector** if the coefficient $\hat{\alpha}_i$ is non-zero.)

Computing the SVC

- Among these support points ($\hat{\alpha}_i > 0$), some will lie on the edge of the margin ($\hat{\xi}_i = 0$, called **margin points**) can be used to solve for β_0, and we typically use an average of all the solutions for numerical stability. (complementary slackness)

- $0 \leq \hat{\alpha}_i \leq C$, $\forall i$. $\hat{\alpha}_i = C$ if $\hat{\xi}_i > 0$. (primal feasibility & dual condition)

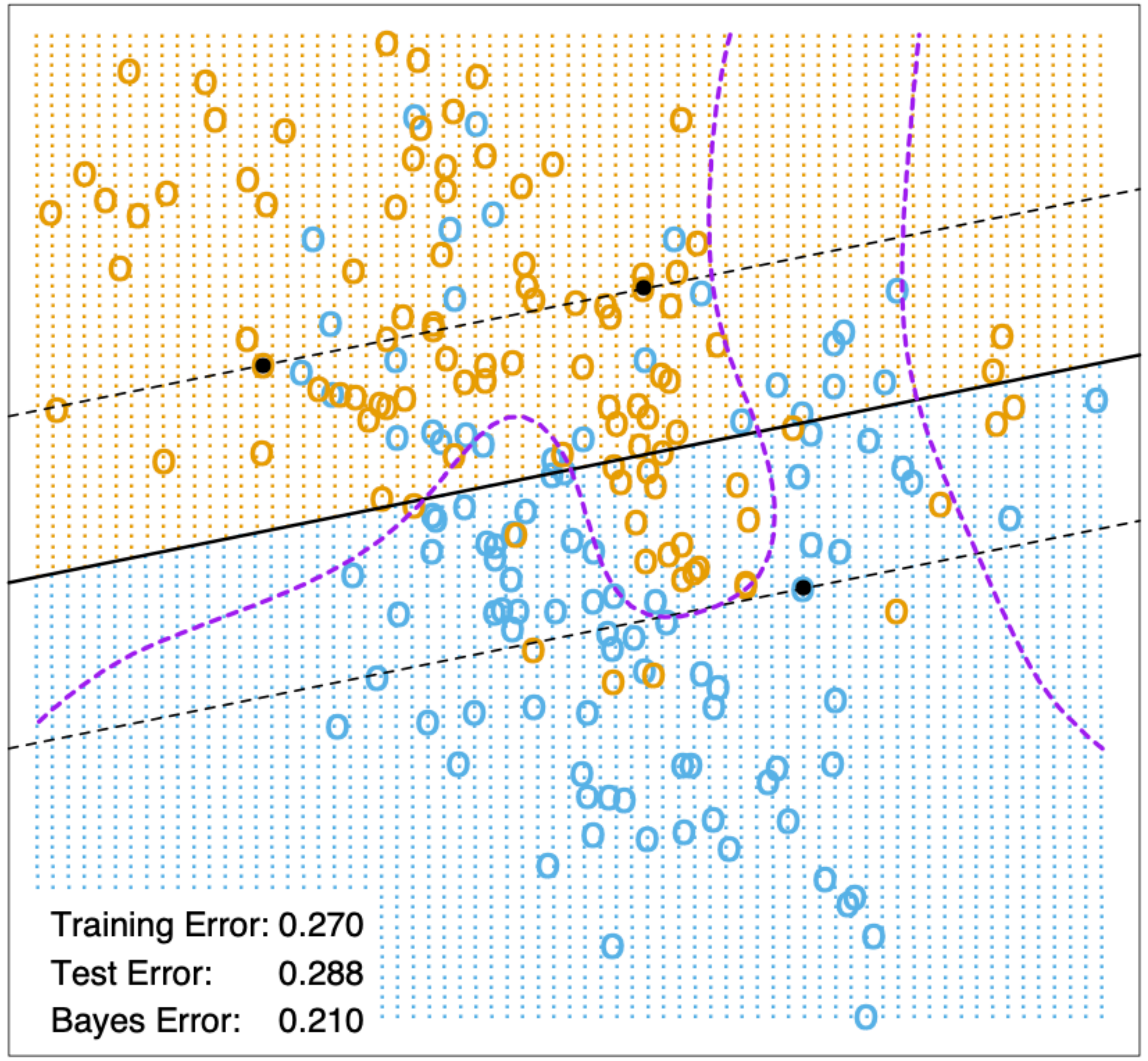
- Given the solutions $\hat{\beta}_0$ and $\hat{\beta}$, the decision function can be written as $\hat{G}(x) = sign[\hat{f}(x)] = sign[x^T\hat{\beta} + \hat{\beta}_0]$.

- Thu tuning parameter of SVC is the cost parameter C.

