SUPPORT VECTOR CLASSIFICATION

Ying-Chao Hung

Department of Statistics
National Chengchi University
E-mail: ***hungy@nccu.edu.tw***

Support Vector Machines (SVM)

- Support Vector Machines have been widely used in the area of *machine learning*, *pattern recognition*, *time series analysis*, and *regression*.

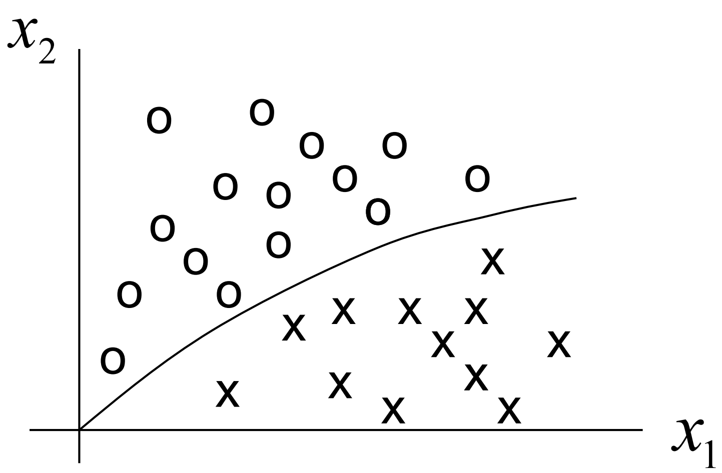
- SVM originates from the framework of statistical learning theory (see Boser et al. 1992, Cortes et al. 1995, Guyon et al. 1993, Scholkopf 1997, Vapnik 1998)

Support Vector Machines (SVM)

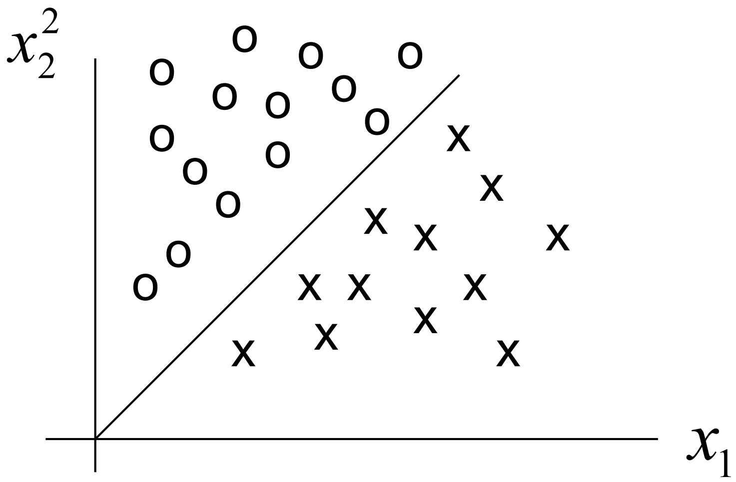
- **Idea:**

  To use a *linear hyperplane* to create a classifier in a *feature space*.

