Support Vector Machine

- Binary Classification
- Linear Classifiers
- Rosenblatt Perceptron
- Maximal Margin Classifier
- Support Vector Machines

- References:

- N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines. Cambridge: Cambridge University Press, 2000.
- V. Vapnik, Statistical Learning Theory. John Wiley & Sons, 1998.

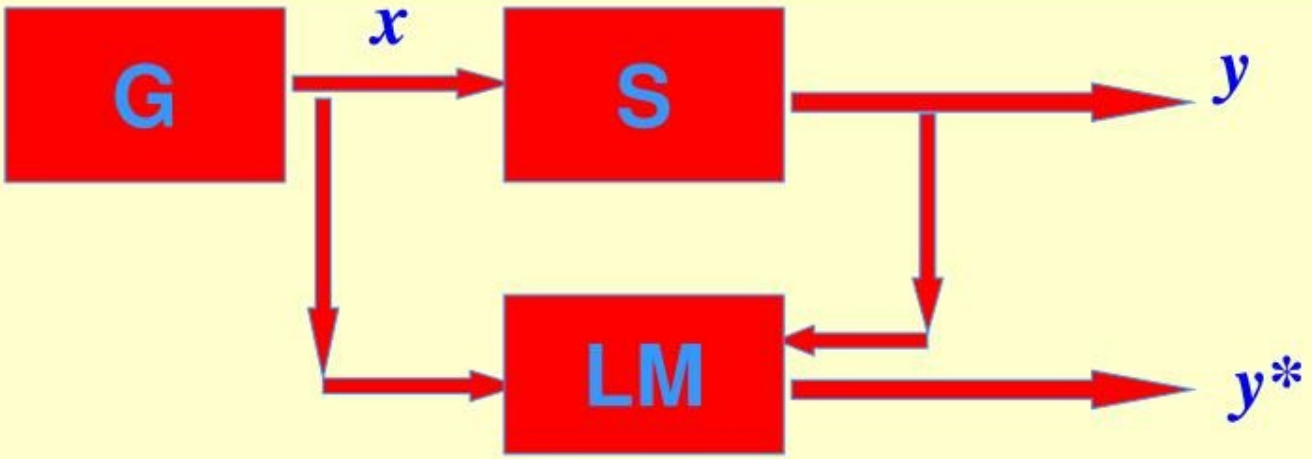
Learning and Inference

The inductive inference process:

- **Observe a phenomenon**
- **Construct a model of that phenomenon**
- **Make predictions using this model**

- **This is more or less the definition of natural sciences.**
- **The goal of Machine Learning is to automate this process.**
- **The goal of Learning Theory is to formalize it.**

The model of learning from examples



- A generator (G) of random vectors x
- A supervisor (S) who returns an output value y to every input vector x
- A learning machine (LM)

➢ During the learning process, the learning machine observes the pairs (x,y) (the training set). After training, the machine must on any given x return a value y*. The goal is to return a value y* that is close to the supervisor's response y.

Model for Binary Classification

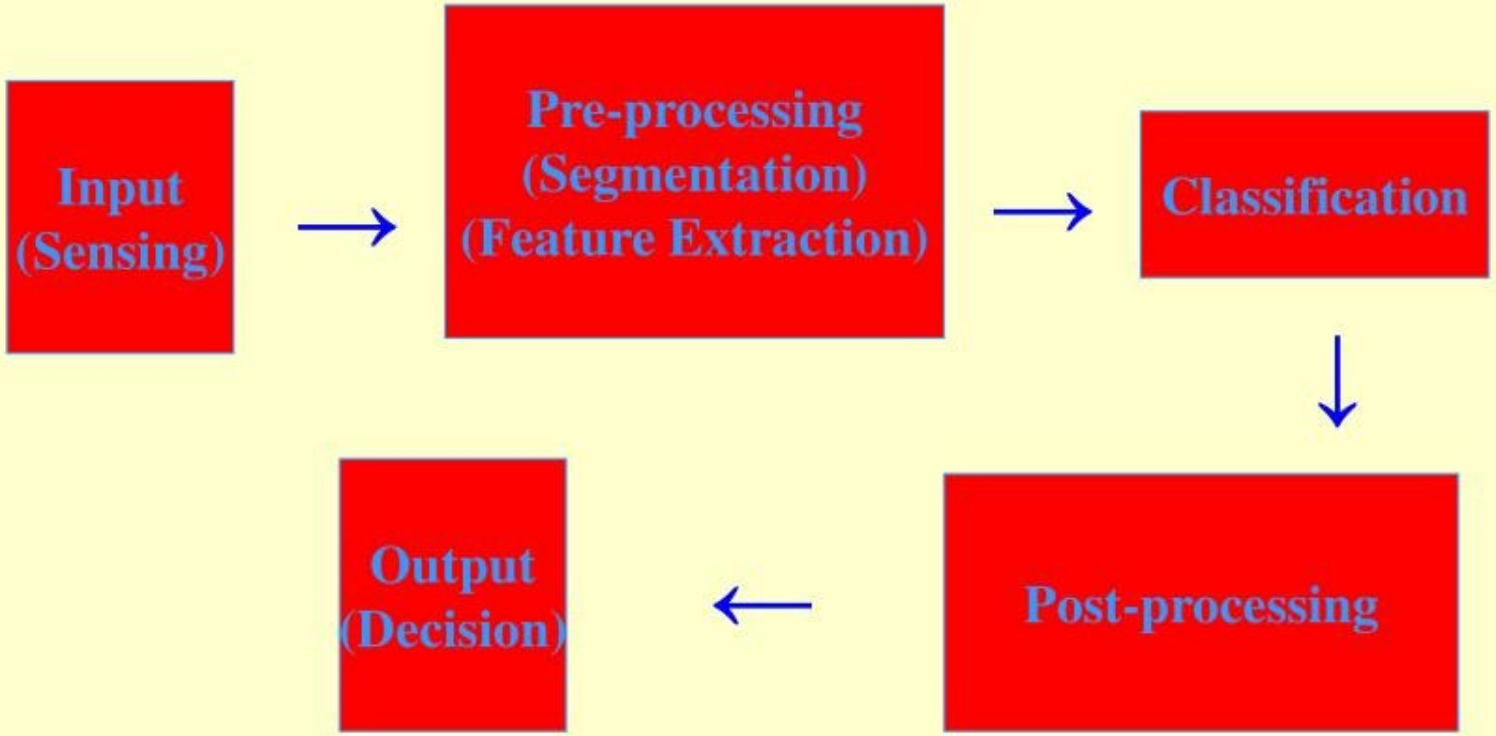
- **Assumption:** Input space X ; Output space Y = {−1, 1}
- **Assumption:** The pairs (x, y) ∈X × Y are distributed according to P (unknown)
- **Data:** Observe a sequence of n i.i.d. pairs:

  $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, sampled according to P
- **Goal:** Construct a function f : X→Y which predicts y from x

Example: Text Classification

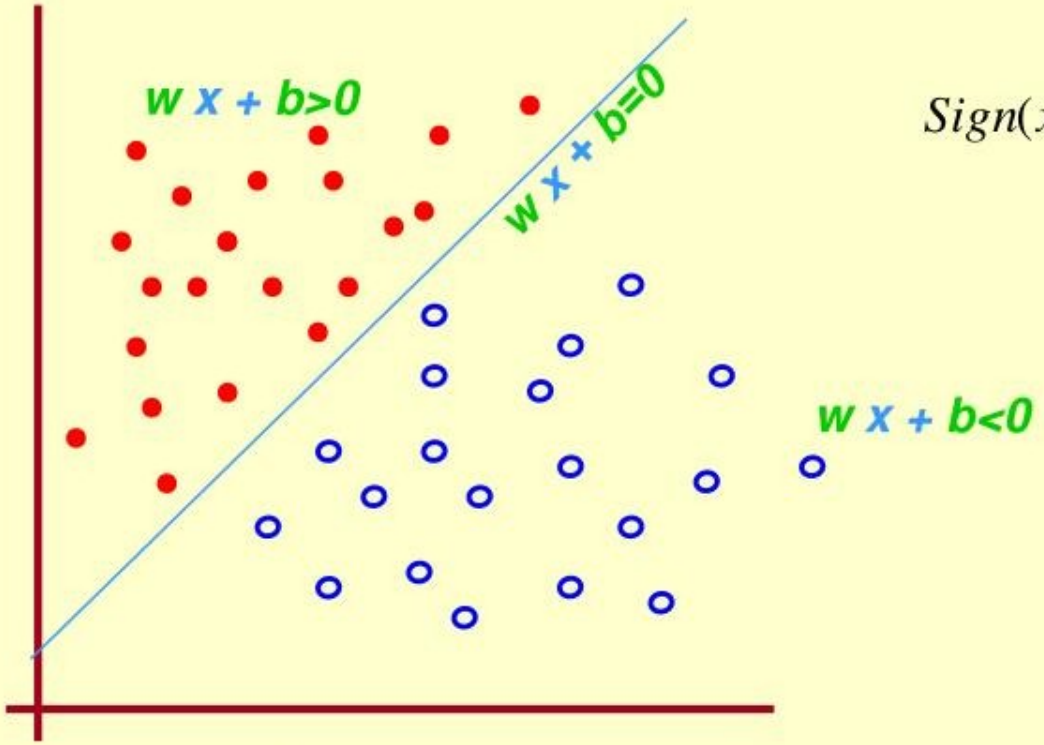
- **Goal:** to classify documents (news articles, emails, Web pages, etc.) into predefined categories
- **Examples**
  - To classify news articles into "business" and "sports"
  - To classify Web pages into personal home pages and others
  - To classify product reviews into positive reviews and negative reviews

Pattern Recognition Systems



Linear Classifiers

- denotes +1
- denotes - 1



$$Sign(x) = \begin{cases} 1 & x > 0 \\ -1 & x \le 0 \end{cases}$$

How would you classify this data?

$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$

Binary Classification Algorithms

- **Binary classifier:**

Find $f : X \subset R^d \rightarrow Y = \{+1, -1\}$

$x = (x_1, x_2, ..., x_n)$ is assigned to the positive class, if $f(x) > 0$

$x = (x_1, x_2, ..., x_n)$ is assigned to the negative class, if $f(x) \leq 0$

- **Linear binary classifier:**

$$f(x) = \langle w, x \rangle + b$$

$$= \sum_{i=1}^{n} w_i x_i + b \qquad w = (w_1, w_2, ..., w_n); \qquad x = (x_1, x_2, ..., x_n)$$

- **Decision function:** $g(x) = \text{sign}\ (f(x))$

Hyperplane

- A hyperplane $H_{w,b}$ in R^d, with normal vector w and bias b:

$$x_i, w_i, b \in R,\ x := \begin{bmatrix} x_1 & \cdots & x_d \end{bmatrix}^T,\ w := \begin{bmatrix} w_1 & \cdots & w_d \end{bmatrix}^T \in R^d.$$

- Define

$$f(w,x,b) := w_1 x_1 + w_2 x_2 + \ldots + w_d x_d + b = \langle w, x \rangle + b,\ x \in R^d.$$

$$\Rightarrow H_{w,b} = \left\{ x \in R^d : f_{w,b}(x) := \langle w, x \rangle + b = 0 \right\}$$

Rosenblatt Perceptron (Primal Form)