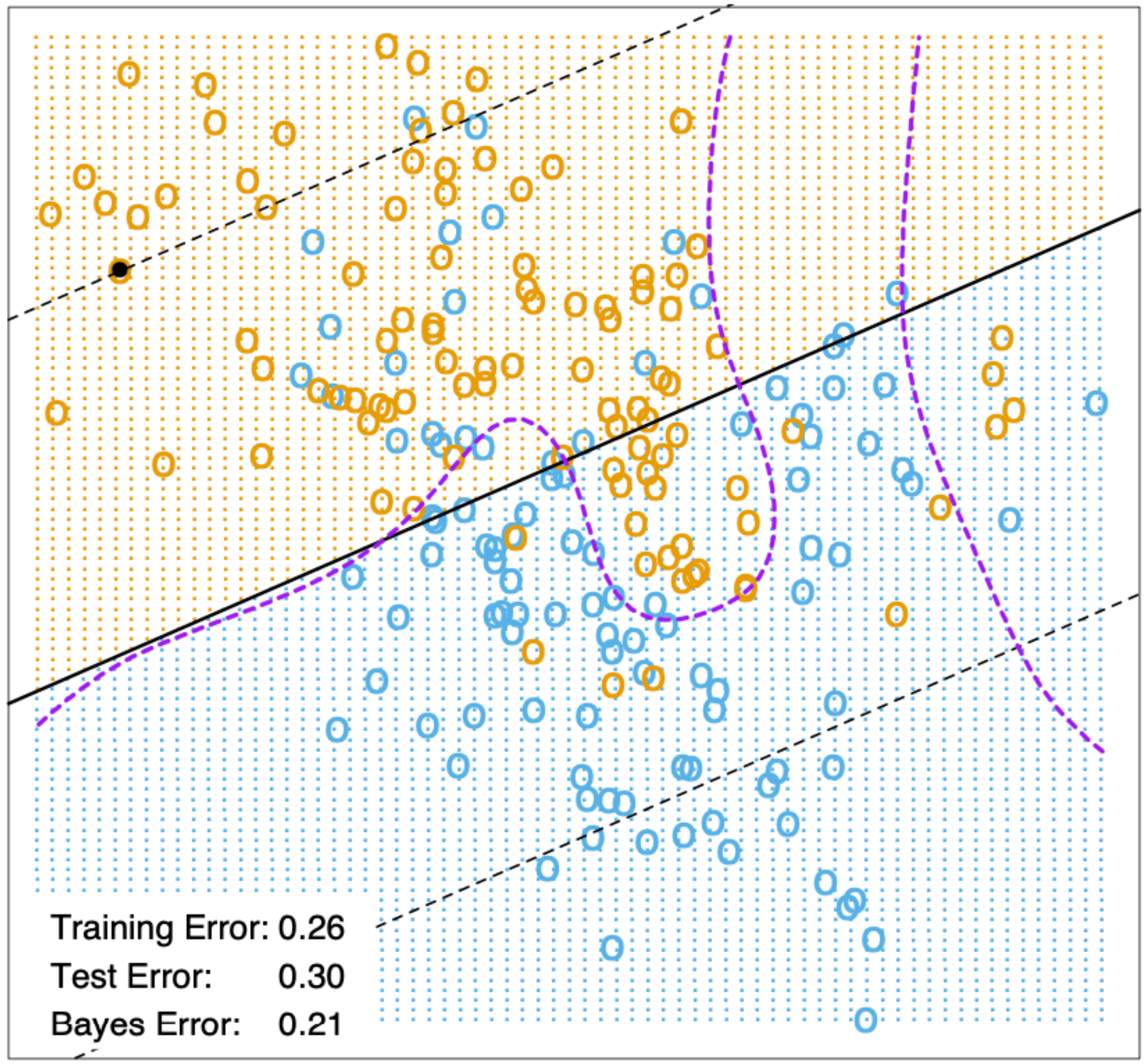
Cost parameter C

- Larger values of C focus attention more on correctly classified points near the decision boundary.

- The optimal value for C can be estimated by cross-validation. If validation set does not include support vector, then the solution is un-changed. i.e. LOOCV is not recommended.



$C = 10000$



$C = 0.01$

Kernel method

- Use basis functions $h_m : \mathbb{R}^p \to \mathbb{R}$, $m = 1,...,M$ and fit the SVC using input $h(x_i) = (h_1(x_i), \ldots, h_M(x_i)) \in \mathbb{R}^m$ instead $x_i \in \mathbb{R}^p$.
- The support vector machine classifier is an extension of this idea, where the dimension of the enlarged space is allowed to get very large.
- The classifier is $\hat{G}(x) = sign(\hat{f}(x))$ where $\hat{f}(x) = h(x)^T \hat{\beta} + \hat{\beta}_0$
- **Dual:** $L_D = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_{i'} \alpha_i \alpha_{i'} y_i y_{i'} \langle h(x_i), h(x_{i'}) \rangle$ and the solution has the form $\hat{\beta} = \sum_i \hat{\alpha}_i y_i h(x_i)$.
- Thus, the solution function is $\hat{f}(x) = h(x)^T \hat{\beta} + \hat{\beta}_0 = \sum_i \alpha_i y_i \langle h(x), h(x_i) \rangle + \hat{\beta}_0$ which means we need not specify the transformation $h(x)$ at all, but require only knowledge of the kernel function $K(x, x') = \langle h(x), h(x') \rangle$.
- K should be a symmetric positive semi-definite functions.

