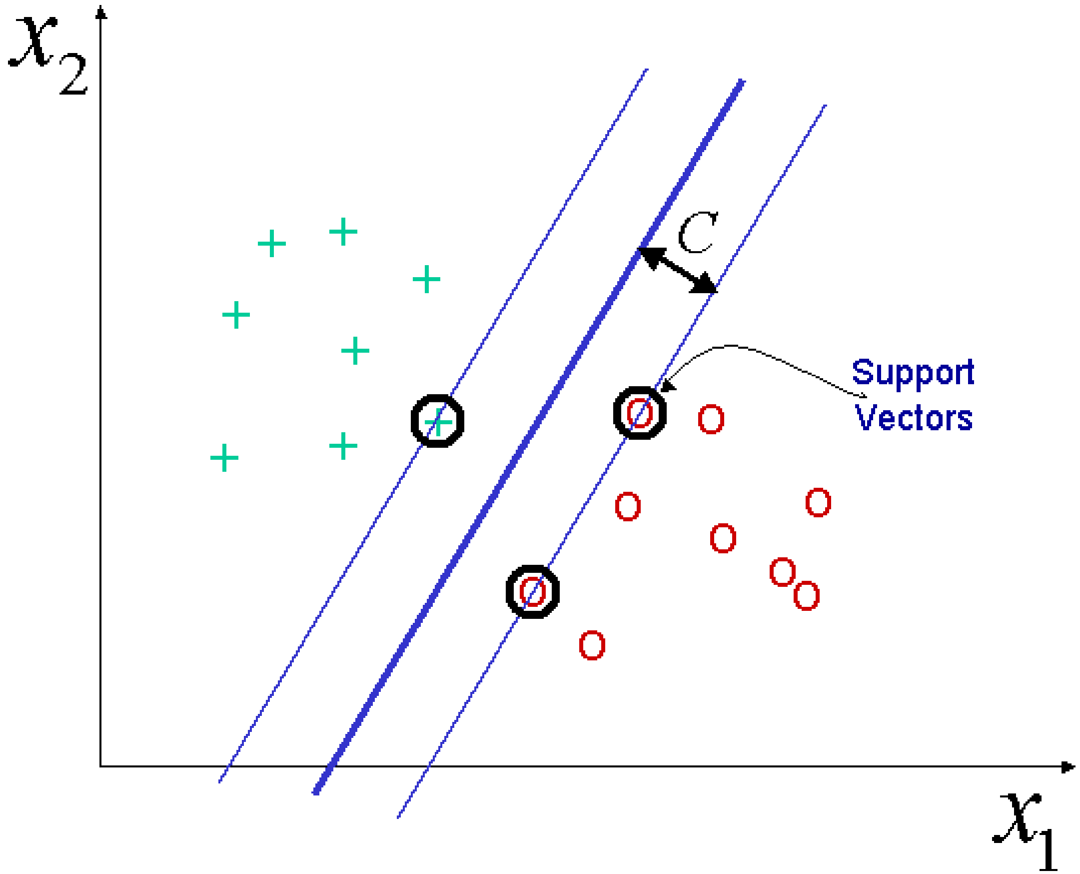
Optimal Separating Hyperplane and the Support Vector Machine

Volker Tresp
Winter 2023-2024

(Vapnik's) Optimal Separating Hyperplane

- Let's consider a linear classifier with $y_i \in \{-1, 1\}$
- If classes are linearly separable, the separating plane can be found
- Among all solutions one chooses the one that maximizes the margin $\mathcal{C}$

Optimal Separating Hyperplane(2D)



Cost Function with Constraints

- Thus we want to find a classifier

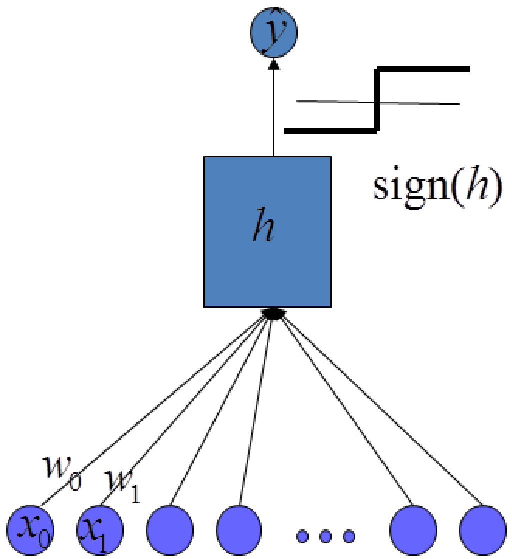
$$\widehat{y}_i = \text{sign}(h_i)$$

with

$$h_i = \sum_{j=0}^{M} w_j x_{i,j}$$

- The following inequality constraints need to be fulfilled at the solution

$$y_i h_i \geq 1 \quad i = 1, \ldots, N$$



ŷ
sign(h)
h
w0
w1
x0
x1

Maximizing the Margin

- Of all possible solutions, one chooses the one that maximizes the margin
- This can be achieved by finding the solution where the sum of the squares of the weights is minimal

$$\mathbf{w}_{opt} = \arg\min_{\mathbf{w}} \tilde{\mathbf{w}}^T \tilde{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{j=1}^{M} w_j^2$$

where $\tilde{\mathbf{w}} = (w_1, \ldots, w_M)$. (this means that in $\tilde{\mathbf{w}}$ the offset w_0 is missing); $y_i \in \{-1, 1\}$