- **Examples:**



做完轉換後



A 2-class Example

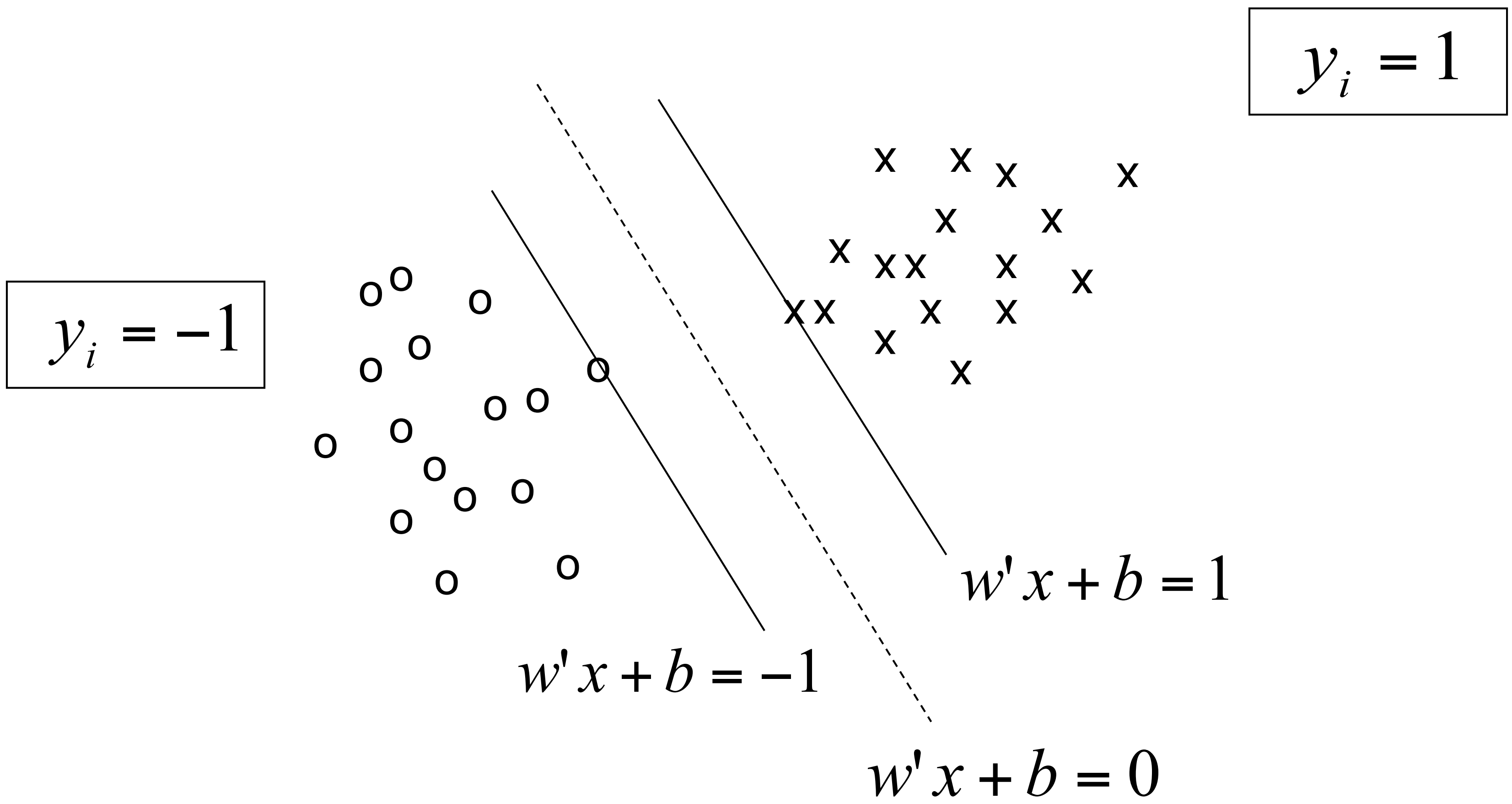
- Given training vectors x_i of length n, $i = 1, \ldots, l$.

  # of attributes

  The class of object i is defined as

$$y_i = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ in class 1,} \\ -1 & \text{if } \mathbf{x}_i \text{ in class 2,} \end{cases}$$

A Linearly Separable 2-class Example



The Bounding Planes

- $w'x + b = 1$ and $w'x + b = -1$ are called: *bounding planes*.

- **Goal of SVM:**

  To find the line $w'x + b = 0$ (i.e., find w and b) such that *the margin of two bounding planes is maximized* !