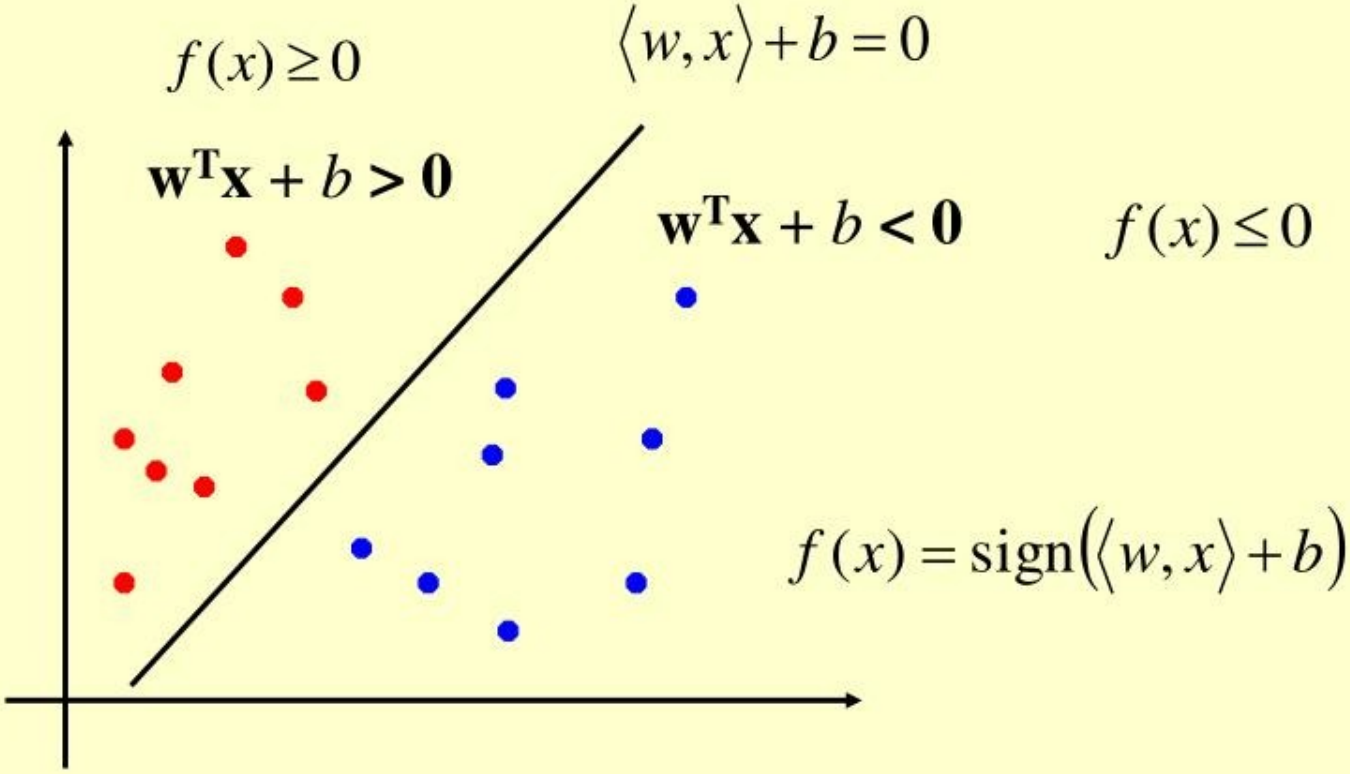
- **Hyperplane:**

A geometric interpretation is that the input space X is split into two parts by the hyperplane $\langle w, x \rangle + b = 0$

- **Linearly separable:**

For training set $S = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ there exists a hyperplane $\langle w, x \rangle + b = 0$ that correctly classifies the training set

Binary Classification Algorithms



Binary Classification Algorithms

- **Margin:**

The functional margin of an example (x_i, y_i) with respect to a hyperplane $\langle w, x \rangle + b = 0$ is the quantity $\gamma_i = y_i\left(\langle w, x_i \rangle + b\right)$

Note that $\gamma_i > 0$ implies correct classification of (x_i, y_i)

- **Margin of a training set** $S = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$

$$\gamma_S := \max_{w,b} \min_{i=1}^{n} \left[y_i \cdot \left(\left\langle \|w\|^{-1} w, x_i \right\rangle + \|w\|^{-1} b \right) \right]$$

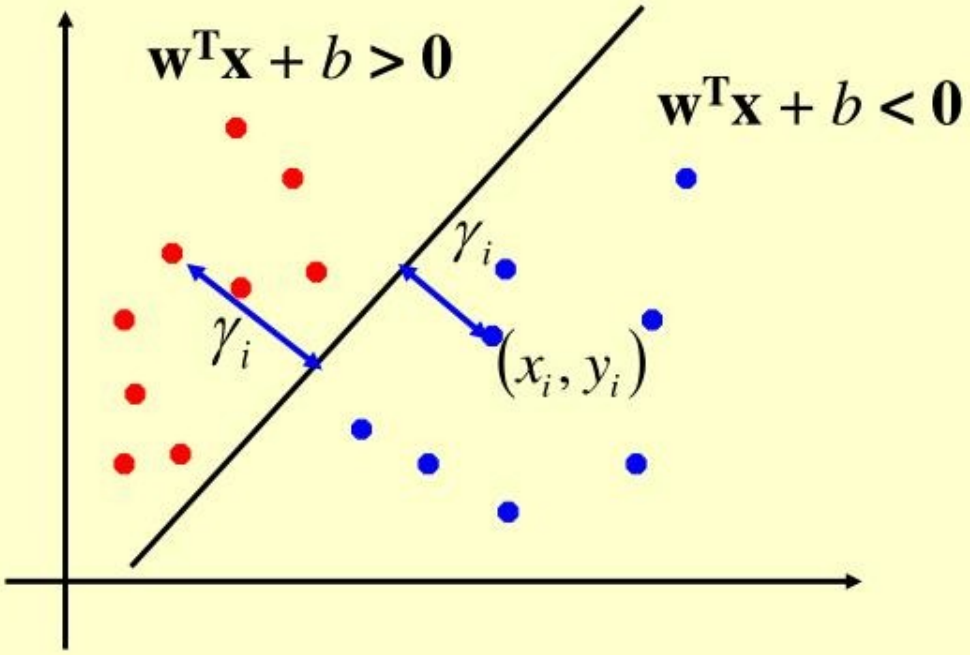
- **Maximal margin hyperplane:**

The margin of a training set S is the maxmum geometric margin over all hyperplane.

A hyperplane realizing this margin is called a maximal margin hyperplane.

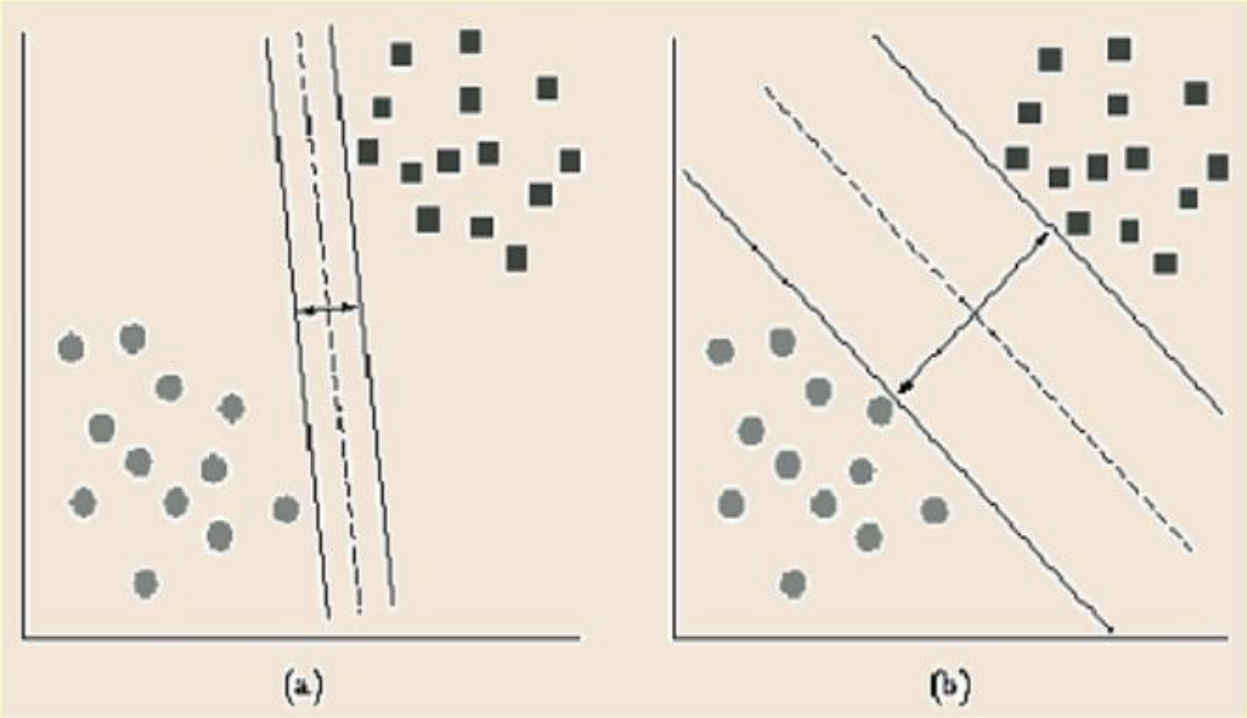
Binary Classification Algorithms

$$\langle w, x \rangle + b = 0; \qquad \langle w, x \rangle = w^T x$$



Binary Classification Algorithms

- **Maximal margin hyperplane**



Rosenblatt Perceptron (Primal Form)

Given a linearly separable training set $S = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$

The perceptron updates the weight vector and bias through the following recurrent procedure:

Step 1. Set $w_0 = 0; \quad b_0 = 0$

Step 2.
$$w_{k+1} = \begin{cases} w_k & \text{if} \quad y_i(\langle w_k, x_i \rangle + b_k) \geq 0 \\ w_k + \eta y_i x_i & \text{if} \quad y_i(\langle w_k, x_i \rangle + b_k) \leq 0 \end{cases}$$

$$b_{k+1} = \begin{cases} b_k & \text{if} \quad y_i(\langle w_k, x_i \rangle + b_k) \geq 0 \\ b_k + \eta y_i R^2 & \text{if} \quad y_i(\langle w_k, x_i \rangle + b_k) \leq 0 \end{cases} \quad R^2 = \max_{1 \leq i \leq n} \| x_i \|$$

Step 3. Return $w_k, b_k = 0, k$: the number of mistakes

Rosenblatt Perceptron (Primal Form)

- Data: the training set $S := \{(x_i, y_i)\}_{i=1}^{n} \subseteq X \times Y$ and a learning rate $\eta > 0$.
- Goal: a hyperplane (w, b) that correctly classifies the training set.

➢ Step 1: $w_0 \leftarrow 0; b_0 \leftarrow 0; k \leftarrow 0;$

➢ Step 2: Choose $R = \max_{i=1}^{n} \|x_i\|;$

Rosenblatt Perceptron (Primal Form)

- Step 3: repeat

  for i=1 to n

  if $y_i \cdot [\langle w_k, x_i \rangle + b_k] \leq 0$, then

  end if $w_{k+1} \leftarrow w_k + \eta y_i x_i; b_{k+1} \leftarrow b_k + \eta y_i R^2; k \leftarrow k+1;$

  until no misclassification within the *for* loop

  return k, (w_k, b_k) where k is the number of mistakes

- **Note**: In case where the training set is not linearly separable, the algorithm will not converge.

Novikoff Theorem

- **Novikoff Theorem**:

  Suppose S is a nontrivial training set and there exist a vector $w^* \in R^d, \|w^*\| = 1$, a number $b^* \in R$, and a positive number $\gamma > 0$ such that

$$y_i \cdot \left[\langle w^*, x_i \rangle + b^*\right] \geq \gamma > 0 \qquad \text{for all} \qquad i$$

  Then the number of mistakes made by the on-line perceptron algorithm on the training set S is at most $(2R/\gamma)^2$.