Margin and Support-Vectors

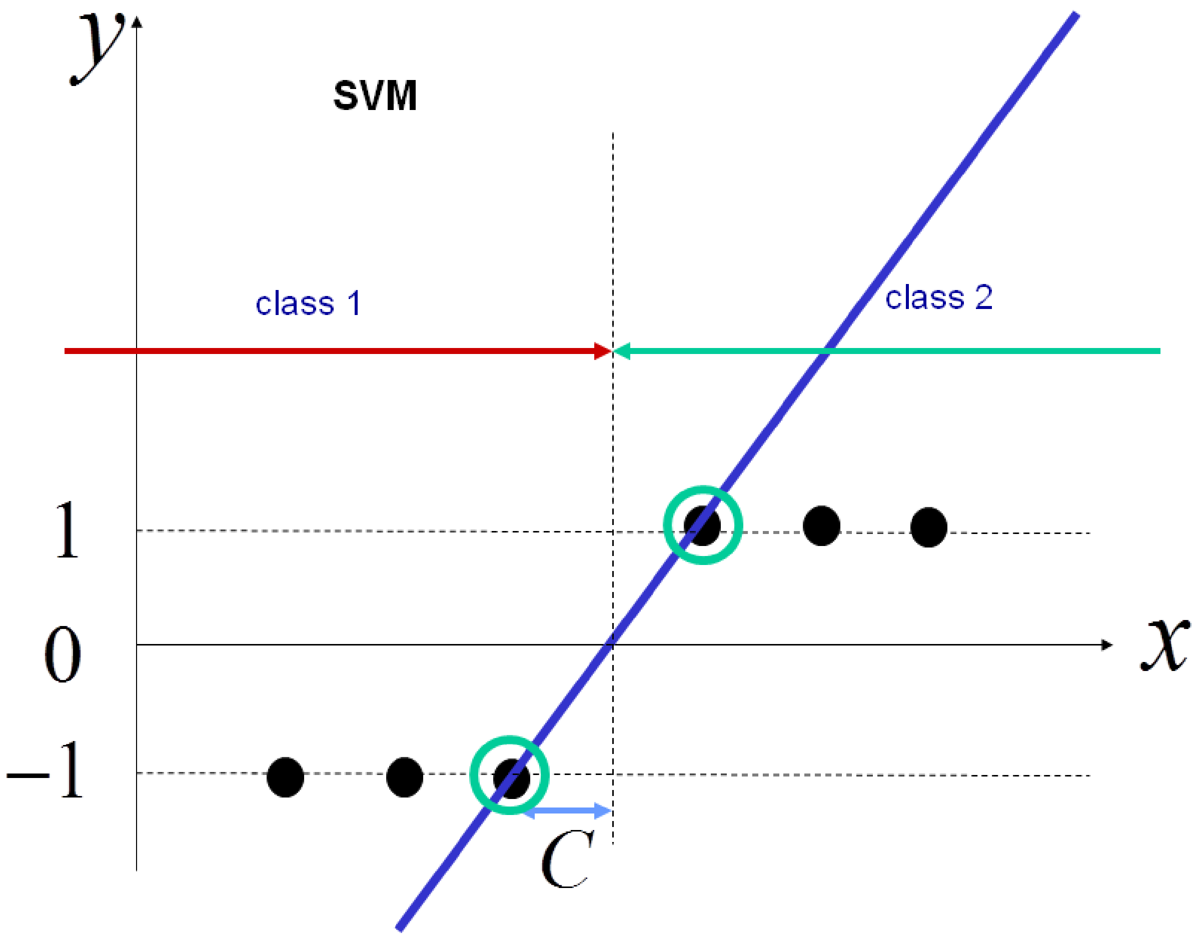
- The margin becomes

$$\mathcal{C} = \frac{1}{||\tilde{\mathbf{w}}_{opt}||}$$

- For the *support vectors* we have,

$$y_i(\mathbf{x}_i^T \mathbf{w}_{opt}) = 1$$

Optimal Separating Hyperplane (1D)



Optimization Problem

- The optimization problem is minimize

$$\tilde{\mathbf{w}}^T \tilde{\mathbf{w}}$$

under the constraint that $\forall i$

$$1 - y_i(\mathbf{x}_i^T \mathbf{w}) \leq 0$$

- We get the Lagrangian

$$L_P = \frac{1}{2}\tilde{\mathbf{w}}^T \tilde{\mathbf{w}} + \sum_{i=1}^{N} \mu_i [1 - y_i(\mathbf{x}_i^T \mathbf{w})]$$

- The Lagrangian is minimized with respect to $\mathbf{w}$ and maximized with respect to the Lagrange multipliers $\mu_i \geq 0$ (saddle point solution)

Solution

- The problem is solved via the Wolfe Dual and the solution can be written as

$$\tilde{\mathbf{w}}_{opt} = \sum_{i=1}^{N} y_i \mu_i \tilde{\mathbf{x}}_i = \sum_{i \in SV} y_i \mu_i \tilde{\mathbf{x}}_i$$

  Thus the sum is over over the terms where the Lagrange multiplier is not zero, i.e., the support vectors

Kernel Formulation

- Also we can write the solution

$$h(\mathbf{x}) = w_0 + \sum_{j=1}^{M} w_j x_j = w_0 + \tilde{\mathbf{x}}^T \tilde{\mathbf{w}} = w_0 + \tilde{\mathbf{x}}^T \sum_{i \in SV} y_i \mu_i \tilde{\mathbf{x}}_i$$

$$= w_0 + \sum_{i \in SV} y_i \mu_i \tilde{\mathbf{x}}^T \tilde{\mathbf{x}}_i = w_0 + \sum_{i \in SV} y_i \mu_i \, k(\mathbf{x}_i, \mathbf{x})$$

  Thus we get immediately a kernel solution with $k(\mathbf{x}_i, \mathbf{x}) = \tilde{\mathbf{x}}^T \tilde{\mathbf{x}}_i$.