The Optimization Problem

- It is easy to show that the *margin* of two bounding planes is $2/\|w\|$, thus,

$$\text{maximize } \frac{2}{\|w\|} \equiv \text{minimize } \frac{\|w\|}{2} \equiv \text{minimize } \frac{\|w\|^2}{2}$$

- This is equivalent to finding the solution of the following optimization problem:

$$\begin{cases} \underset{w,b}{\text{minimize}} & \dfrac{\|w\|^2}{2} \\ \text{subject to} & y_i(w'x+b) \geq 1, \quad i = 1, \ldots, l. \end{cases}$$

避免w=0, 加入限制為要分對

Some Remarks

- The idea of maximizing the margin 2/||w|| is based on Vapnik's Structural Risk Minimization, which is a *convex (or quadratic) optimization problem*. (see Vapnik, 1998)

- Note that the constraints

$$y_i(w' x_i + b) \geq 1 \quad \text{for all } i$$

guarantee that the two classes are *linearly separable*.

Solving the Optimization Problem

- Using *Lagrange Multiplier Method*, consider

$$L(w,b,\alpha_i) = \underbrace{\frac{\|w\|^2}{2}}_{\text{objective}} - \sum_{i=1}^{l} \alpha_i \big[\underbrace{y_i(w'x+b)-1}_{\text{constraints}}\big]$$

➔ Minimize L by choosing $w,\ b,$ and α_i.

- Taking first derivatives w.r.t. w and b yields

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w - \sum_{i=1}^{l} \alpha_i y_i x_i = 0$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{l} \alpha_i y_i = 0$$

Karuah-Kahn Tucker (KKT) Conditions

- All above constraints can be summarized as:

$$
\begin{cases}
w - \sum_{i=1}^{l} \alpha_i y_i x_i = 0 \\
\sum_{i=1}^{l} \alpha_i y_i = 0 \\
y_i (w' x + b) - 1 \geq 0 \\
\alpha_i \geq 0
\end{cases}
$$

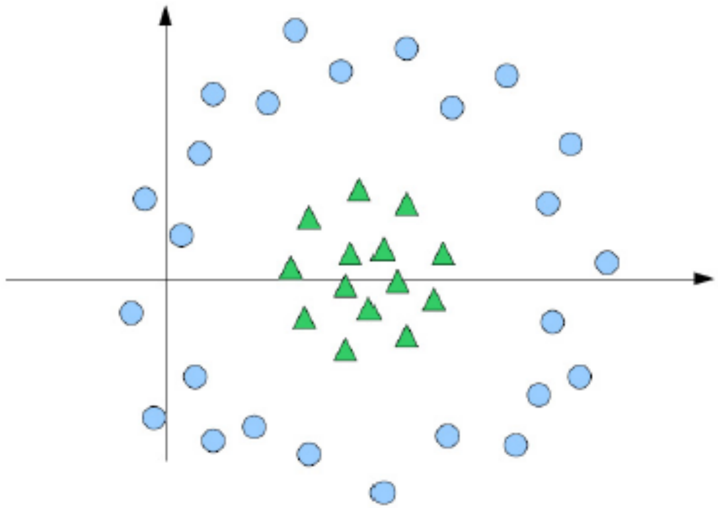
➔ Necessary and sufficient for solving the problem.

The Duel Problem

- Based on the KKT conditions, the original optimization problem can be written as:

$$\begin{cases} \underset{\alpha_{\mathrm{i}}}{\text{maximize}} \ \sum_{i=1}^{l} \alpha_{\mathrm{i}} - \frac{1}{2} \sum_{i,j} \alpha_{\mathrm{i}} \alpha_{\mathrm{j}} y_i y_j < x_i, x_j > \\ \text{subject to the KKT conditions} \end{cases}$$

Linearly Non-separable Examples



- Note that for this example, there exists no (w, b) that can satisfy the constraints $y_i(w'x + b) \geq 1 \text{ for all } i$.