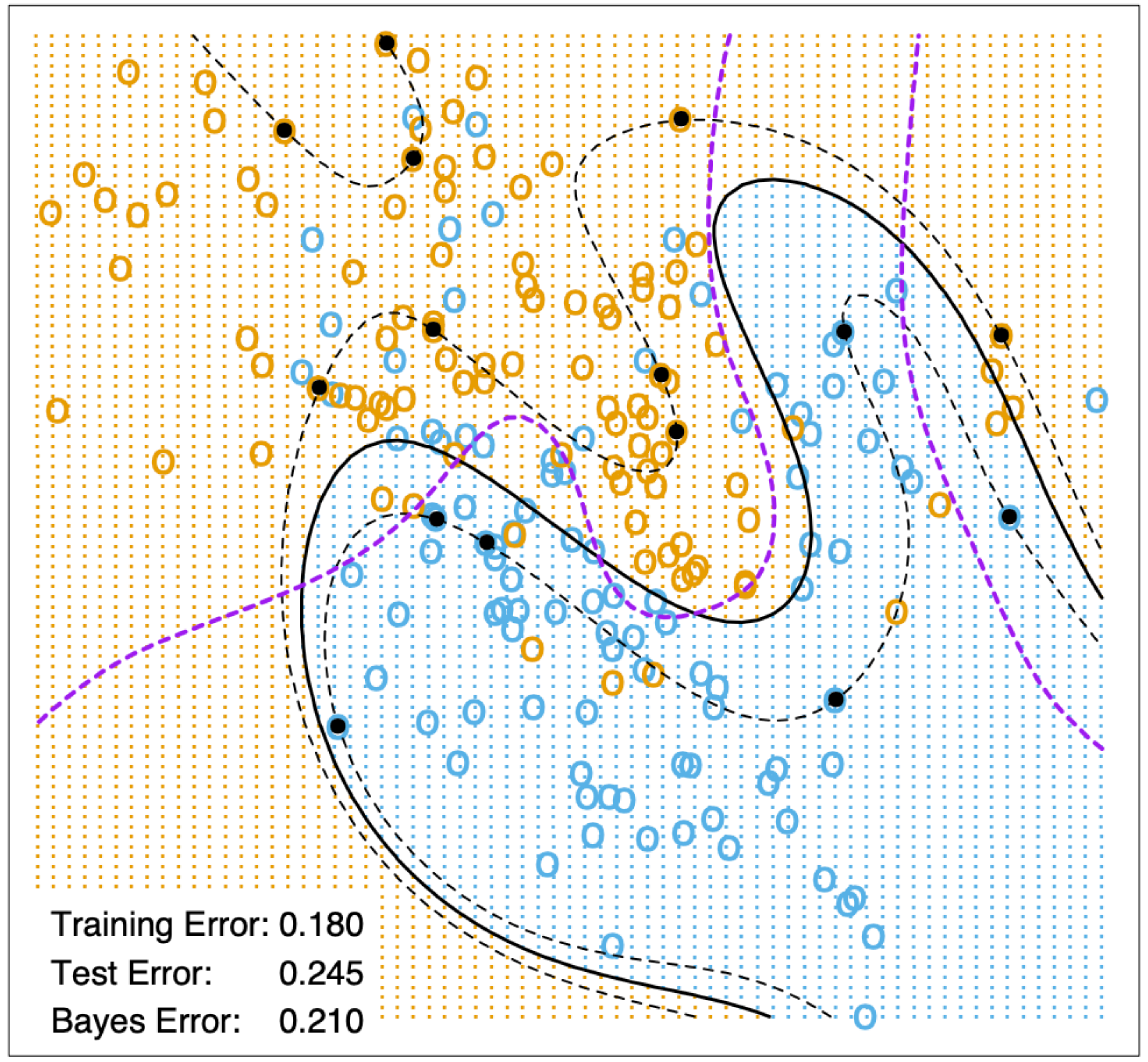
Kernel method

- Popular choices K in the SVM literature are

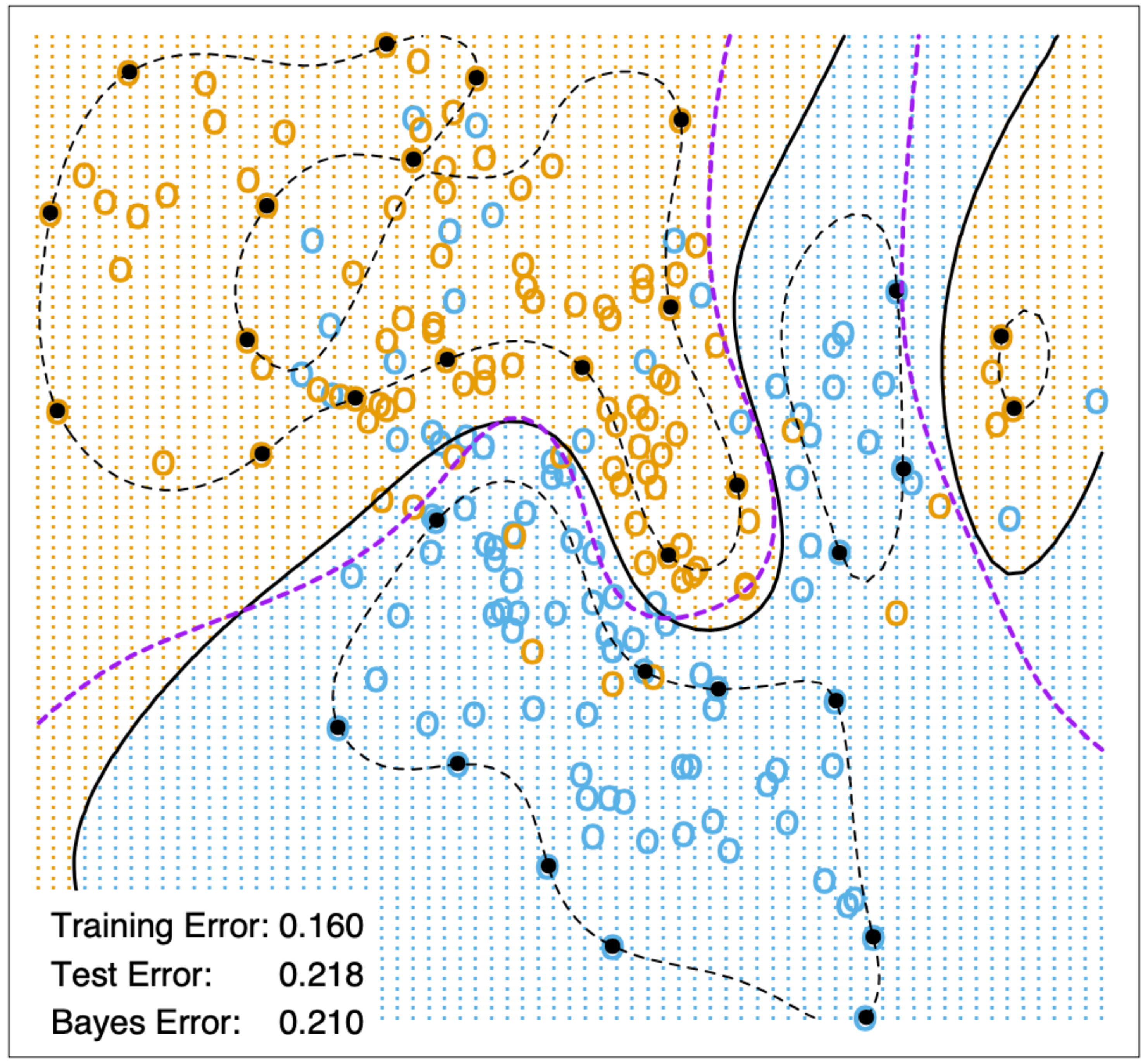
$$\begin{aligned} d\text{th-Degree polynomial: } & K(x, x') = (1 + \langle x, x' \rangle)^d, \\ \text{Radial basis: } & K(x, x') = \exp(-\gamma \|x - x'\|^2), \\ \text{Neural network: } & K(x, x') = \tanh(\kappa_1 \langle x, x' \rangle + \kappa_2). \end{aligned}$$