- The solution can be written as a weighted sum over support vector kernels!

- Naturally, if one works with basis functions, one gets

$$k(\mathbf{x}, \mathbf{x}_i) = \vec{\phi}(\mathbf{x})^T \vec{\phi}(\mathbf{x}_i)$$

Non-separable Classes

- If classes are not linearly separable one needs to extend the approach. On introduces the so-called *slack* variables ξ_i

- Find

  $$\mathbf{w}_{opt} = \arg\min_{\mathbf{w}} \tilde{\mathbf{w}}^T \tilde{\mathbf{w}}$$

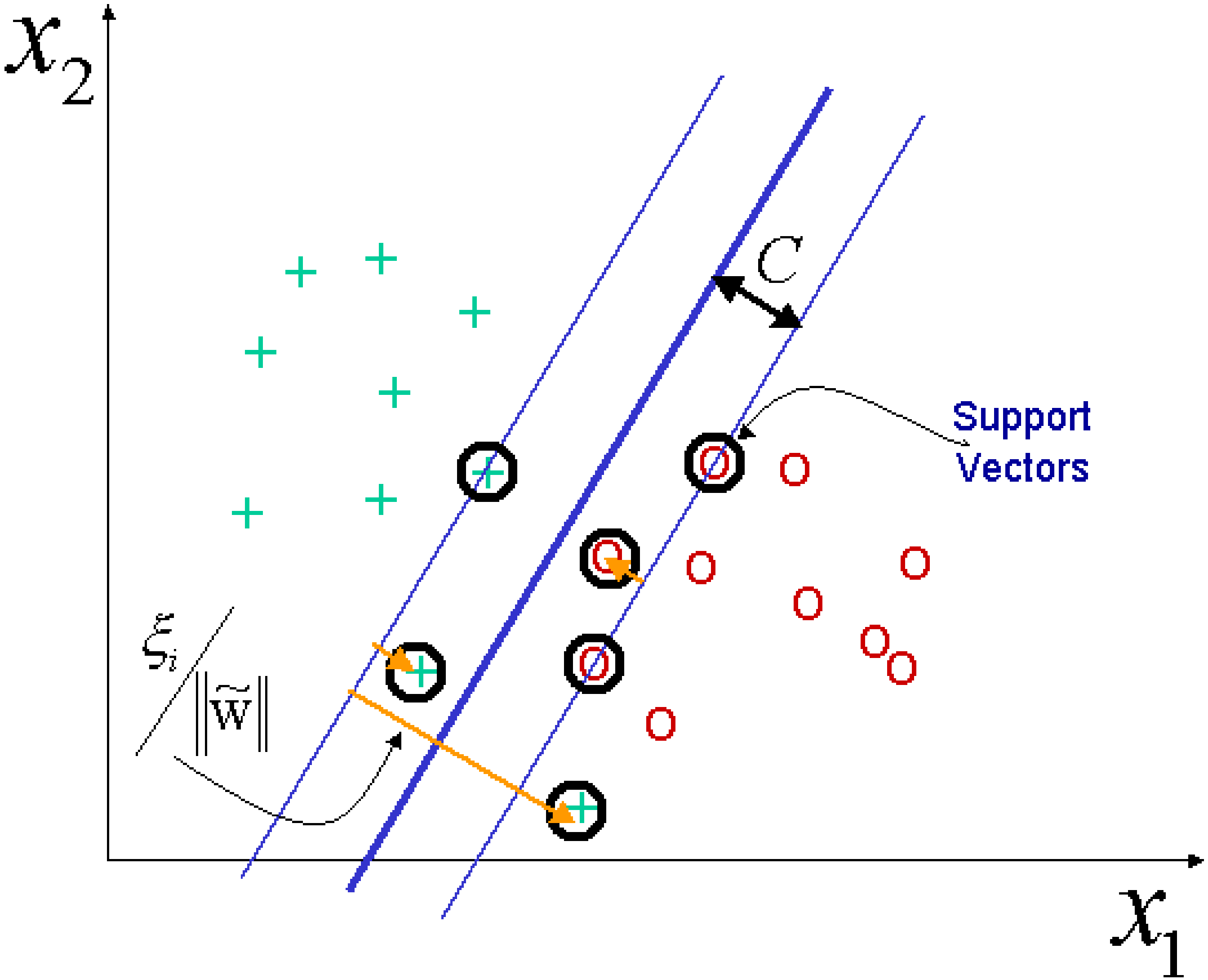
  under the constraint that

  $$y_i(\mathbf{x}_i^T \mathbf{w}) \geq 1 - \xi_i \quad i = 1, \ldots, N$$

  and

  $$\xi_i \geq 0 \text{ where } \sum_{i=1}^{N} \xi_i \leq 1/\gamma$$

- The smaller $\gamma > 0$, the more slack is permitted. For $\gamma \rightarrow \infty$, one obtains the hard constraint



x_2
x_1
C
Support Vectors
$\frac{\xi_i}{\|\tilde{w}\|}$

Optimization

- The optimal separating hyperplane is found via an evolved optimization of the quadratic cost function with linear constraints
- γ is a hyperparameter

Optimization via Penalty Method

- We define

$$\arg\min_{\mathbf{w}} 2\gamma \sum |1 - y_i(\mathbf{x}_i^T \mathbf{w})|_+ + \tilde{\mathbf{w}}^T \tilde{\mathbf{w}}$$