  (i.e. Hyperplanes can not perfectly separate classes)

Slack Variables

- To overcome the linearly non-separable cases, for each object *i* we introduce the following *slack variable* ξ_i : (Cortes and Vapnik, 1995)
- Define 若分對 這項為≥1 -> max(0,-)=0

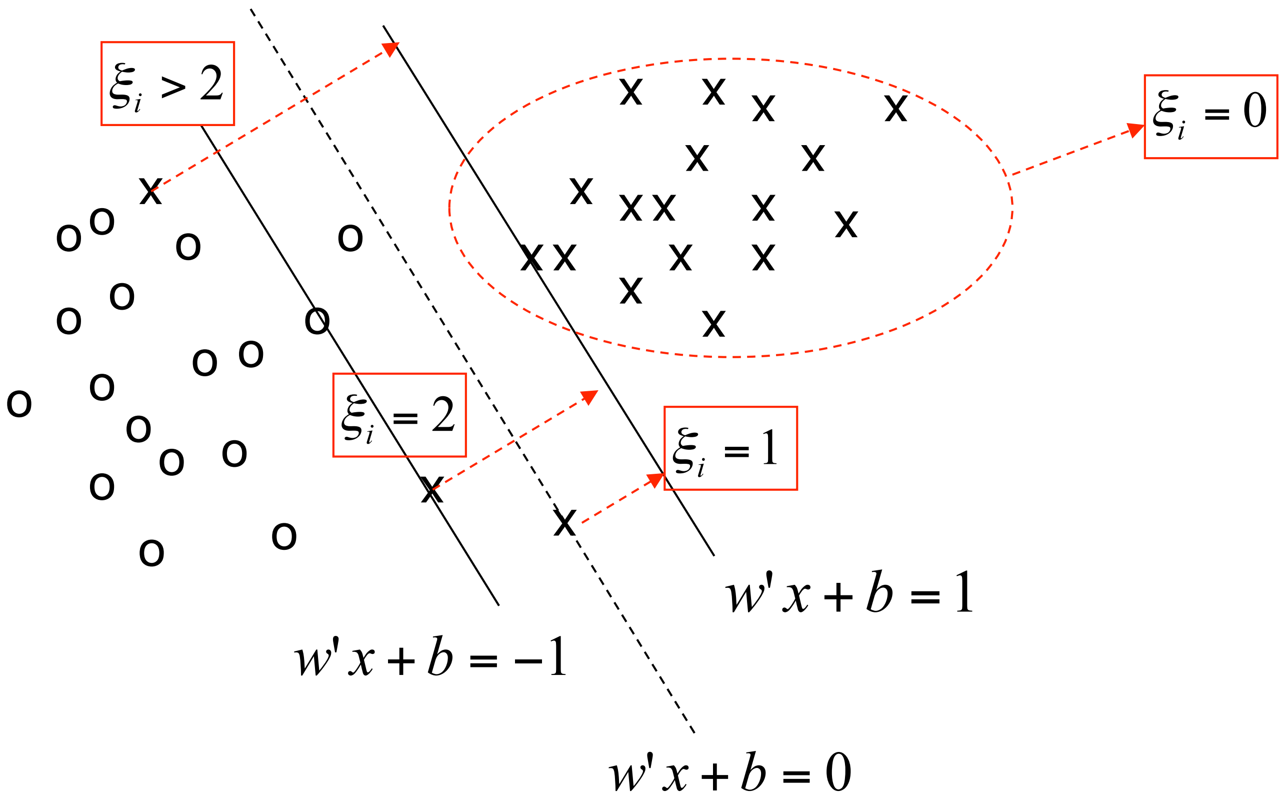
$$\xi_i = \max\{0, 1 - y_i(w'x + b)\} \text{ for } i = 1, \ldots, l.$$

➔ This can be viewed as *the penalty of wrong classification* for each object. 分錯的懲罰

- $\xi_i = 0 \Rightarrow$ object *i* has a right classification

  $\xi_i > 0 \Rightarrow$ object *i* has a wrong classification

Illustration of Slack Variables



Some Remarks

- Note that the larger the slack variable ξ_i is, the further object i is away from the bounding plane.