- The role of the cost parameter C is clearer, since perfect separation is often achievable there. A large value of C will often lead to an overfit wiggly boundary in the original feature space.

SVM - Degree-4 Polynomial in Feature Space



SVM - Radial Kernel in Feature Space



In each case C was tuned to approximately achieve the best test error performance, and $C = 1$ worked well in both cases.

Kernel method

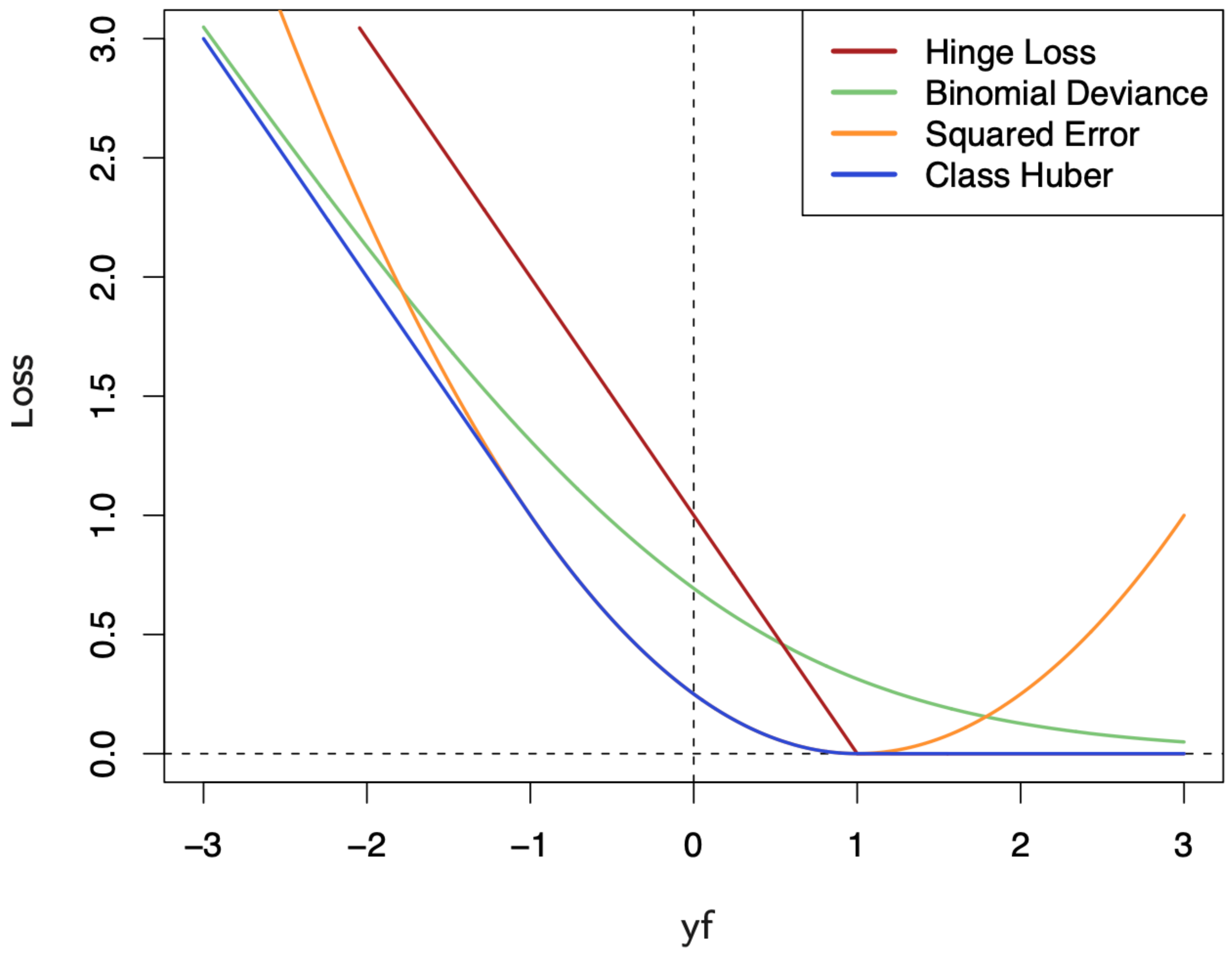
- SVM optimization: $\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 + C\sum_i \xi_i$ subject to $y_i f(x_i) \geq 1 - \xi_i,\ \xi_i \geq 0, \forall i$

$$\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 + C\sum_i \xi_i \text{ subject to } \xi_i \geq [1 - y_i f(x_i)]_+, \forall i$$

- It is only the support vectors ($\xi_i = [1 - y_i f(x_i)]_+$) that affects the solution, above solution is equivalent to solve following optimization problem:

$$\min_{\beta,\beta_0} C[\frac{1}{2C}\|\beta\|^2 + \sum_i [1 - y_i f(x_i)]_+]$$

$$= \min_{\beta,\beta_0} \sum_i [1 - y_i f(x_i)]_+ + \frac{\lambda}{2}\|\beta\|^2 \text{ where } \lambda = 1/C$$



Kernel method

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

The SVM hinge loss estimates **the mode of the posterior class probabilities**, whereas the others estimate a linear transformation of these probabilities.