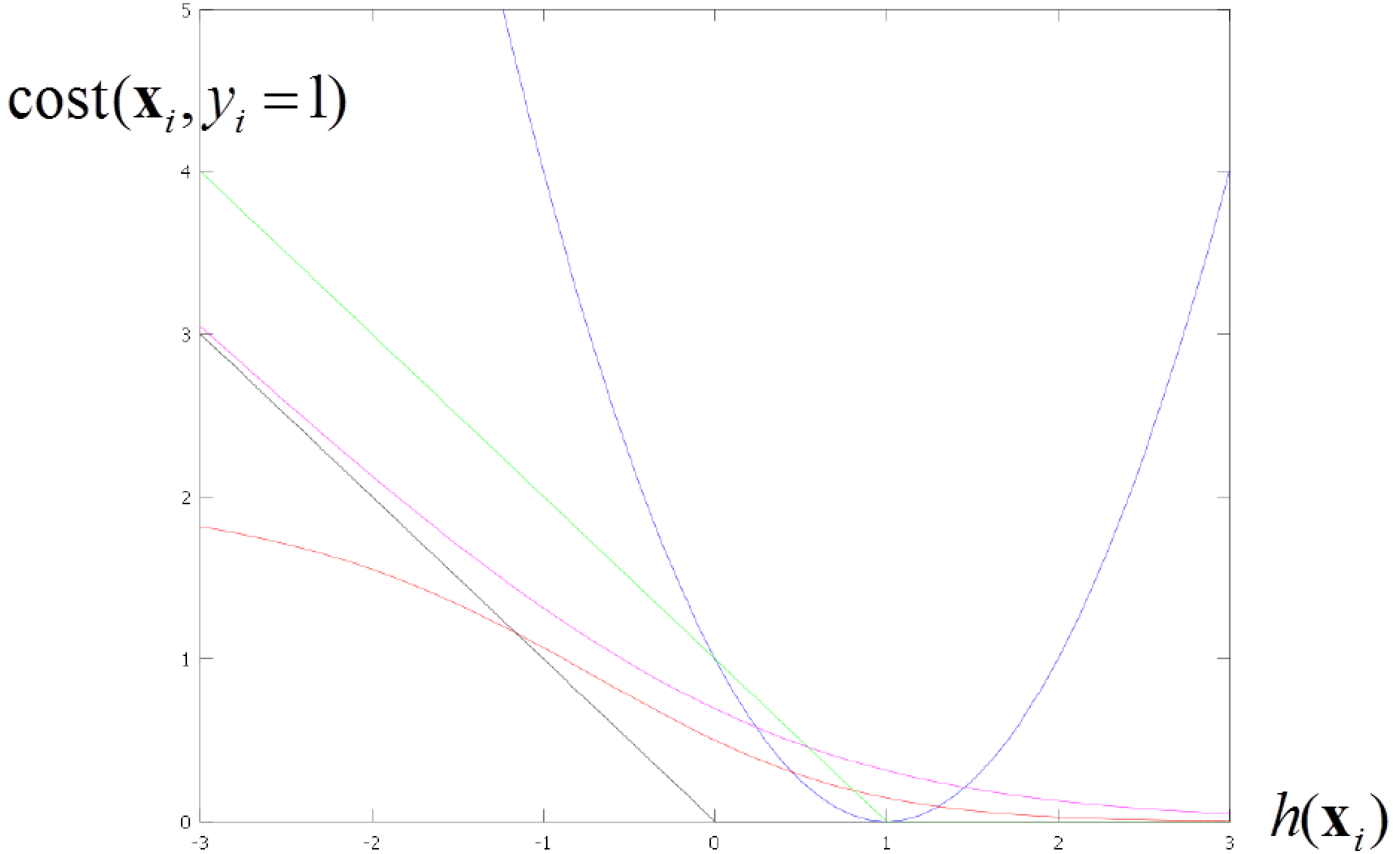
  where $\sum |1 - y_i(\mathbf{x}_i^T \mathbf{w})|_+$ is the penalty term

  Here, $|arg|_+ = \max(arg, 0)$.