- By introducing the slack variables ξ_i , we can *justify the constraints* in the primal optimization problem as:

$$\boxed{\begin{cases} y_i(w'x_i + b) \ge 1 - \xi_i \\ \xi_i \ge 0 \quad \text{for all } i = 1, \dots, l. \end{cases}}$$

調整限制式

The Optimization Problem

- The corresponding optimization problem can then be written as:

$$\begin{cases} \underset{w,b,\xi_i}{\text{minimize}} & \dfrac{\|w\|^2}{2} + C \cdot \sum_{i=1}^{l} \xi_i \\ \text{subject to} & y_i(w'x_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad \text{for all } i = 1, \ldots, l \end{cases}$$

- Note that the constraints allows the training data may not be on the correct side of the bounding plane.

Some Remarks

- Note that $C > 0$ is the penalty parameter, which accounts for the *tradeoff* between the "*margin length*" and the "*classification accuracy*".

- If data are linearly separable, can prove that when C is larger than a certain number, the new problem reduces to the original problem and all ξ_i are zero.

  (Lin, 2001)

Nonlinear SVM: Mapping data into a higher dimensional space

- In practice, data can be distributed in a highly nonlinear way (recall the previous example).
  ➔ Using merely linear function may result in poor performance in classification accuracy.

- Instead of modeling linear curves, one can map data into a higher-D space, e.g., using a mapping function

$$\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \cdots).$$

Examples

- Example 1: Mapping $\mathbf{x}$ form $\mathcal{R}^3$ to $\mathcal{R}^{10}$:

$$\phi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_3, x_1^2, x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3)$$

- Example 2: Mapping $x \in \mathcal{R}^1$ to an infinite dimensional space:

$$\phi(x) = \left[1, \frac{x}{1!}, \frac{x^2}{2!}, \frac{x^3}{3!}, \cdots\right]$$

Back To The Primal Optimization Problem

- The corresponding optimization problem can then be written as:

$$\begin{cases} \underset{w,b,\xi_i}{\text{minimize}} \quad \dfrac{\| w \|^2}{2} + C \cdot \sum_{i=1}^{l} \xi_i \\ \text{subject to} \quad y_i (w' \phi(x_i) + b) \geq 1 - \xi_i \\ \qquad\qquad\quad \xi_i \geq 0 \quad \text{for all } i = 1, \ldots, l \end{cases}$$

The New Duel Problem

- After some algebra, the corresponding dual problem can be summarized as:

$$
\begin{cases}
\underset{\alpha_{\mathrm{i}}}{\text{maximize}} & \displaystyle\sum_{i=1}^{l} \alpha_{\mathrm{i}} - \frac{1}{2} \sum_{i,j} \alpha_{\mathrm{i}} \alpha_{\mathrm{j}} y_i y_j < \phi(x_i), \phi(x_j) > \\
\text{subject to} & 0 \le \alpha_i \le C, \ i = 1, \ldots l, \\
 & \displaystyle\sum_{i=1}^{l} y_i \alpha_i = 0
\end{cases}
$$

Some Remarks

- Note that the optimal solution is related to data only through the *dot product* (or *inner product*) $< \phi(x_i), \phi(x_j) >$.
- Therefore, instead of considering explicitly the mapping function $\phi(\cdot)$, we can use a "*kernel function*" to represent the inner product:

$$K(x_i, x_j) = < \phi(x_i), \phi(x_j) > .$$

Popular Choices of Kernel Functions

- Gaussian kernel or Radial bassis function (RBF) kernel:
  $$\exp\left\{-\gamma \| x_i - x_j \|^2\right\}.$$
- Polynomial kernel: $(\mathbf{x}_i^T \mathbf{x}_j / \gamma + \delta)^d$.
- Sigmoid kernel: $\tanh(\delta \mathbf{x}_i^T \mathbf{x}_j + \gamma)$.
- Where γ, δ and d are kernel parameters.