Reproducing kernels

- Suppose the basis h arises from the Eigen expansion of a positive definite kernel $K(x, x') = \sum_{m=1}^{\infty} \phi_m(x)\phi_m(x')\delta_m$.
- Let $h_m(x) = \sqrt{\delta_m}\phi_m(x)$. Then, $\theta_m = \sqrt{\delta_m}\beta_m$, we can rewrite the hinge loss:

$$\min_{\beta_0,\theta} \sum_{i=1}^{N} [1 - y_i(\beta_0 + \sum_{m=1}^{\infty} \theta_m\phi_m(x_i))]_+ + \frac{\lambda}{2}\sum_{m=1}^{\infty} \frac{\theta_m^2}{\delta_m}$$

- The theory of reproducing kernel Hilbert spaces described there guarantees a finite-dimensional solution of the form

$$f(x) = \beta_0 + \sum_{i=1}^{N} \alpha_i K(x, x_i)$$

- For $K(x, x') = h(x)^T h(x')$, above optimization problem is equivalent to

$$\min_{\beta_0,\theta} \sum_{i=1}^{N} [1 - y_i f(x_i)]_+ + \frac{\lambda}{2}\alpha^T \mathbf{K}\alpha \text{ where } \mathbf{K}_{ij} = K(x_i, x_j),\ i, j = 1,\ldots,N$$

Reproducing kernels

- The optimization problem can be expressed more generally as $\min_{f \in \mathscr{H}} \sum_{i=1}^{N} [1 - y_i f(x_i)]_+ + \lambda J(f)$ where $\mathscr{H}$ is the structured space of functions, and $J(f)$ an appropriate regularizer.

- e.g. Suppose $\mathscr{H} = \left\{ f : f(x) = \sum_{j=1}^{p} f_j(x_j),\ x = (x_1, \ldots, x_p)^T \right\}$ and $J(f) = \sum_j \int \{f_j''(x_j)\}^2 dx_j$. Then the solution is an additive cubic spline, and has a kernel $K(x, x') = \sum_j K_j(x_j, x_j')$ where each of the K_j is the univariate smoothing spline in x_j.

- Conversely, any of the kernels we mentioned can be used with any convex loss function, and will also lead to a finite-dimensional representation.

- e.g. For the binomial log-likelihood loss, $\hat{f}(x) = logit(\hat{P}r(Y = 1 \,|\, x)) = \hat{\beta}_0 + \sum_{i=1}^{N} \hat{\alpha}_i K(x, x_i)$.

Curse of dimensionality

$X_1, X_2, X_3, X_4 |_{y=-1} \sim N(0,1)$ for noise cases, add 6 standardized normal noise for both labels.

$X_1, X_2, X_3, X_4 |_{y=1} \sim N(0,1)$ subject to $9 \leq \sum_j X_j^2 \leq 16$

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.*

	Method	Test Error (SE): No Noise Features	Test Error (SE): 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

It is also very sensitive to the choice of kernel and adversely affected by the six noise features