- With a finite γ, slack is permitted

Comparison of Cost Functions

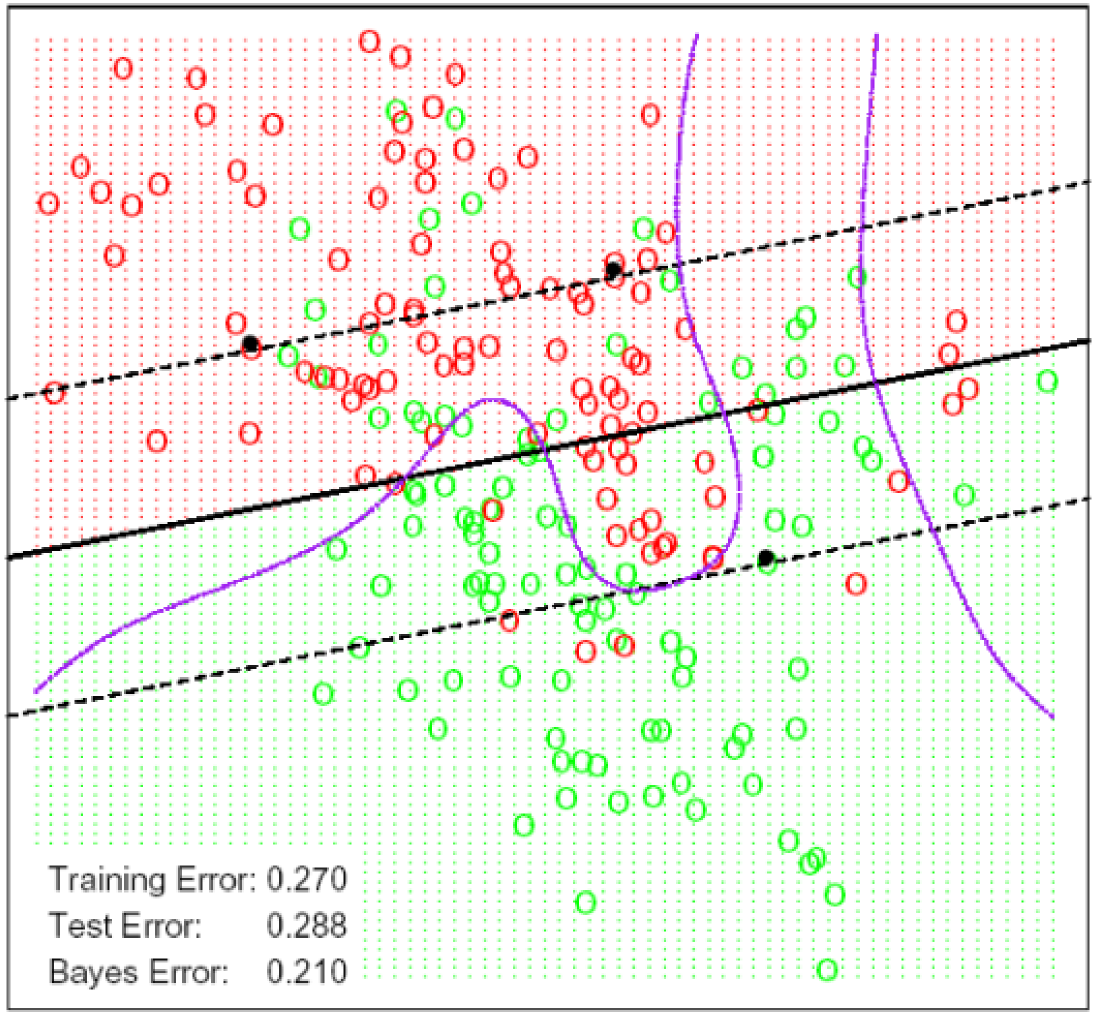
- We consider a data point of class 1 and the contribution of one data point to the cost function/negative log-likelihood
- The contribution of $\mathbf{x}_i$ to the cost is:
  - Least squares (blue) : $\mathrm{cost}(\mathbf{x}_i, y_i = 1) = (1 - h(\mathbf{x}_i))^2$
  - Perceptron (black) $\mathrm{cost}(\mathbf{x}_i, y_i = 1) = |-h(\mathbf{x}_i)|_+$
  - Vapnik's optimal hyperplane (green): $\mathrm{cost}(\mathbf{x}_i, y_i = 1) = \gamma|1 - h(\mathbf{x}_i)|_+$
  - Logistic Regression (magenta): $\mathrm{cost}(\mathbf{x}_i, y_i = 1) = \log(1 + \exp(-h(\mathbf{x}_i)))$
  - Neural Network (red): $\mathrm{cost}(\mathbf{x}_i, y_i = 1) = (1 - \mathrm{sig}(h(\mathbf{x}_i)))^2$



$\text{cost}(\mathbf{x}_i, y_i = 1)$
$h(\mathbf{x}_i)$
5
4
3
2
1
0
-3
-2
-1
0
1
2
3

Toy Example

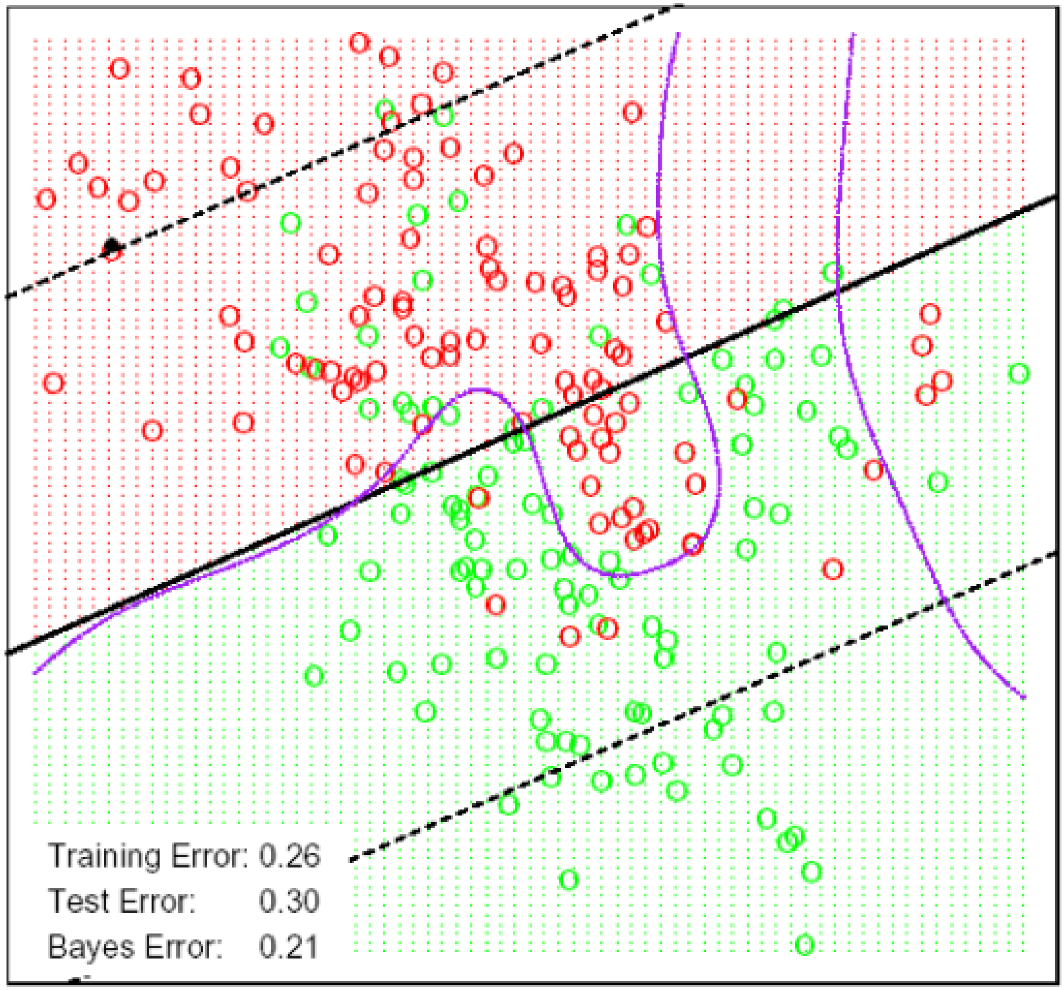
- Data for two classes (red, green) are generated
- Classes overlap
- The true class boundary is in violet
- The continuous line is the separating hyperplane found by the linear SVM
- All data points within the dotted region are support vectors (62% of all data points)
- γ is large (little slack is permitted)