Rosenblatt Perceptron (Dual Form)

In the primal form of Rosenblatt algorithm starting from $w_0 = 0$, the final weight is of the form $w = \sum_{i=1}^{n} \beta_i \eta y_i x_i = \sum_{i=1}^{n} \alpha_i y_i x_i \quad \alpha_i := \beta_i \eta \geq 0, \quad \text{for all } i$

and β_i, is the number of mistakes when using (x_i, y_i) as training example.

Then we have

$$f(x) := \langle w, x \rangle + b = \left\langle \sum_{j=1}^{n} \eta \beta_j y_j x_j, x \right\rangle + b = \sum_{j=1}^{n} \eta \beta_j y_j \langle x_j, x \rangle + b$$

$$y_i \cdot f(x) = y_i \cdot [\langle w, x \rangle + b] = y_i \cdot \left[\sum_{j=1}^{n} \eta \beta_j y_j \langle x_j, x \rangle + b \right]$$

This means that the decision rule can be evaluated using just inner products between the test point x and the training points x_i's, i.e., $\langle x, x_i \rangle$.

Rosenblatt Perceptron (Dual Form)

- Data: the training set $S := \{(x_i, y_i)\}_{i=1}^n \subseteq X \times Y$ and a learning rate $\eta > 0$.
- Goal: a hyperplane (β, b) that correctly classifies the training set.

➢ Step 1: $\beta \leftarrow 0; b \leftarrow 0;$

➢ Step 2: Choose $R = \max_{i=1}^{n} \|x_i\|;$

➢ Step 3: repeat

for i=1 to n , if $y_i \cdot \left[\sum_{j=1}^{n} \eta \beta_j y_j \langle x_j, x_i \rangle + b \right] \le 0$

then $\beta_i \leftarrow \beta_i + 1; b \leftarrow b + \eta y_i R^2$

end if

end for

until no misclassification within the *for* loop

return (β, b) to define the separating hyperplane.

Rosenblatt Perceptron (Dual Form)

- Usually we choose $\eta = 1$. In which case, we have $a_i = b_i$ for all i.
- The training data only enter the algorithm through the entries of the Grammian matrix

$$G := \left[\left\langle x_i, x_j \right\rangle\right] \in R^{n \times n}.$$

- In the preceding algorithm, the integer $\|\beta\|_1 := \beta_1 + \beta_2 + \ldots + \beta_n$ is equal to the number of mistakes. By Novikoff Theorem, we have

$$\|\beta\|_1 \leq (2R/\gamma)^2.$$

Maximal Margin Classifier

- **Functional margin** of an example (x_i, y_i), w.r.t. the hyperplane $f(x) = \langle w, x \rangle + b = 0$

$$\gamma_i := y_i \cdot \left[\langle w, x_i \rangle + b \right] = y_i \cdot f(x_i) \qquad f(x_i) = \langle w, x_i \rangle + b$$

- **Geometric margin** of an example (x_i, y_i), w.r.t. the hyperplane $f(x) = \langle w, x \rangle + b = 0$

$$\eta_i := y_i \cdot \left[\left\langle \|w\|^{-1} w, x_i \right\rangle + \|w\|^{-1} b \right] = y_i \cdot g(x_i)$$

$$g(x_i) = \left\langle \|w\|^{-1} w, x_i \right\rangle + \|w\|^{-1} b$$

Maximal Margin Classifier

- **Functional margin** of a hyperplane (w,b) w.r.t. the training set S:

$$\gamma_S(w,b) := \min_{1 \le i \le n} \gamma_i$$

- **Geometric margin** of a hyperplane (w,b) w.r.t. the training set S:

$$\eta_S(w,b) := \min_{1 \le i \le n} \eta_i$$

How to determine the hyperplane (w,b)

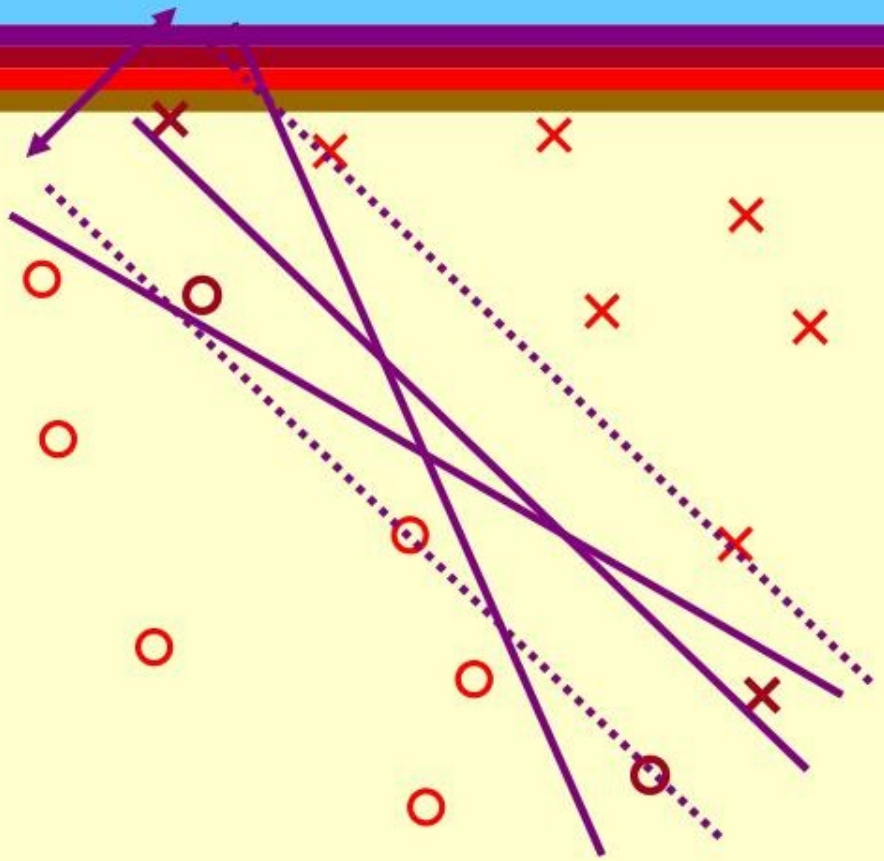
Maximal Margin Classifier

一般而言，一个点距离超平面的远近可以表示为分类预测的确信或准确程度。在超平面w*x+b=0确定的情况下，|w*x+b|能够相对的表示点x到距离超平面的远近，而w*x+b的符号与类标记y的符号是否一致表示分类是否正确，所以，可以用量y*(w*x+b)的正负性来判定或表示分类的正确性和确信度，于此，我们便引出了函数间隔functional margin的概念。

Maximal Margin Classifier

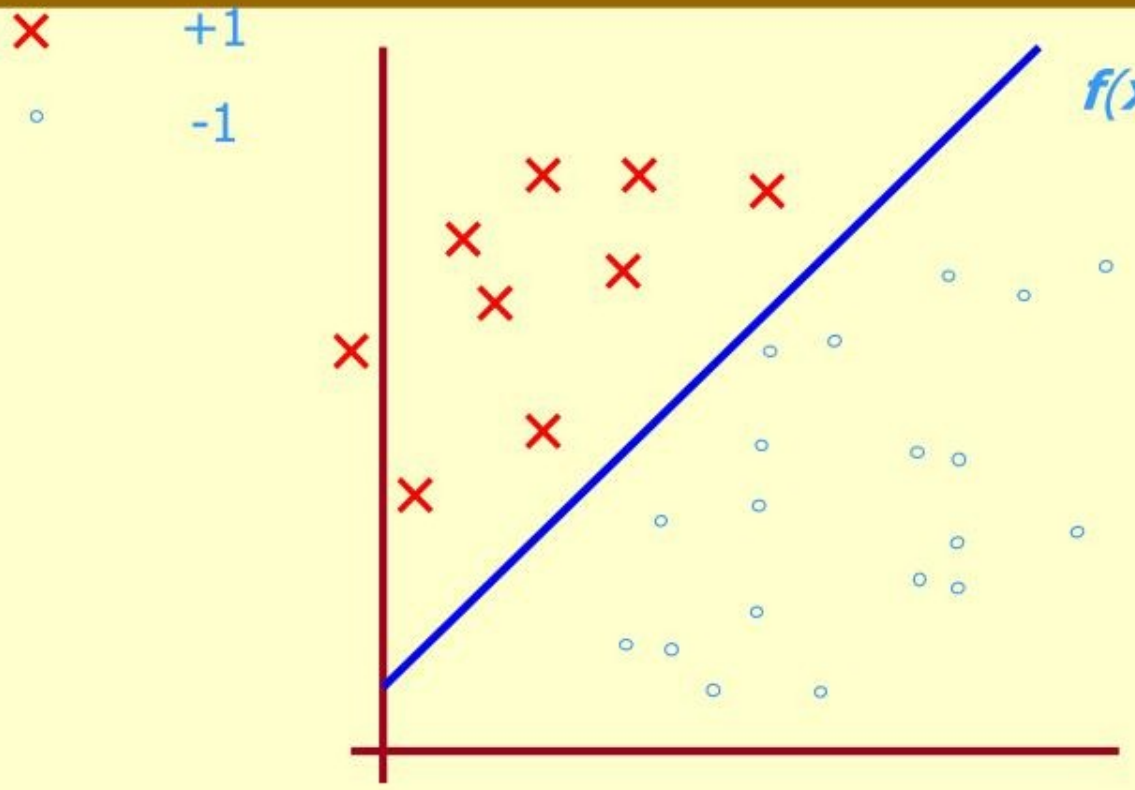
不过这里我们有两个 margin 可以选，不过 functional margin 明显是不太适合用来最大化的一个量，因为在 hyper plane 固定以后，我们可以等比例地缩放 w 的长度和 b 的值，这样可以使得 $f(x)$ = w*x+b的值任意大，亦即 functional margin $\gamma\hat{}$ 可以在 hyper plane 保持不变的情况下被取得任意大，而 **geometrical margin**则没有这个问题，因为除上了 $//w//$ 这个分母，所以缩放 w 和 b 的时候 $\gamma\tilde{}$ 的值是不会改变的，它只随着 hyperplane 的变动而变动，因此，这是更加合适的一个 margin 。