A path algorithm for the SVC

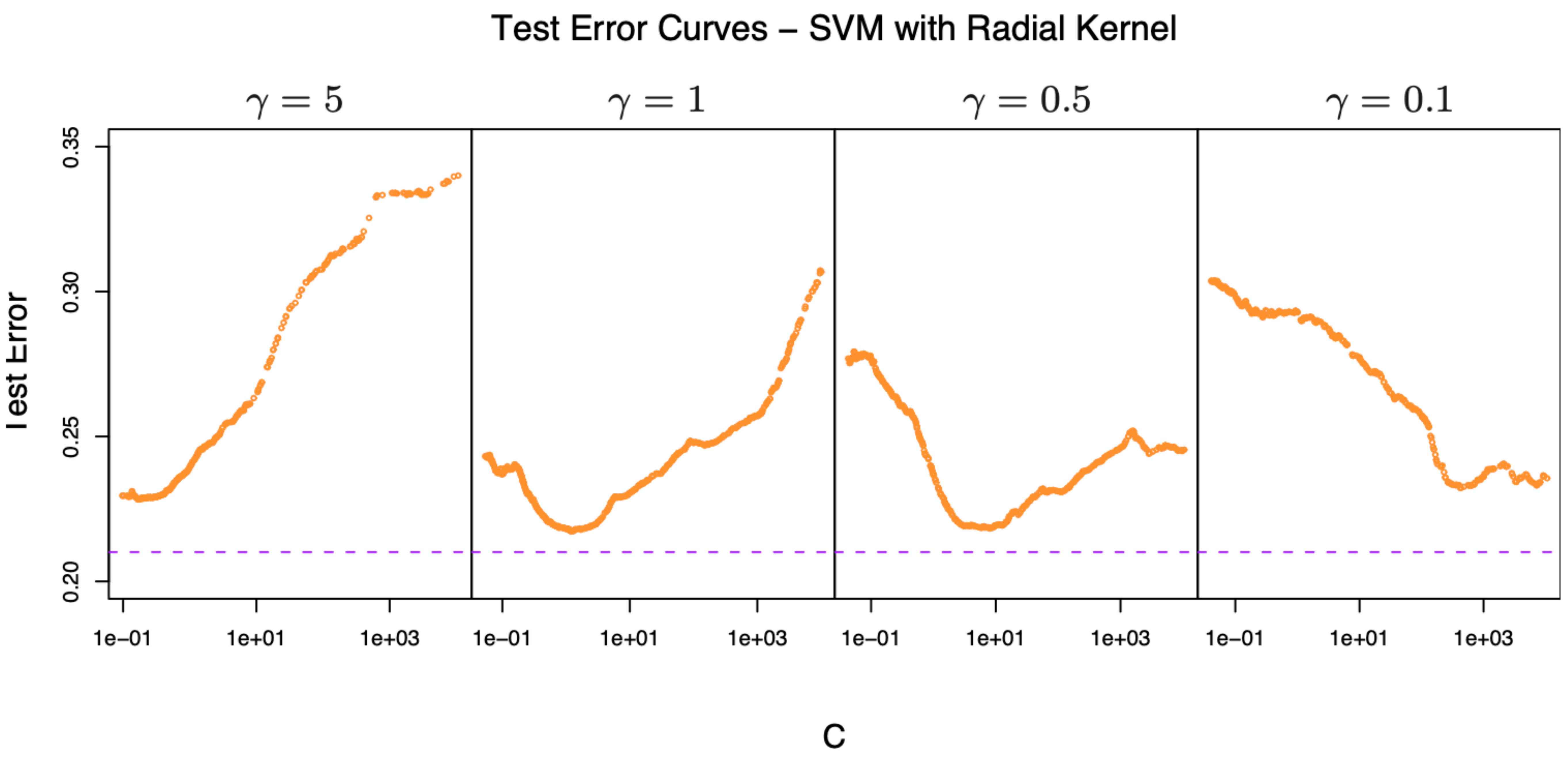
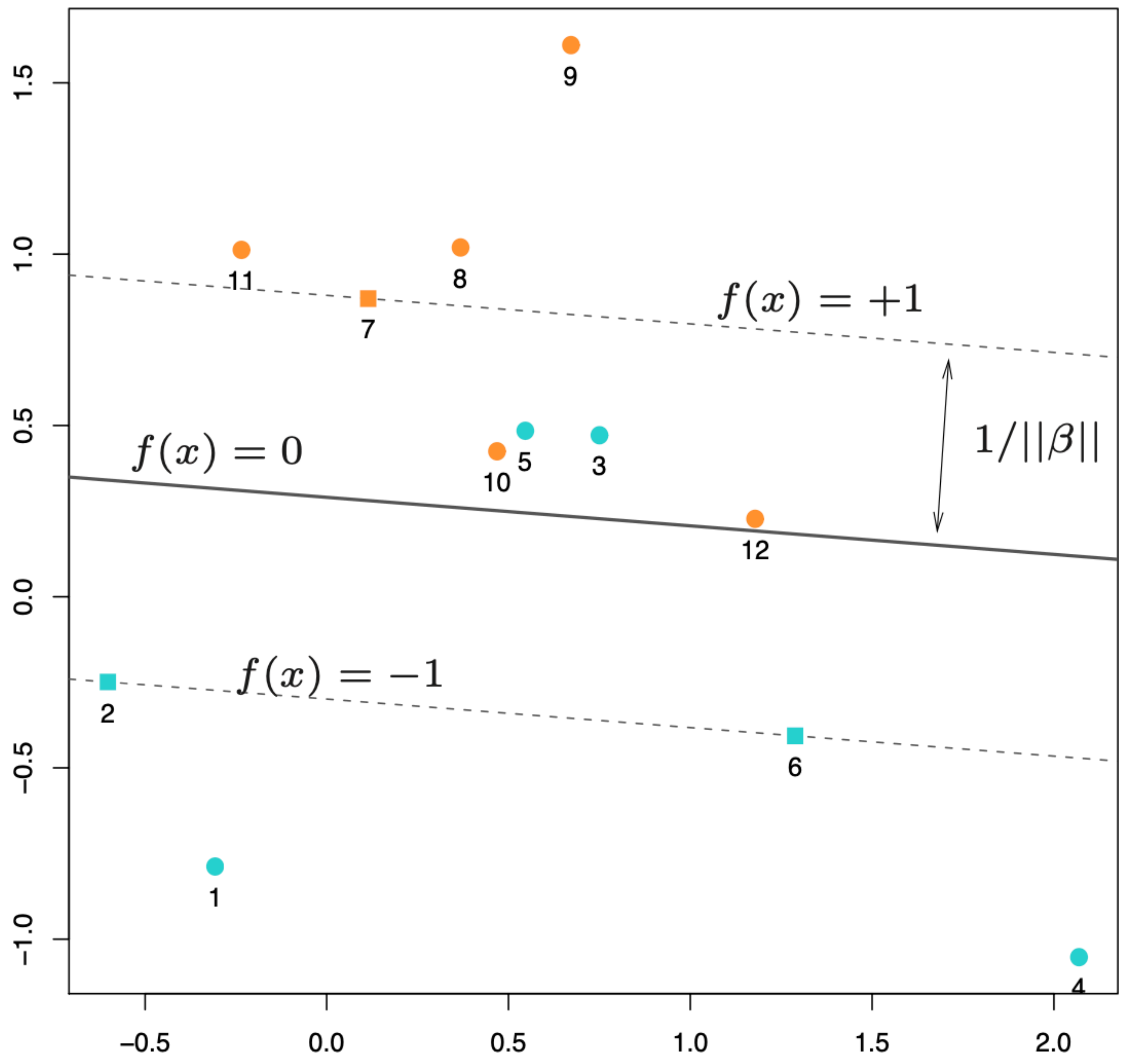