$\gamma = 10000$

Toy Example (cont'd)

- Linear SVM with small γ: the solution has many more support vectors (85% of all data points)
- The test error is almost the same

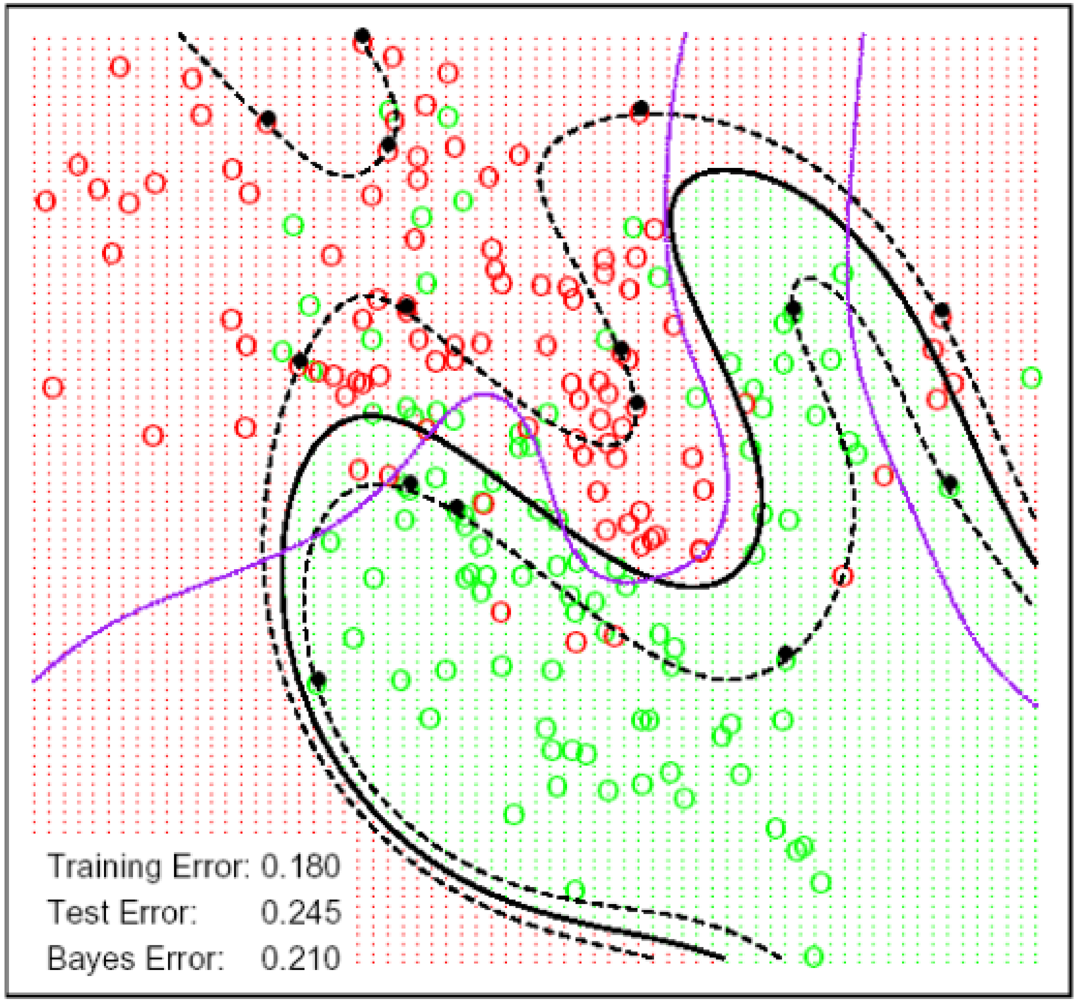


$\gamma = 0.01$

Toy Example (cont'd)

- With polynomial kernels
- The test error is reduced since the fit is better
- Note that although the support vectors are close to the separating plane in the basis function space, this is not necessarily true in input space