Maximal Margin Classifier

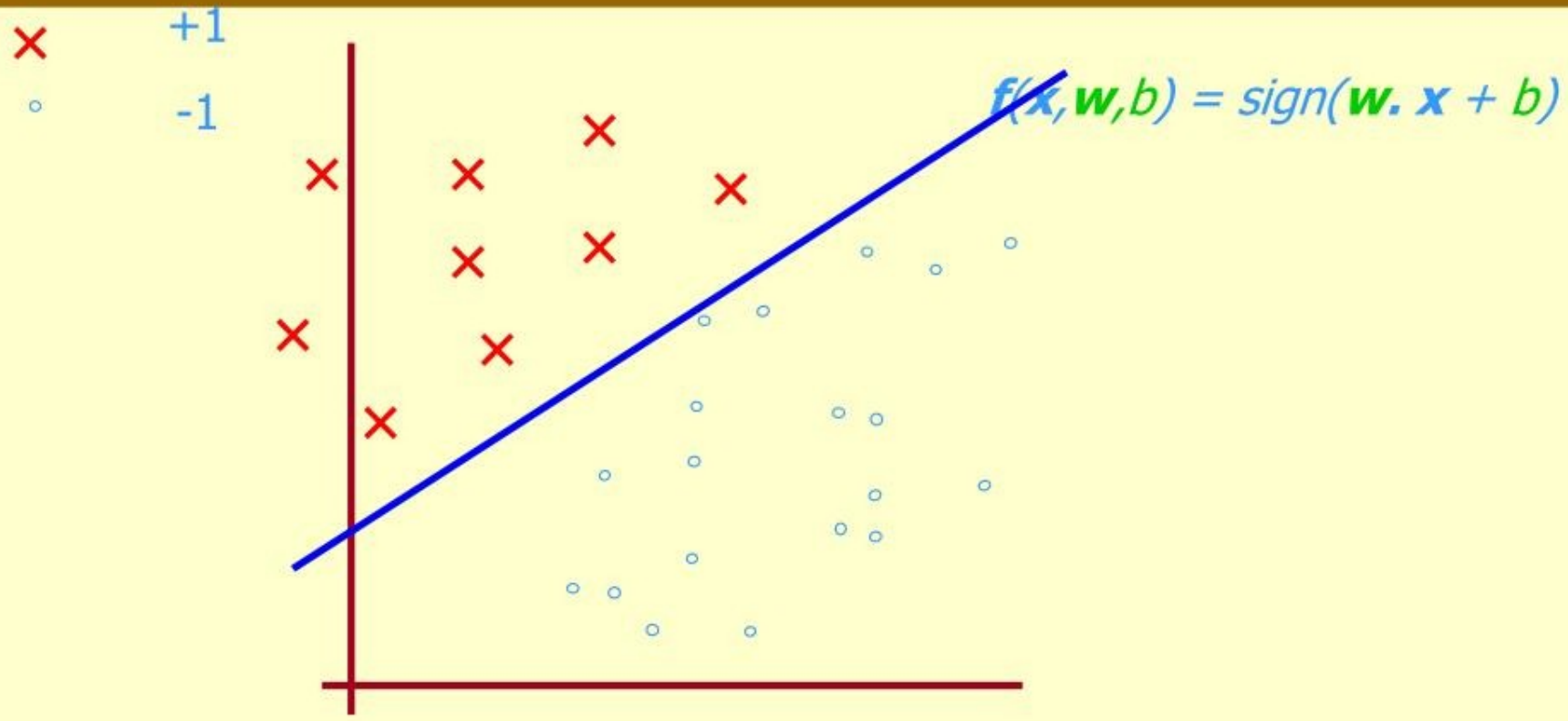


- vectors $\mathbf{X}_i$
- labels $y_i = \pm 1$

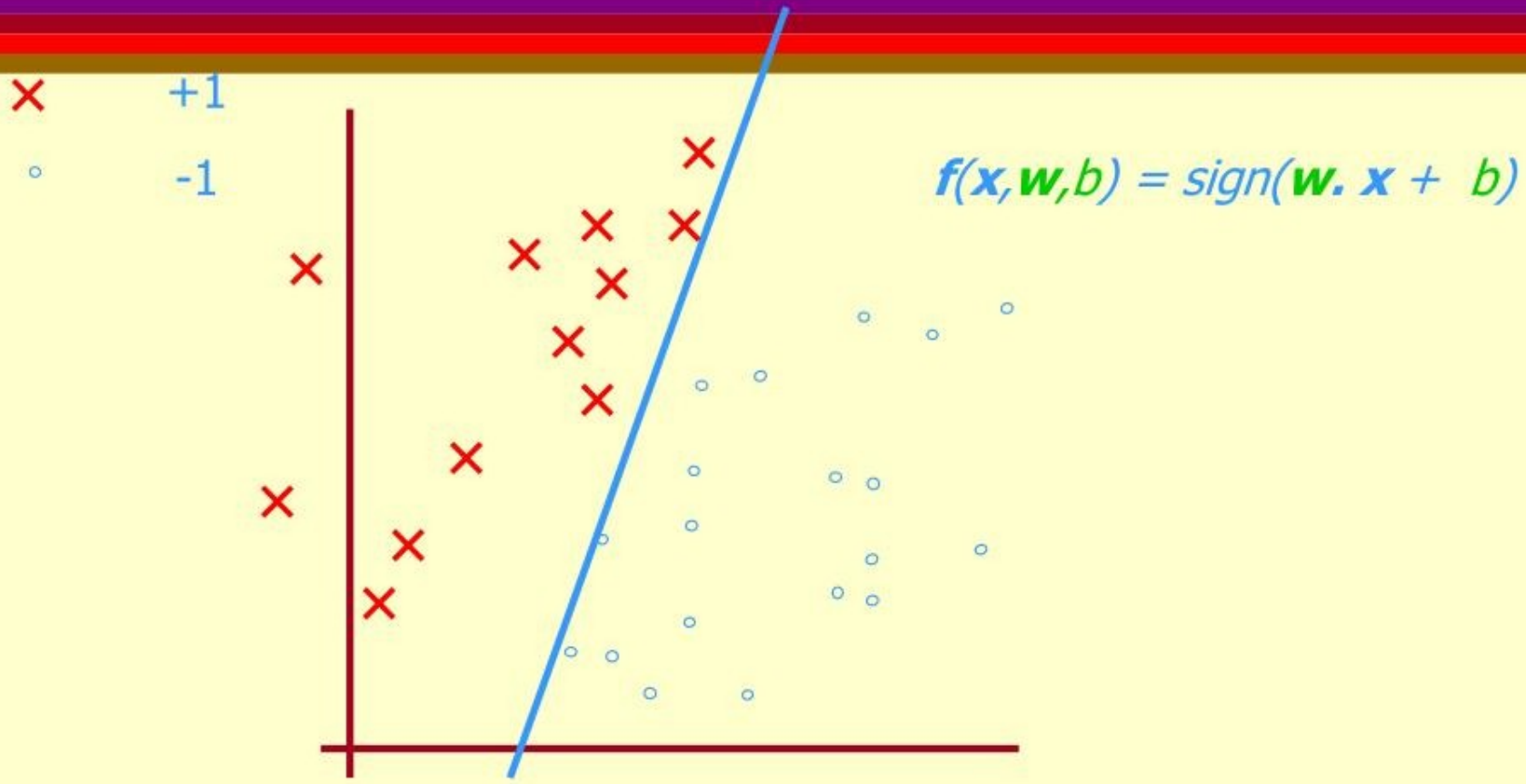
Maximal Margin Classifier



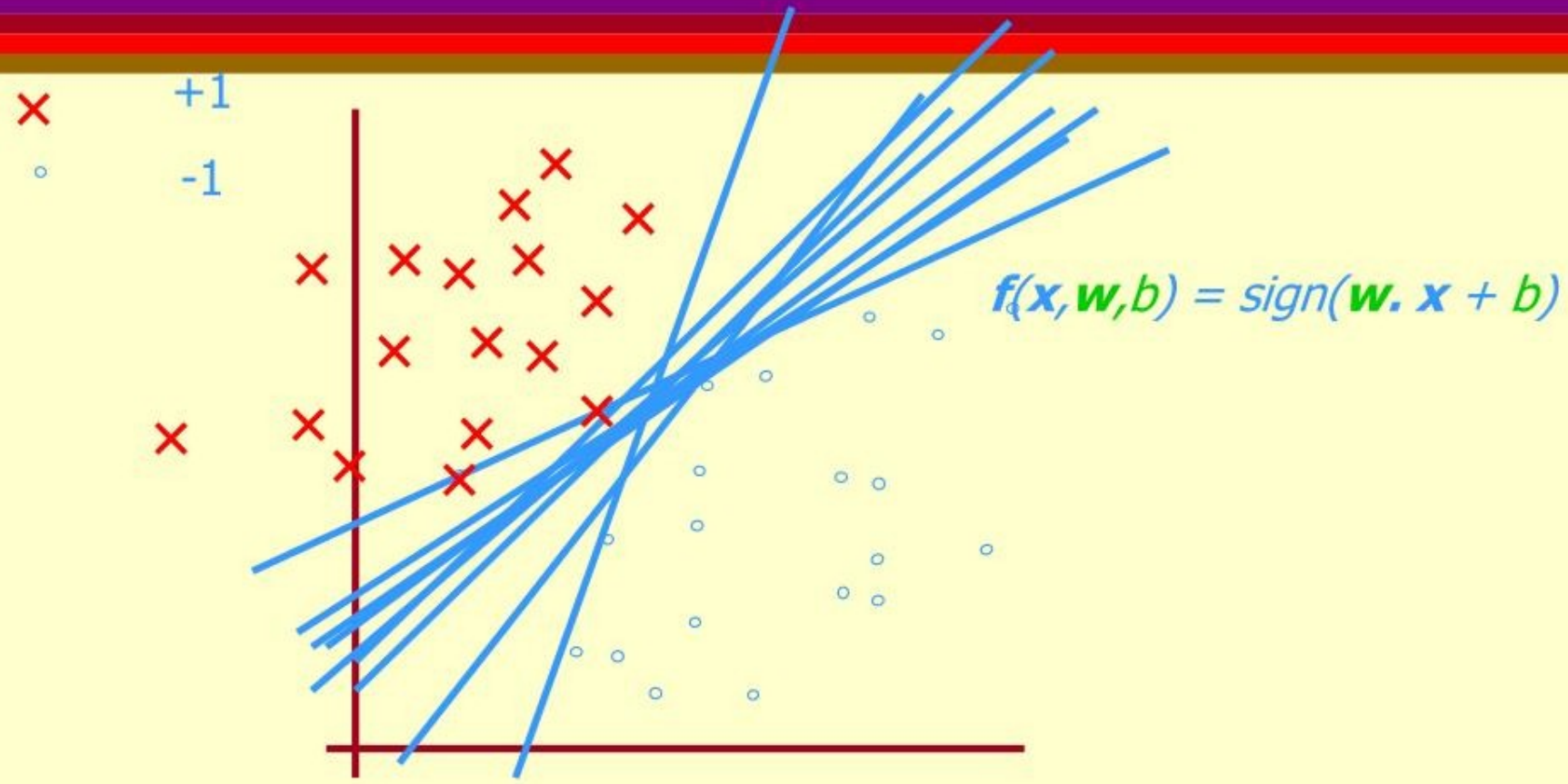
Maximal Margin Classifier



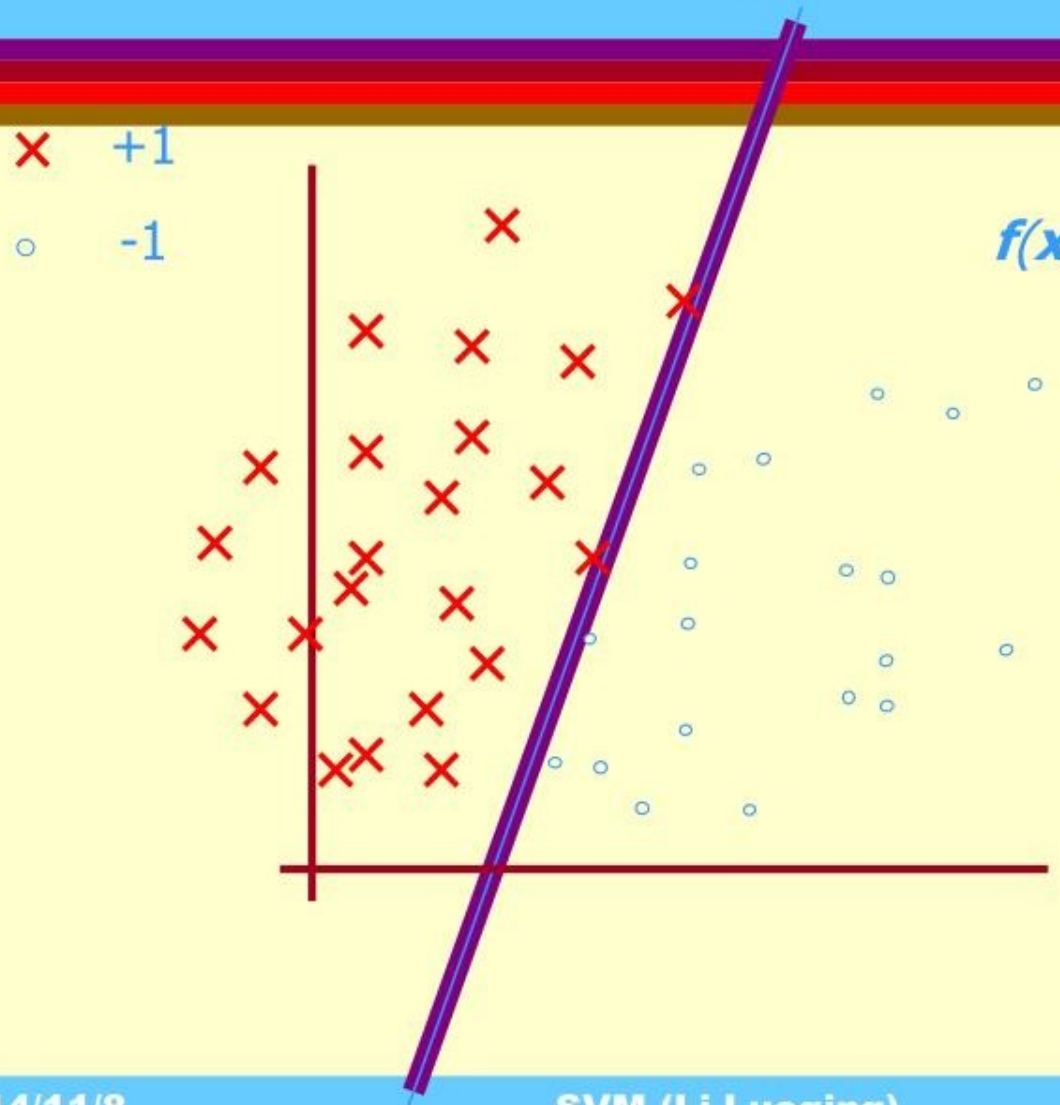
Maximal Margin Classifier



Maximal Margin Classifier



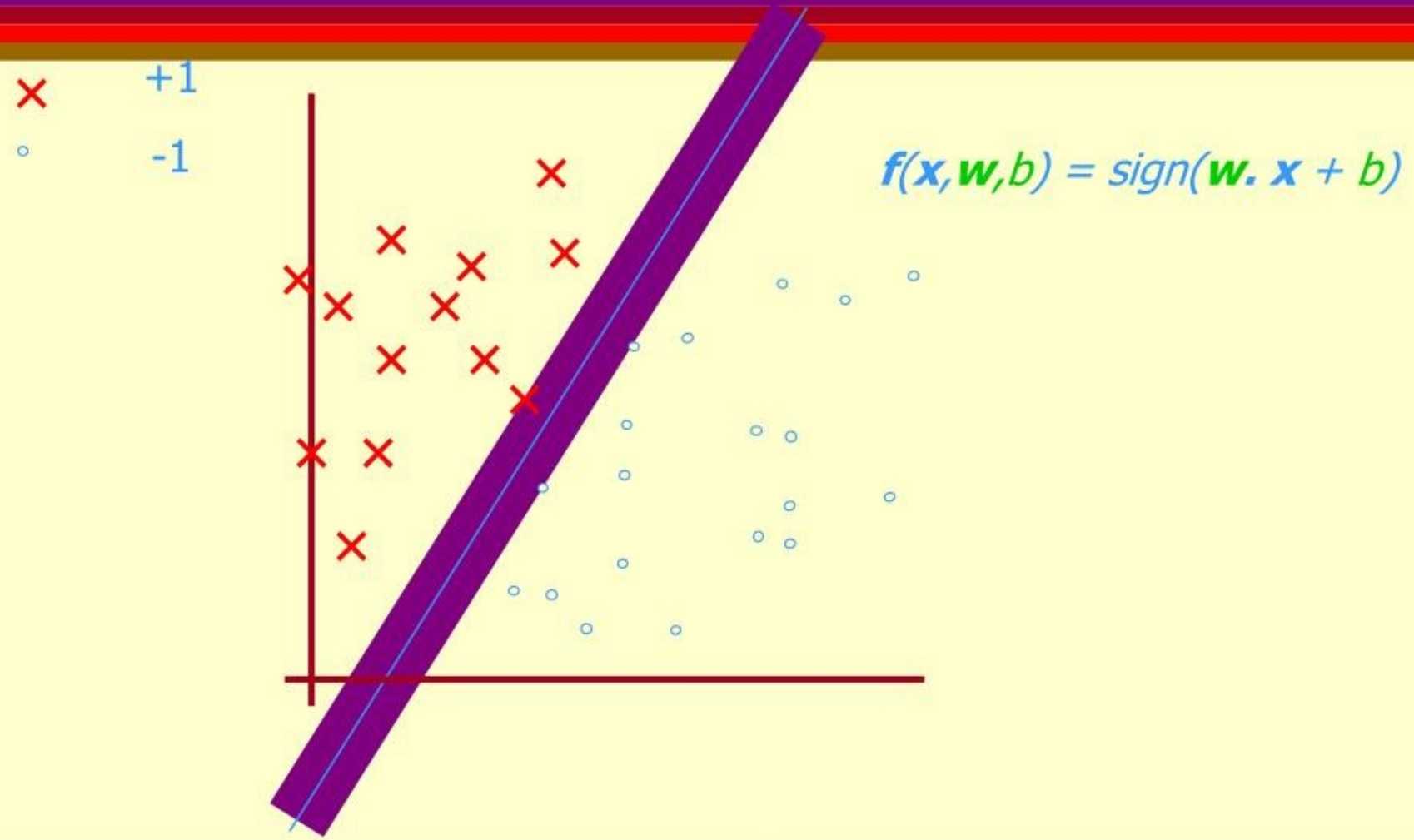
Maximal Margin Classifier



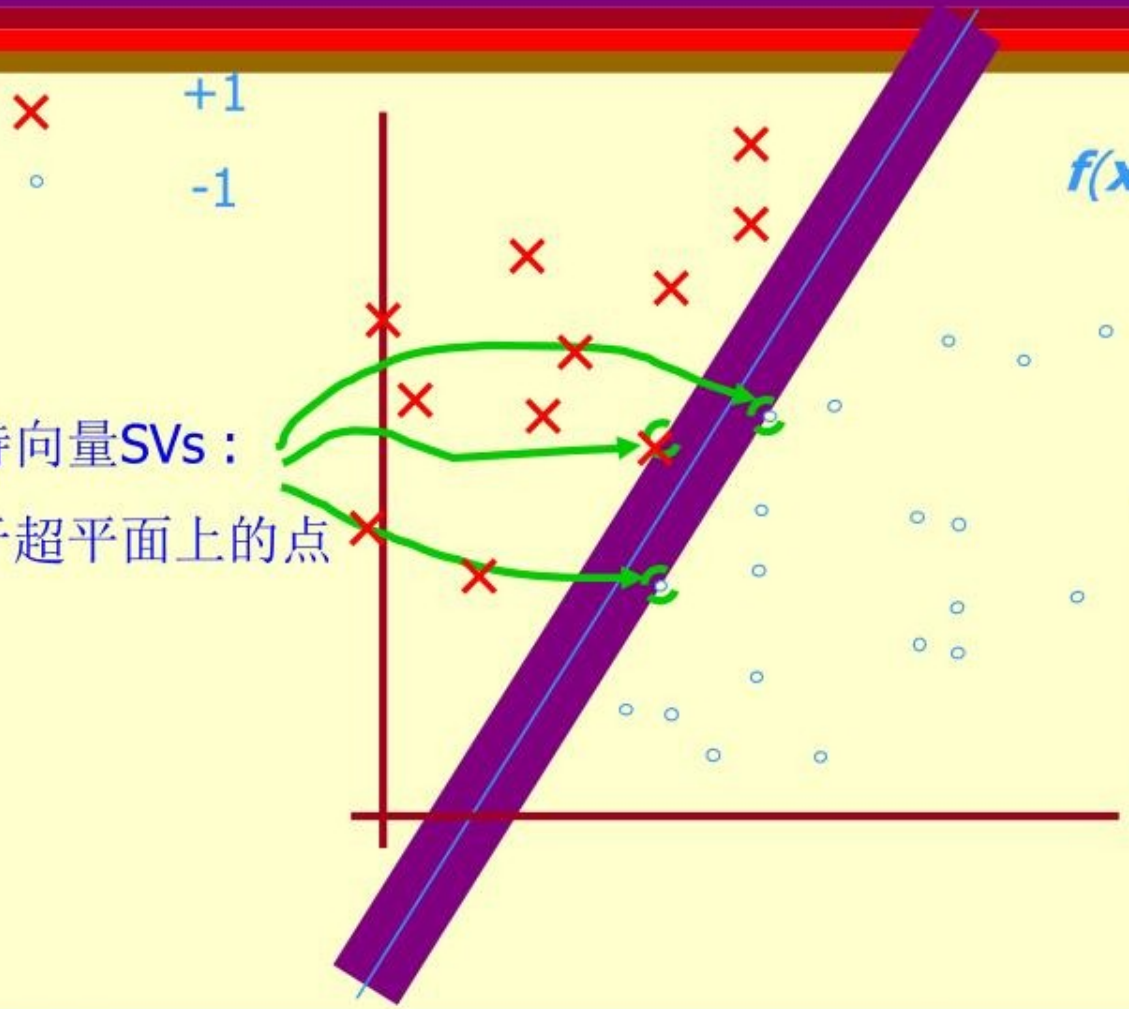
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\mathbf{x} + b)$

线性分类器的间隔（margin）：到超平面最近的样本与此超平面之间的距离。

Maximal Margin Classifier



Maximal Margin Classifier



$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\mathbf{x} + b)$

支持向量SVs：

位于超平面上的点

优点

- 最大扰动
- 预测能力强

Maximal Margin Classifier

- Let $X \subseteq R^n$ and $Y := \{1, -1\}$.
- Training examples: $S := \{(x_i, y_i)\}_{i=1}^n \subseteq X \times Y$
- Define $I_S^+ := \{i : y_i = 1\}$, $I_S^- := \{j : y_j = -1\}$.
- **Fact**: The following statements are equivalent:
  - (a) The margin of S is γ_S.

Maximal Margin Classifier

- (b) There exist $w^* \in \Re^n, \|w^*\| = 1, b^* \in \Re$, and $\gamma_s > 0$ such that

$$y \cdot \left[\left\langle w^*, x_i^* \right\rangle + b^*\right] \geq \gamma_S > 0 \text{ for all } i$$

$$\left\langle w^*, x_i^* \right\rangle + b^* = \gamma_S \text{ for some } i \in I_S^+$$

$$\left\langle w^*, x_j^* \right\rangle + b^* = -\gamma_S \text{ for some } j \in I_S^-$$