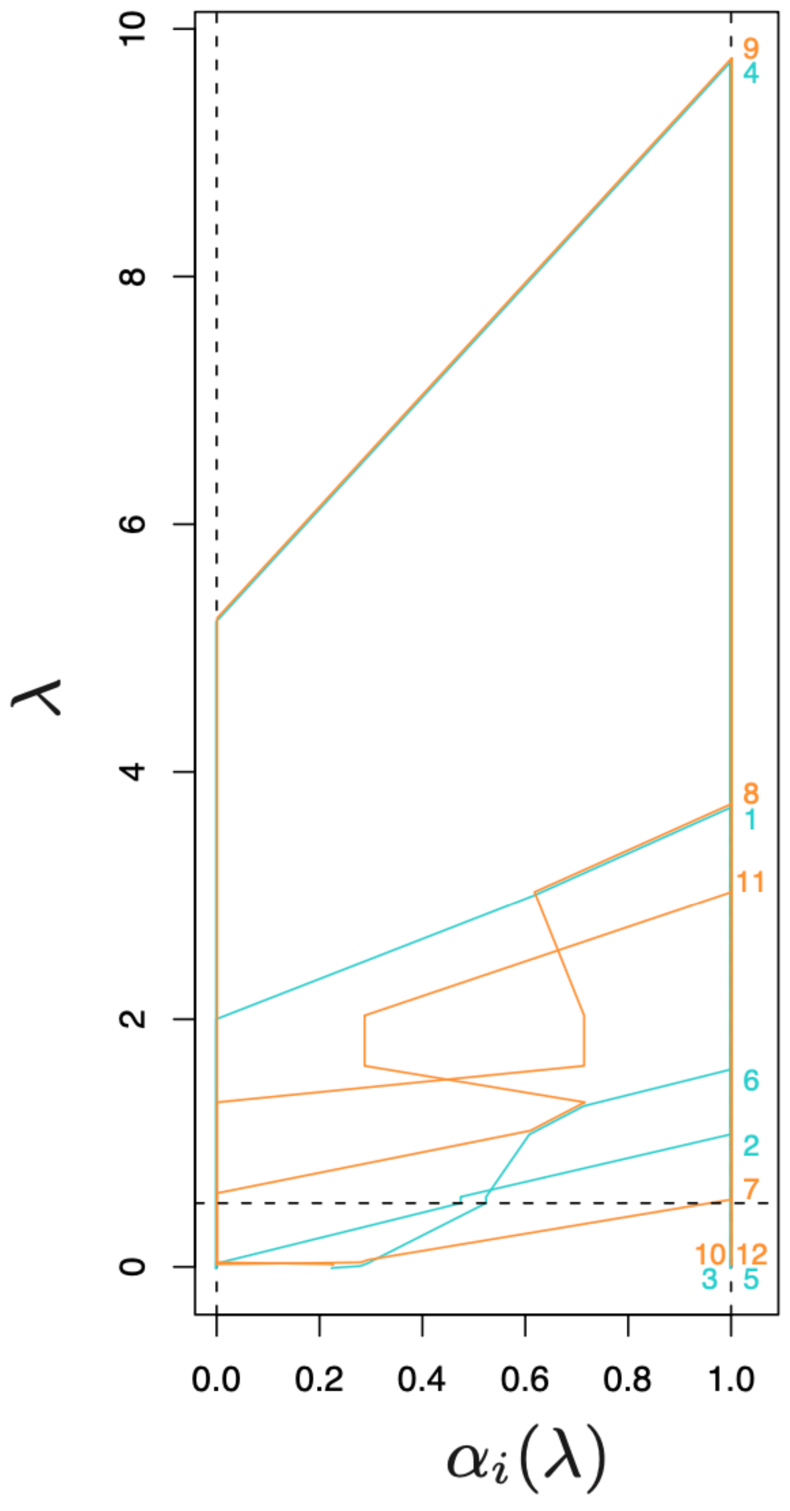
FIGURE 12.6. *Test-error curves as a function of the cost parameter C for the radial-kernel SVM classifier on the mixture data. At the top of each plot is the scale parameter γ for the radial kernel: $K_\gamma(x,y) = \exp -\gamma||x-y||^2$. The optimal value for C depends quite strongly on the scale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

The regularization parameter for the SVM is C or $1/\lambda$. To set C high, leading often to some what overfit classifiers.

We need to determine a good choice for C, perhaps by cross-validation.

A path algorithm for the SVC





$$\beta_\lambda = \frac{1}{\lambda} \sum_i \alpha_i y_i x_i,\ \alpha_i \in [0,1]\ \forall i$$

KKT optimality conditions imply that:

- $y_i f(x_i) > 1,\ \alpha_i = 0$ for correctly classified and outside their margins

- $y_i f(x_i) = 1,\ \alpha_i \in (0,1)$ for sitting on their margins

- $y_i f(x_i) < 1,\ \alpha_i = 1$ for inside their margins.

Since margin is $1/||\beta_\lambda||$, as λ decreases, margin gets narrower.