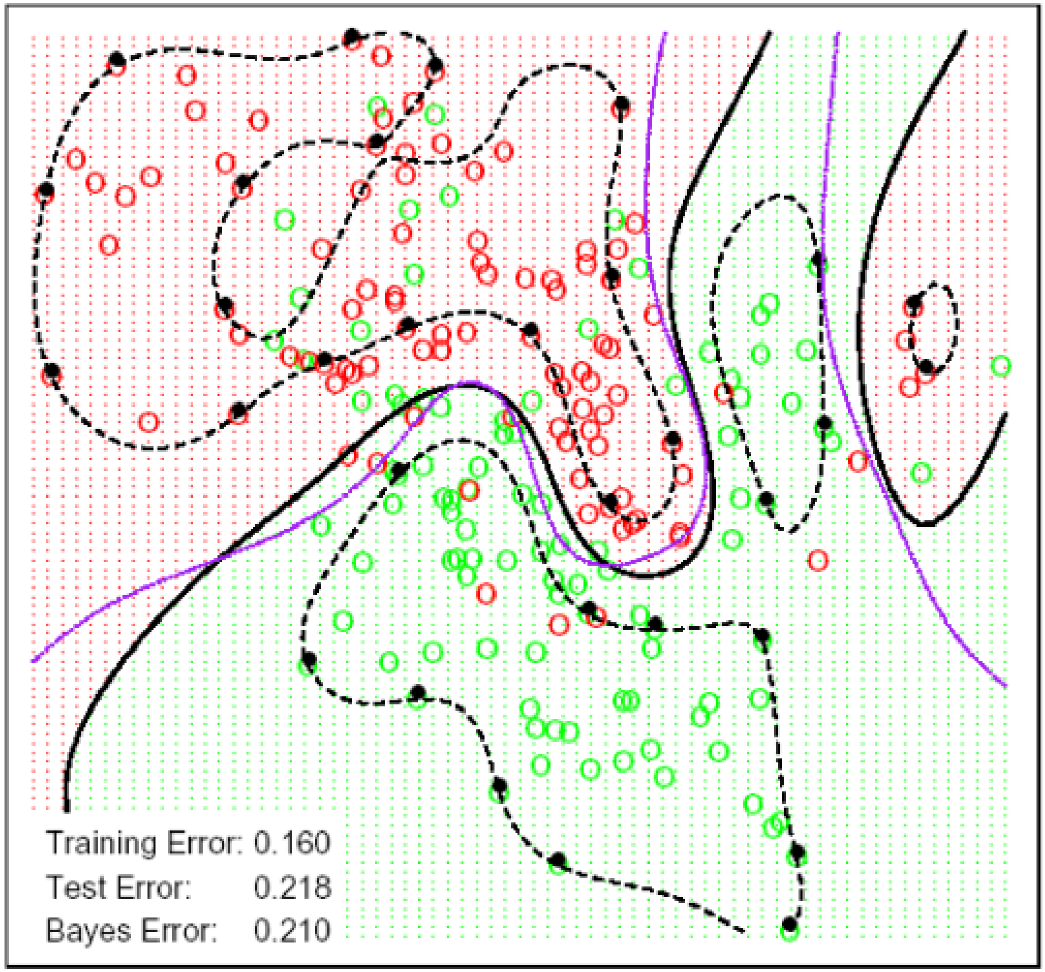
SVM - Degree-4 Polynomial in Feature Space



Toy Example (cont'd)

- Gaussian kernels give the best results
- Most data points are support vectors

SVM - Radial Kernel in Feature Space



Comments

- The ideas of searching for solutions with a large margin has been extended to many other problems

Optimal Separating Hyperplane and Perceptron

- Vapnik was interested in solutions which focus on the currently misclassified examples such as the Perceptron. He could preserve this idea also in the Optimal Separating Hyperplanes
- The Perceptron requires

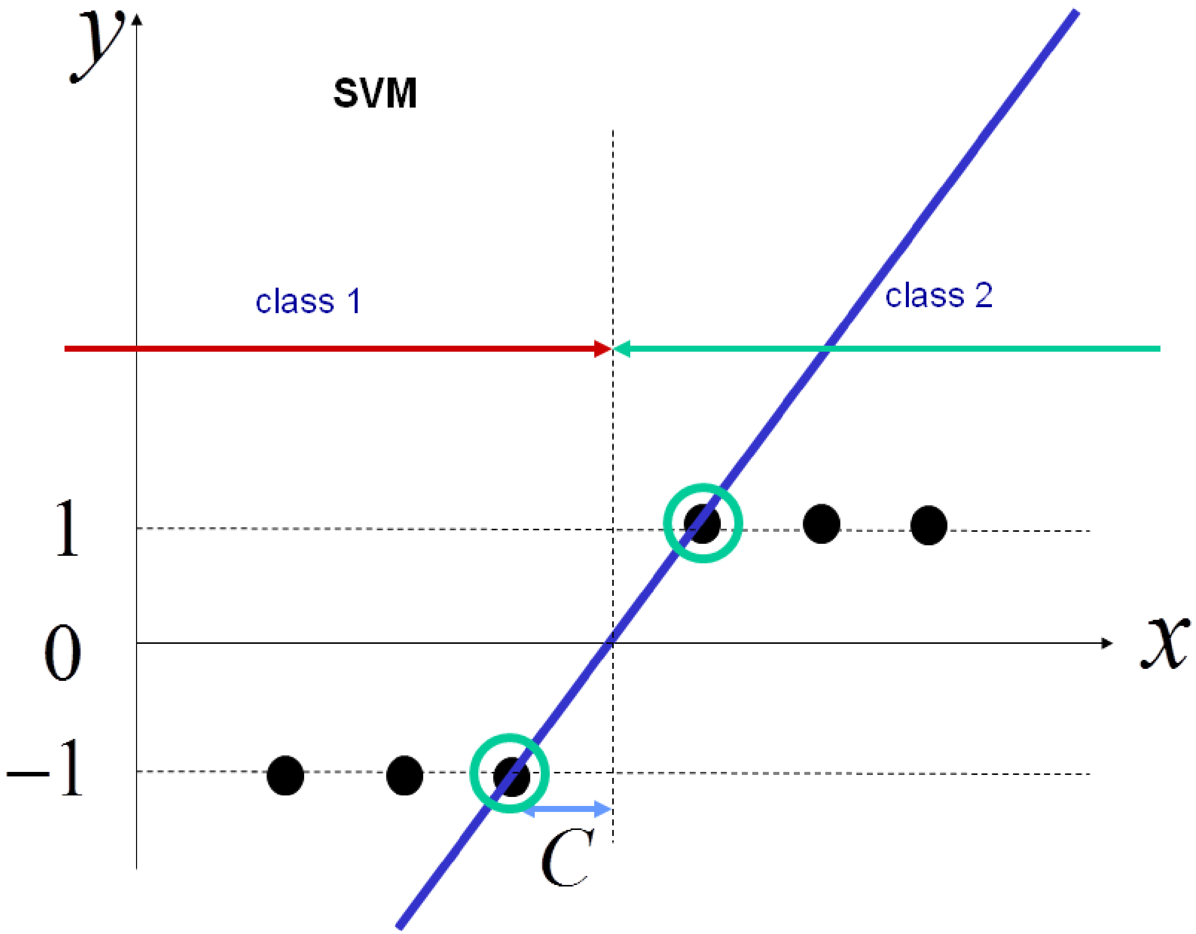
$$y_i h_i \geq 0 \quad i = 1, \ldots, N$$