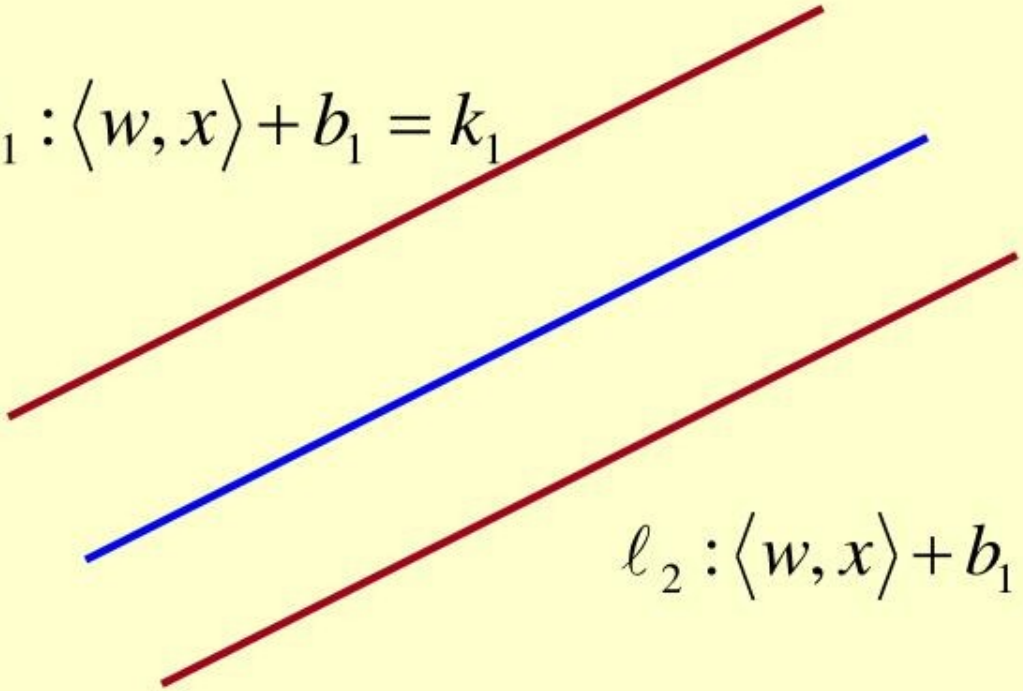
- (c) There exist $w_0 \in R^n$ and $b_0 \in R$, with $\gamma_S := \|w_0\|^{-1} > 0$, such that

$$y_i \cdot \left[\left\langle w_0, x_i \right\rangle + b_0\right] \geq 1 > 0 \text{ for all } i$$

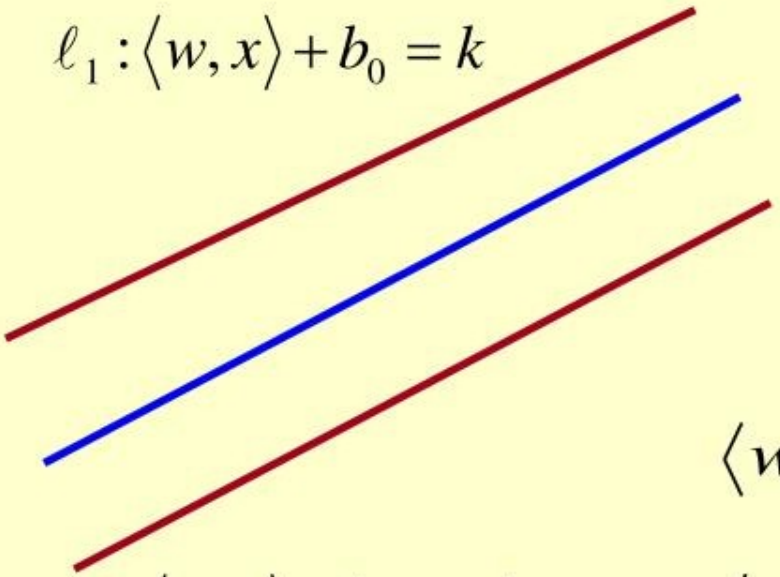
$$\left\langle w^*, x_i^* \right\rangle + b^* = \gamma_S \text{ for some } i \in I_S^+$$

$$\left\langle w_0, x_j^* \right\rangle + b_0 = -1 \text{ for some } j \in I_S^-$$

Maximal Margin Classifier



Maximal Margin Classifier



$\ell_1 : \langle w, x \rangle + b_0 = k$

$\ell_2 : \langle w, x \rangle + b_0 = -k$

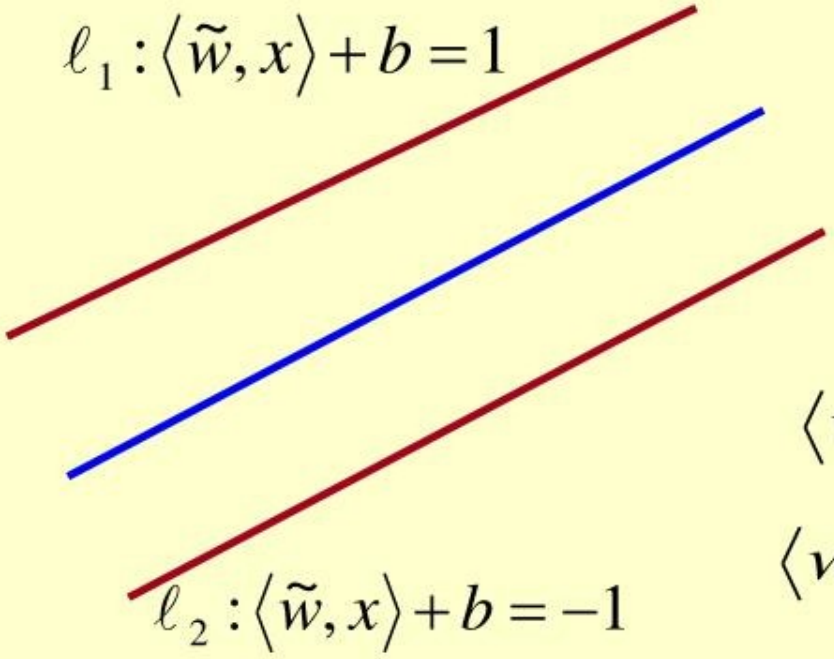
Set $k = \frac{1}{2}(k_1 - k_2), \quad b_0 = b_1 - \frac{1}{2}(k_1 + k_2)$

We obtain

$$\langle w, x \rangle + b_1 = k_1 \Leftrightarrow \langle w, x \rangle + b_0 = k$$

$$\langle w, x \rangle + b_1 = k_2 \Leftrightarrow \langle w, x \rangle + b_0 = -k$$

Maximal Margin Classifier



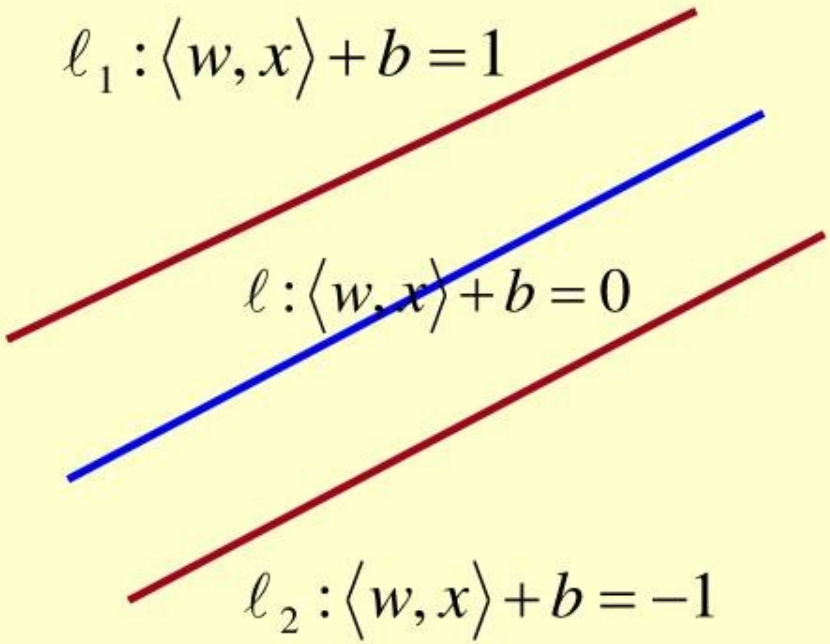
Set $\tilde{w} = \frac{1}{k} w, \quad b = \frac{1}{k} b_0$

We obtain

$$\langle w, x \rangle + b_0 = k \Leftrightarrow \langle \tilde{w}, x \rangle + b = 1$$

$$\langle w, x \rangle + b_0 = k \Leftrightarrow \langle \tilde{w}, x \rangle + b = -1$$

Maximal Margin Classifier



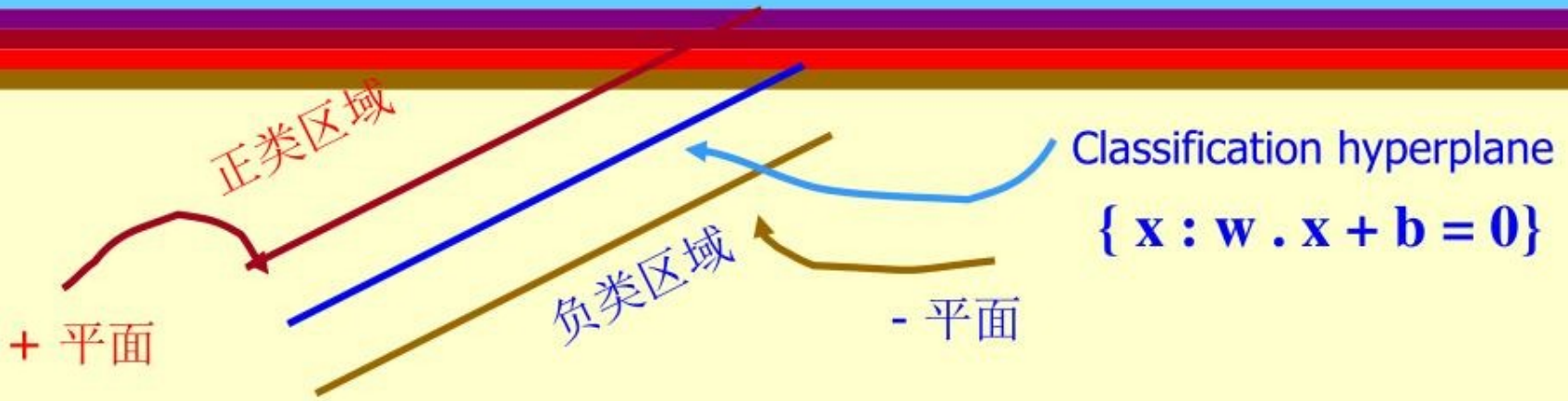
It is suitable to choose the hyperplane classifier as