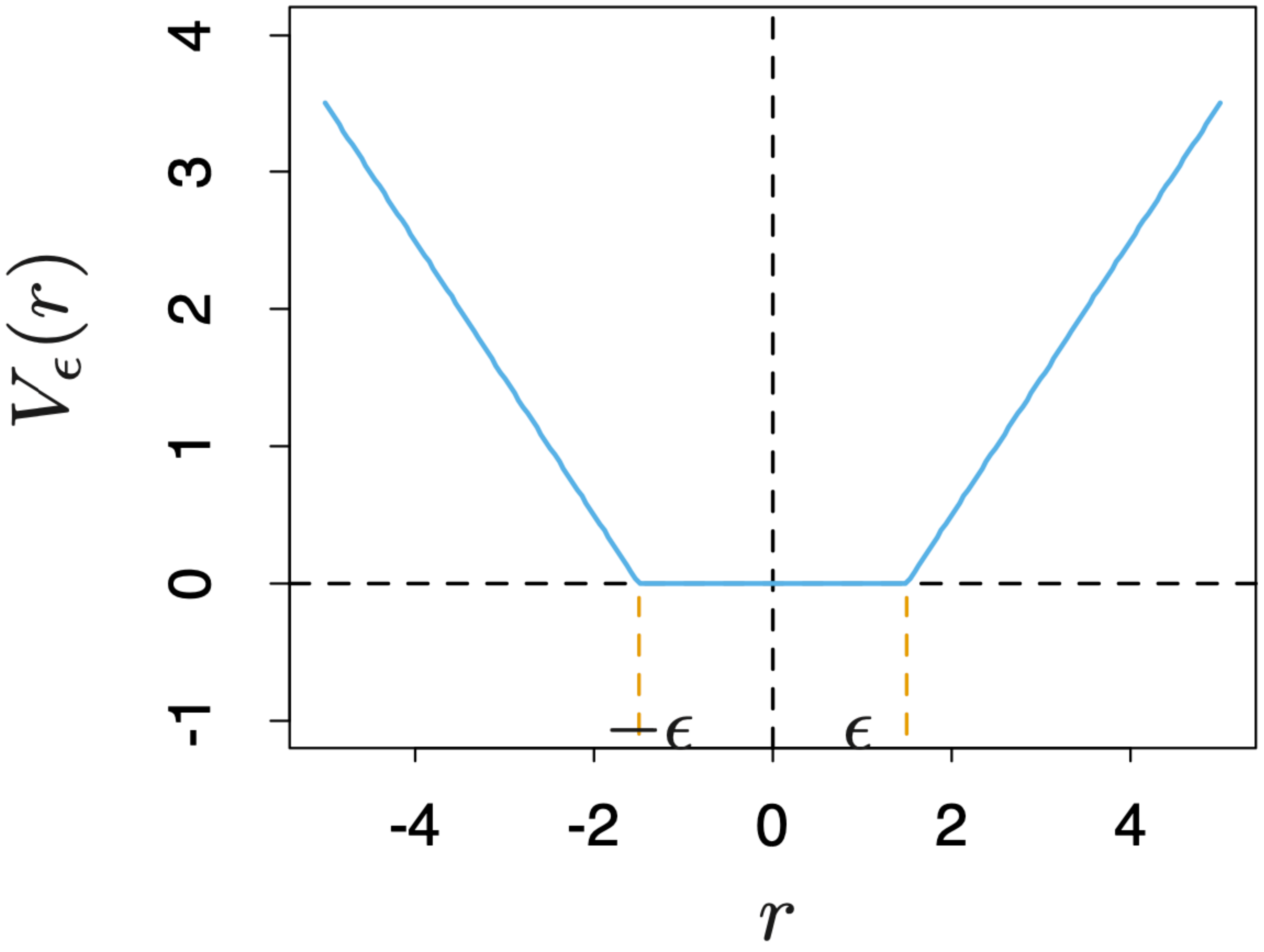
All that changes as λ decreases are the $\alpha_i \in (0,1)$ of those points on the margin.

Exactly, the same idea works for non linear models, SVMs.

SVM for regression

- For linear regression model $f(x) = x^T\beta + \beta_0$, we consider minimization of $H(\beta, \beta_0) = \sum_{i=1}^{N} V(y_i - f(x_i)) + \frac{\lambda}{2}\|\beta\|^2$ where

$V_\epsilon(r) = \begin{cases} 0 & \text{if } |r| < \epsilon, \\ |r| - \epsilon, & \text{otherwise.} \end{cases}$, "ϵ-insensitive" error measure, ignoring errors size less than ϵ. ϵ depends on the scale of r.



Rough analogy with the SVC setup, where points on the **correct side** of the decision boundary and **far away** from it, are **ignored** in the optimization.

The solution function has the form:

$$\hat{\beta} = \sum_{i=1}^{N} \underbrace{(\hat{\alpha}_i^* - \hat{\alpha}_i)}_{\text{Non-zeros are support vectors}} x_i, \quad \hat{f}(x) = \sum_{i=1}^{N} (\hat{\alpha}_i^* - \hat{\alpha}_i)\langle x, x_i \rangle + \beta_0$$

where $\hat{\alpha}_i, \hat{\alpha}_i^*$ are positive and solve

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

subject to $\alpha_i, \alpha_i^* \in [0, 1/\lambda], \alpha_i\alpha_i^* = 0 \ \forall i, \ \sum_i (\alpha_i^* - \alpha_i) = 0$

SVM for regression and kernels

- Suppose we consider approximation of the regression function in terms of a set of basis $\{h_m(x)\}_1^M$: $f(x) = \sum_{m=1}^{M} \beta_m h_m(x) + \beta_0$
- To estimate β and β_0 we minimize $H(\beta, \beta_0) = \sum_{i=1}^{N} V(y_i - f(x_i)) + \frac{\lambda}{2}\|\beta\|^2$ for some error measure $V(r)$.
- For any choice of $V(r)$, the solution has the form

$$\hat{f}(x) = \sum_m \hat{\beta}_m h_m(x) + \hat{\beta}_0 = \sum_{i=1}^{N} \hat{a}_i K(x, x_i) \quad \text{where} \quad K(x, y) = \sum_m h_m(x) h_m(y)\,.$$

- Without constant term β_0, the optimization problem re-written as $H(\beta) = V(\mathbf{y}, \mathbf{H}\beta)) + \frac{\lambda}{2}\|\beta\|^2$ where $\mathbf{H}_{im} = h_m(x_i)$
- For squared error loss, $\hat{\alpha} = (\mathbf{H}\mathbf{H}^T + \lambda\mathbf{I})^{-1}\mathbf{H}\mathbf{H}^T\mathbf{y}$. Since $\mathbf{H}\mathbf{H}^T_{ii'} = K(x_i, x_{i'})$, only the inner product kernel K need be evaluated, at the N-training points for each i, i'.