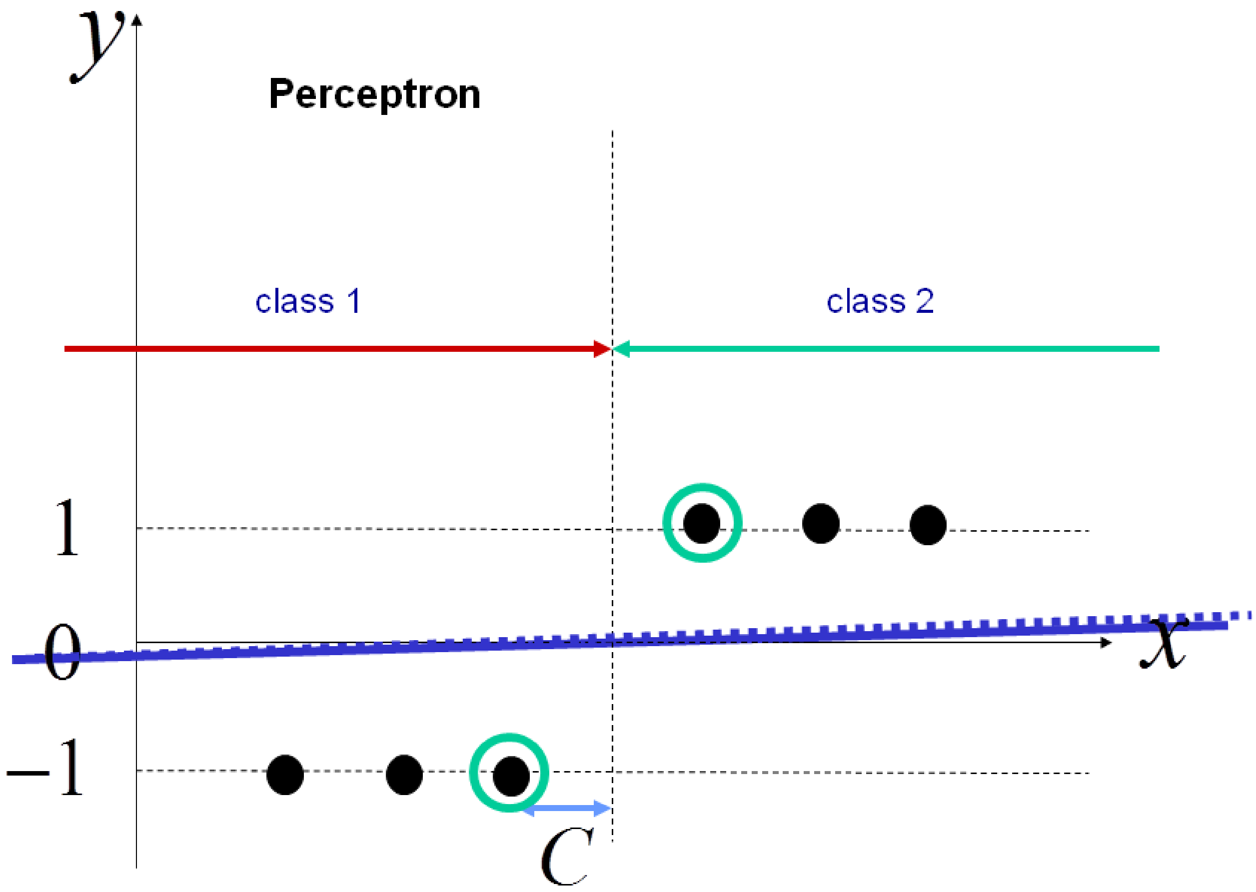
  which leads to non-unique solutions (even when weight decay is added)

- The important insight of Vapnik was that requiring

$$y_i h_i \geq 1 \quad i = 1, \ldots, N$$

  leads to unique solutions when weight decay is added


y
SVM
class 1
class 2
1
0
−1
x
C


y
Perceptron
class 1
class 2
1
0
x
−1
C