$$\ell : \langle w, x \rangle + b = 0$$

$$\ell_1 : \langle w, x \rangle + b = 1$$

$$\ell_2 : \langle w, x \rangle + b = -1$$

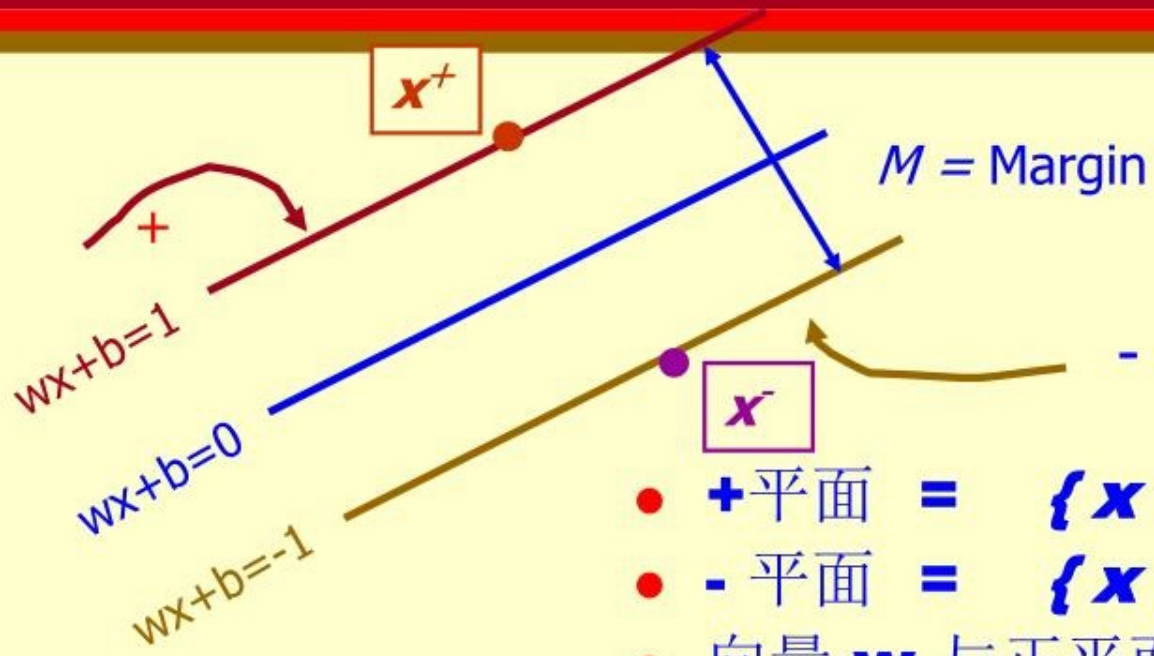
Computing the Margin



- + 平面 $\{x : w . x + b = +1\}$
- -- 平面 $\{x : w . x + b = -1\}$

- 分类： + 若 $w . x + b >= +1$
  -- 若 $w . x + b < -1$

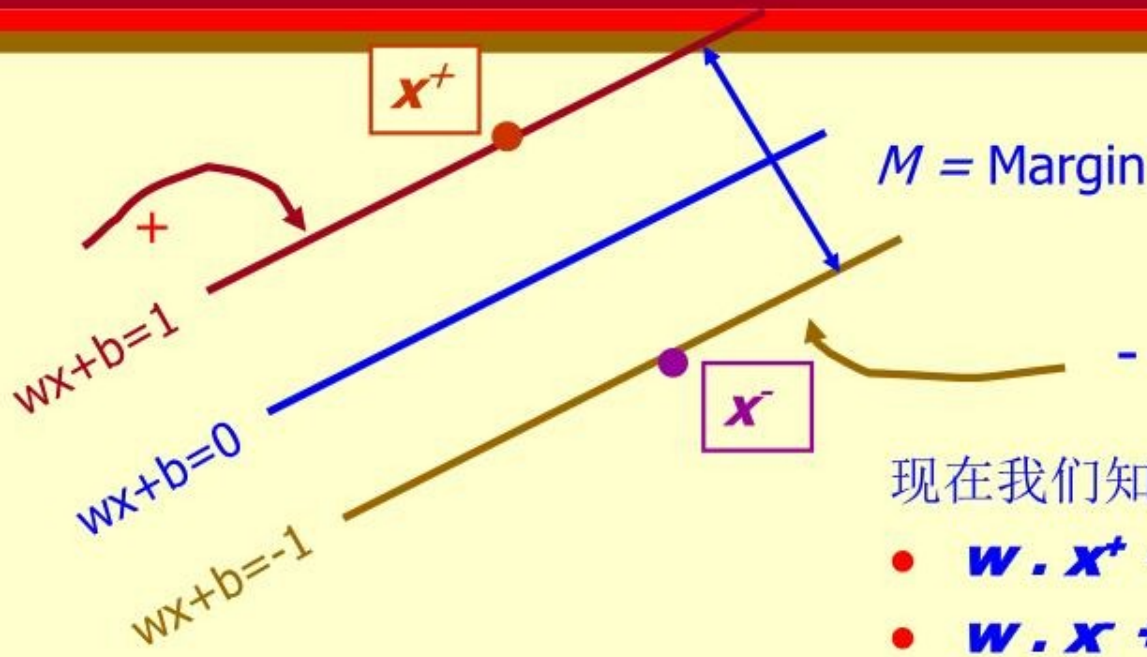
Computing the Margin



利用 ***w***与*b* 计算间隔

- **+**平面 = ***{x : w . x + b = +1}***
- **-** 平面 = ***{x : w . x + b = -1}***
- 向量 **w** 与正平面垂直
- 设 ***x-*** 是**--**平面上任意一点
- 设 ***x+***是**+**平面上距离 ***x-***最近的点

Computing the Margin



利用 ***w***与*b* 计算间隔

现在我们知道:

- $\boldsymbol{w} \cdot \boldsymbol{x}^{+} + b = +1$
- $\boldsymbol{w} \cdot \boldsymbol{x}^{-} + b = -1$
- $\boldsymbol{x}^{+} = \boldsymbol{x}^{-} + \lambda \boldsymbol{w}$
- $|\boldsymbol{x}^{+} - \boldsymbol{x}^{-}| = M$

于是很容易由***w*** 和***b*** 得到 ***M***

Computing the Margin

$$\left.\begin{array}{l}\langle w, x^+\rangle + b = +1 \\ \langle w, x^-\rangle + b = -1 \\ x^+ = x^- + \lambda w\end{array}\right\} \Rightarrow \left.\begin{array}{c}\langle w, (x^- + \lambda w\rangle + b = +1 \\ \langle w, x^-\rangle + b + \lambda\langle w, w\rangle = +1\end{array}\right\}$$

$$\Rightarrow -1 + \lambda\langle w, w\rangle = +1 \Rightarrow \lambda = \frac{2}{\langle w, w\rangle} = \frac{2}{\| w \|^2}$$

$$\text{Margin} \quad M = \mid x^+ - x^- \mid = \lambda \| \mathbf{w} \| = \frac{2}{\| w \|^2} \| w \| = \frac{2}{\| w \|}$$

Maximal Margin Classifier

- **Observation**:

maximization of the margin of S is equivalent to minimize the Euclidean norm of the weight vector w

- **Primal problem**: (P0)

$$\text{minimize} \quad \frac{1}{2}\|w\|^2$$

$$\text{subject to} \quad y_i \cdot \left[\langle w, x_i \rangle + b\right] \geq 1 \quad \text{for all} \quad i$$

Maximal Margin Classifier

- The cost functional $\frac{1}{2}||w||^2$ is continuous, convex, and quadratic in w. Furthermore, the constraints are affine in w and b.
- Optimal discriminant function: $f^*(x) = \langle w^*, x \rangle + b^*$
- Margin: $\|w^*\|^{-1}$

Maximal Margin Classifier

- **Dual problem**: Lagrangian:

$$L(w,b,\alpha) := 2^{-1}\langle w,w\rangle + \sum_{i=1}^{n}\alpha_i\left(1 - y_i\langle w,x_i\rangle - y_i b\right)$$

Lagrange multiplier vector: $\alpha := [\alpha_1 \quad \cdots \quad \alpha_n]^T \in R^n$

- Derivation:

$$0 = \frac{\partial L}{\partial w} = w - \sum_{i=1}^{n}\alpha_i y_i x_i, \quad 0 = \frac{\partial L}{\partial b} = -\sum_{i=1}^{n}\alpha_i y_i$$

$$\Rightarrow w = \sum_{i=1}^{n}\alpha_i y_i x_i, \qquad \sum_{i=1}^{n}\alpha_i y_i = 0$$

$$\Rightarrow L(w,b,\alpha) = \sum_{i=1}^{n}\alpha_i - 2^{-1}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\langle x_i, x_j\rangle$$

Maximal Margin Classifier

- **Dual problem(D0):**

$$\text{maximize} \quad \sum_{i=1}^{n}\alpha_i - 2^{-1}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \langle x_i, x_j \rangle$$

$$\text{subject to} \quad \sum_{i=1}^{n}\alpha_i y_i = 0 \text{ and } \alpha_i \geq 0 \text{ for all } i$$

Maximal Margin Classifier

- The cost functional to be maximized depends only on the input patterns in the form of a set of inner products,

$$\langle x_i, x_j \rangle, \qquad i, j$$

- The relation $w = \sum_{i=1}^{n} \alpha_i y_i x_i$ shows that the hypothesis can be described as a linear combination of the training points.
- Optimal weight: $w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$

Maximal Margin Classifier

- **KKT conditions**: for all i

$$\alpha_i^*\left[1-y_i\langle w^*,x_i\rangle-y_i b^*\right]=0,\quad 1-y_i\left[\langle w^*,x_i\rangle+b^*\right]\le 0,\quad \alpha_i^*\ge 0$$

- Define $I_{sv}:=\{i:\alpha_i^*>0\}$
- optimal weight: $w^*=\sum_{i=1}^{n}\alpha_i^* y_i x_i=\sum_{i\in I_{sv}}\alpha_i^* y_i x_i$
- optimal discriminant function:

$$f^*(x)=\langle w^*,x\rangle+b^*=\sum_{i\in I_{sv}}\alpha_i^* y_i\langle x_i,x\rangle+b^*$$

Maximal Margin Classifier

- Optimal discriminant function:

$$f^*(x) = \langle w^*, x \rangle + b^* = \sum_{i \in I_{sv}} \alpha_i^* y_i \langle x_i, x \rangle + b^*$$

where

$$b^* = y_k - \sum_{i \in I_{sv}} \alpha_i^* y_i \langle x_i, x_k \rangle, \qquad \alpha_k^* > 0.$$

- Obviously, the Lagrange multiplier associated with each point quantifies how important a given training is in forming the final solution.

Maximal Margin Classifier

- Points that have zero α_i^* have no influence
- For any $i \in I_{sv}$ we have $\alpha_i^* > 0$

KKT conditions $\Rightarrow y_i\left[\langle w^*, x_k \rangle + b^*\right] = 1$

- This implies that the functional margin of (x_i, y_i) with respect to the maximal margin hyperplane is one and therefore lies closest to the maximal margin hyperplane.

Maximal Margin Classifier

- **positive support vector**: any pattern x_i with $i \in I_{sv}$ and $y_i = 1$
- **negative support vector**: any pattern x_i with $i \in I_{sv}$ and $y_i = -1$
- In conceptual terms, the support vectors are those data points that lie closest to the decision surface and are therefore the most difficult to classify.
- The fact that only a subset of the Lagrange multipliers is nonzero is referred to as **sparseness**, and means that the support vectors contain all the information necessary to reconstruct the optimal hyperplane.

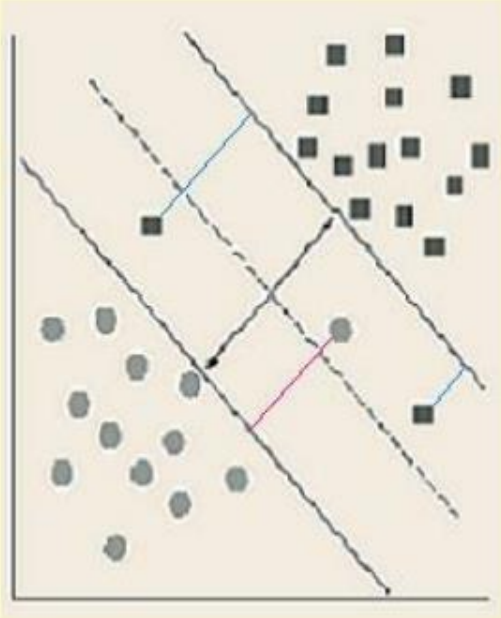
Slack Variable for Classification

- **Definition**: Let $\gamma>0$ be given. The **margin slack variable** ξ_i of an example (x_i, y_i) with respect to the hyperplane $H:(w, b)$ and target margin γ is defined by

$$\xi_i := \max(0, \gamma - y_i \cdot [\langle w, x_i \rangle + b])$$

- From the definition, we have

$$\xi_i \geq 0, \xi_i + y_i \cdot [\langle w, x_i \rangle + b] \geq \gamma.$$



Slack Variable for Classification

- For simple geometric interpolation, we assume that $\|w\| = 1$. Let H^+ (resp., H^-) be the hyperplane in the positive (resp., negative) region parallel to H and g distance apart. Thus we have

$$H^+ := \{x \in R^n : \langle w, x \rangle + b - \gamma = 0\},$$

$$H^- := \{x \in R^n : \langle w, x \rangle + b + \gamma = 0\}.$$

- The open region between H^+ and H^- is called the **region of separation**.

Slack Variable for Classification

- The quantity defined by

$$\|\xi\|_2 := \left(\sum_{i=1}^{n} \xi_i^2 \right)^{1/2} \quad \text{or} \quad \|\xi\|_1 := \sum_{i=1}^{n} \xi_i$$

measures the amount by which the training set fails to have margin, and takes into account any misclassifications of the training data.