Illustration of RBF Kernel

Assume $x \in \mathcal{R}^1$ and $\gamma > 0$.

$$
\begin{aligned}
e^{-\gamma(x_i - x_j)^2} &= e^{-\gamma x_i^2 + 2\gamma x_i x_j - \gamma x_j^2} \\
&= e^{-\gamma x_i^2 - \gamma x_j^2}\left(1 + \frac{2\gamma x_i x_j}{1!} + \frac{(2\gamma x_i x_j)^2}{2!} + \frac{(2\gamma x_i x_j)^3}{3!} + \cdots\right) \\
&= e^{-\gamma x_i^2 - \gamma x_j^2}\left(1 \cdot 1 + \sqrt{\frac{2\gamma}{1!}}x_i \cdot \sqrt{\frac{2\gamma}{1!}}x_j + \sqrt{\frac{(2\gamma)^2}{2!}}x_i^2 \cdot \sqrt{\frac{(2\gamma)^2}{2!}}x_j^2\right. \\
&\qquad \left. + \sqrt{\frac{(2\gamma)^3}{3!}}x_i^3 \cdot \sqrt{\frac{(2\gamma)^3}{3!}}x_j^3 + \cdots\right) \\
&= \langle \phi(x_i), \phi(x_j) \rangle.
\end{aligned}
$$

where

$$
\phi(x) = e^{-\gamma x^2}\left[1, \overbrace{\sqrt{\frac{2\gamma}{1!}}x, \sqrt{\frac{(2\gamma)^2}{2!}}x^2, \sqrt{\frac{(2\gamma)^3}{3!}}x^3, \cdots}^{\text{Infinite dimension}}\right]
$$

Some Remarks

- Theoretically, choosing appropriate γ in RBF kernel can result a perfect classification

  (i.e., apparent error rate = 0).

- ***The RBF kernel is the most popular choice for SVM beginners*** since:

  (i) It can easily handle complex data by mapping them into high-D (or infinite-D) spaces.

  (ii) It has relatively few parameter(s) that has to be determined before solving the optimization problem.

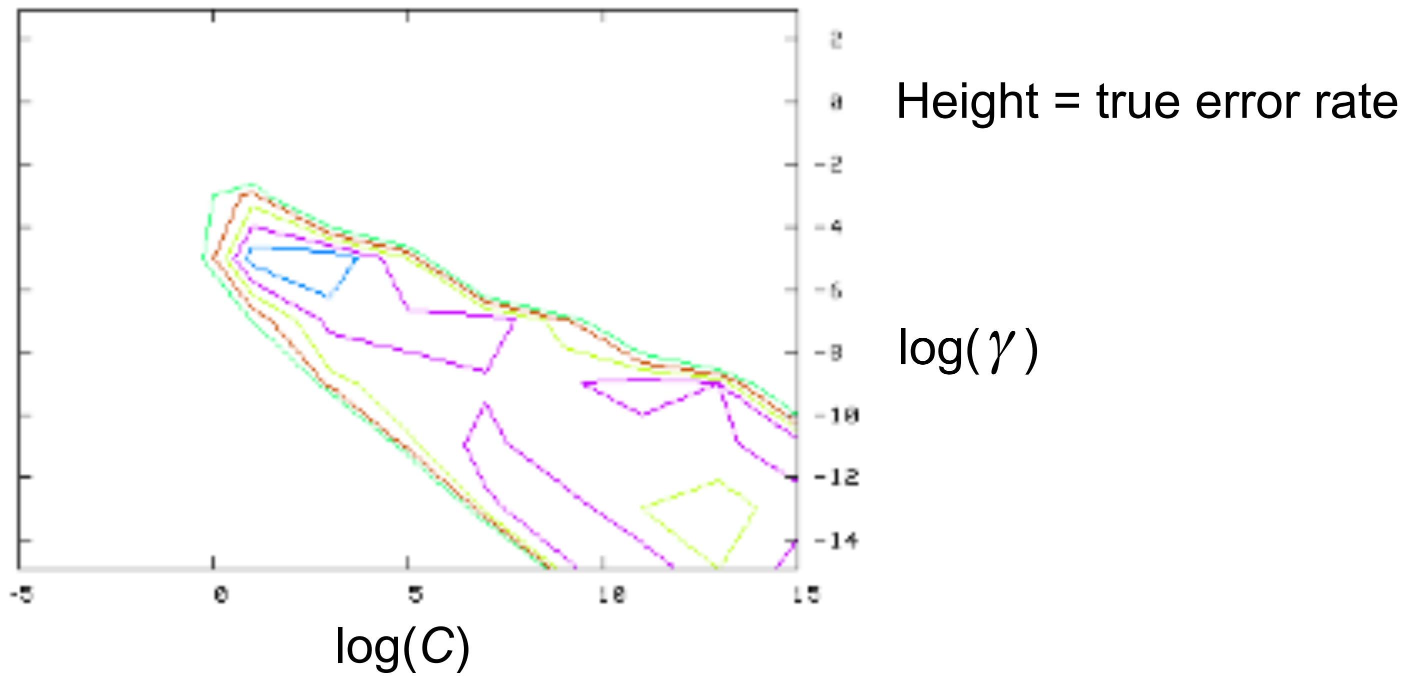
  (say, only C and γ)

Grid Search for Optimal (C, γ)

- In practice, the optimal choice of (C, γ) can be found by superimposing "grids" over a reasonable region in R^2

➔ Finding the one that minimizes the "true error rate" by CV.

- This can be done by exploring the following <u>contour plot</u>:



The Decision Function (SVM Classifier)

- A decision function is given by

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T \phi(\mathbf{x}) + b) = \text{sign}\left(\sum_{i=1}^{l} y_i \alpha_i \overbrace{\phi(\mathbf{x}_i)^T \phi(\mathbf{x})}^{<\phi(\mathbf{x}_i),\phi(\mathbf{x})>} + b\right)$$

- For a test vector $\mathbf{x}$, if $\sum_{i=1}^{l} y_i \alpha_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + b > 0$, we classify it to be in the class 1. Otherwise, we think it is in the second class.

- For the objects with corresponding $\alpha_i > 0$, we call them the "***support vectors***".

- It is noted that only those support vectors in data with affect the resulting SVM classifier.

The Multi-class Problems

- In real applications we often encounter data with more than two classes (e.g. the hand-written recognition system)

- For general K-class problems, there are two ways to deal with this problem:

  - One-against-one
  - One-against-all (one-against-the-rest)

One-against-one

- Suppose data contain K classes, we can construct $\binom{K}{2}$ decision functions for all possible 2-class problems.

- To classify each object, we ***vote for class of majority***.

  Example: $K = 4$

Resulting Class

$$\left.\begin{array}{c}1\\1\\1\\2\\3\\4\end{array}\right\} \dashrightarrow \text{Vote for Class 1}$$

One-against-the-rest

- Suppose data contain K classes, for each class c, we treat the rest of data (not belong to class c) as a second class. Thus, we can construct K decision functions for all possible 2-class problems.
- <u>Example</u>: $K = 4$

$y_i = 1$	$y_i = -1$	Decision function	
class 1	class 2,3,4	$f^1(\mathbf{x}) = (\mathbf{w}^1)^T\mathbf{x} + b^1$	值越大越確定是那邊
class 2	class 1,3,4	$f^2(\mathbf{x}) = (\mathbf{w}^2)^T\mathbf{x} + b^2$	
class 3	class 1,2,4	$f^3(\mathbf{x}) = (\mathbf{w}^3)^T\mathbf{x} + b^3$	
class 4	class 1,2,3	$f^4(\mathbf{x}) = (\mathbf{w}^4)^T\mathbf{x} + b^4$	

One-against-the-rest

- For any test data $\mathbf{x}$, if it is in the ith class, we would expect that

$$f^i(\mathbf{x}) \geq 0 \text{ and } f^j(\mathbf{x}) < 0, \text{ if } j \neq i.$$

- Therefore, $f^i(\mathbf{x})$ has the largest values among $f^1(\mathbf{x}), \cdots, f^4(\mathbf{x})$ and hence the decision rule is

$$\text{Predicted class } = \arg \max_{i=1,\cdots,4} f^i(\mathbf{x})$$

➔ For each object x, vote for the class so that it has the maximum decision function against the rest of classes.