1-norm Soft Margin Classifier

- **Primal problem**:

  minimize $2^{-1}\langle w, w\rangle + C\sum_{i=1}^{n}\xi_i$

  subject to $y_i \cdot \left[\langle w, x_i\rangle + b\right] \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all i

- **Dual problem**:

  maximize $\sum_{i=1}^{n}\alpha_i - 2^{-1}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle$

  subject to $\sum_{i=1}^{n}\alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C$ for all i

Lagrange multiplier

$$L = \frac{1}{2}\| w \|^2 + C\sum_i \xi_i + \sum_i \alpha_i [1 - \xi_i - y_i(wx_i - b)] - \sum_i \pi_i \xi_i$$

KKT conditions

$$\nabla_w = w - \sum_i \alpha_i y_i x_i = 0$$

$$\frac{\partial L}{\partial b} = \sum_i \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \xi} = C - \alpha_i - \pi_i = 0$$

$$\alpha_i \geq 0 \quad \alpha_i [1 - \xi_i - y_i(wx_i - b)] = 0$$

$$\pi_i \geq 0 \quad \pi_i \xi_i = 0$$

Lagrange multiplier

It follows

$$w = \sum_i \alpha_i y_i x_i$$

$$\sum_i \alpha_i y_i = 0 \quad 0 \le \alpha_i \le C$$

$$\alpha_i [1 - \xi_i - y_i (w x_i - b)] = 0$$

$$\pi_i \ge 0 \quad \pi_i \xi = 0$$

$$C - \alpha_i - \pi_i = 0$$

Determine α， then obtain (w, b)

1-norm Soft Margin Classifier

- Define $I_{sv} := \{i : \alpha_i^* > 0\}$.
- Optimal weight: $w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i = \sum_{i \in I_{sv}} \alpha_i^* y_i x_i$
- Optimal discriminant function:

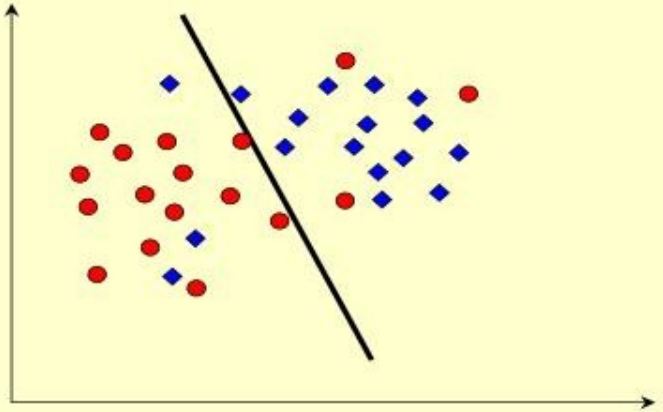
$$f^*(x) = \langle w^*, x \rangle + b^* = \sum_{i \in I_{sv}} \alpha_i^* y_i \langle x_i, x \rangle + b^*$$

where

$$b^* = y_k - \sum_{i \in I_{sv}} \alpha_i^* y_i \langle x_i, x_k \rangle, \qquad 0 < \alpha_k^* < C.$$

Nonlinear Classifier

- 最大化间隔
- 最小化误差
- 两者折衷
- 两种方法: 罚函数方法, 核方法 (特征映射)

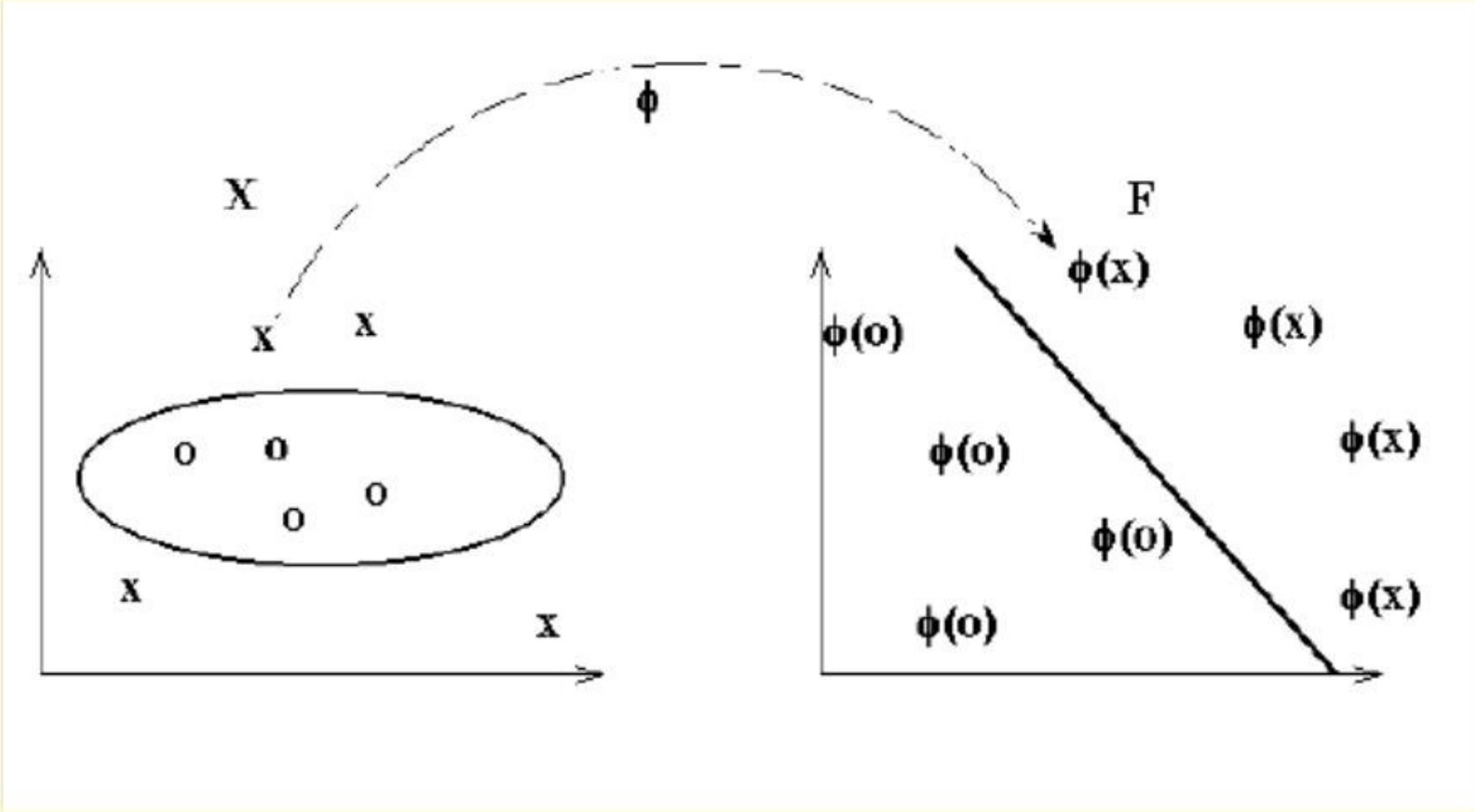


- 非线性可分的数据样本在高维空间有可能转化为线性可分。
- 在训练问题中，涉及到训练样本的数据计算只有两个样本向量点乘
- 使用特征映射 $\phi(x)$，将所有样本点映射到高为空间，则新的样本集为

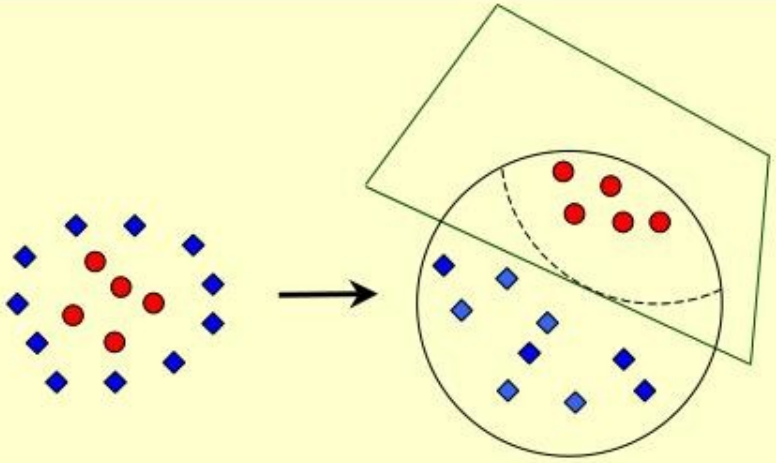
$$((\phi(x_1), y_1), \cdots, (\phi(x_n), y_n))$$

- 核函数 $K(x, y) = \phi(x) \cdot \phi(y)$

Feature Map



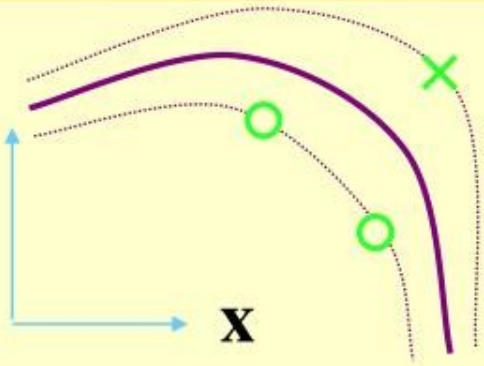
Feature Map



- 不需要映射的显示表示
- 用核函数代替点积

$$K(x,x_i)=\varphi(x)\cdot\varphi(x_i)$$

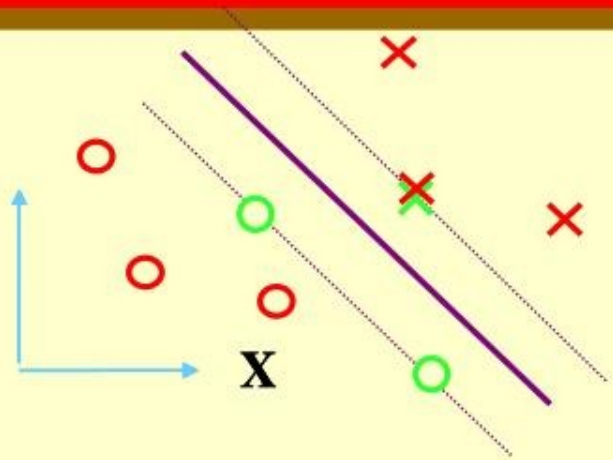
SVM (Kernel Machine)



$\Phi(\cdot)$

$\mathbf{X}_i = \Phi(\mathbf{x}_i)$

$\mathbf{X} = \Phi(\mathbf{x})$



$$y = \text{sign}(\sum_{i \in S} \alpha_i y_i \mathbf{X}_i \cdot \mathbf{X} + b)$$

$\mathbf{X}_i \cdot \mathbf{X} = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})$

$$y = \text{sign}(\sum_{i \in S} \alpha_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) + b)$$

$K(\cdot,\cdot)$

$\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) = K(\mathbf{x}_i, \mathbf{x})$

$$y = \text{sign}(\sum_{i \in S} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b)$$

Kernel

- **Definition**: Let $(F,\langle .,. \rangle)$, called the feature space, be a real inner product space and $X \subseteq R^n$. A kernel is a real-valued function on $X \times X$ such that

$$K(x,z) := \langle \varphi(x), \varphi(z) \rangle, \qquad x,z \in X,$$

where ϕ, called the feature map, is a mapping from X to F.

Kernel

- The idea of a kernel generalizes the standard inner product in R^n by making the feature map the identity map, i.e.,

$$K(x,z) := \langle \phi(x), \phi(z) \rangle = \langle x, z \rangle := x^T z$$

- **Example**: Suppose we let $\phi(x) := Ax$. Then we have

$$K(x,z) := \langle \phi(x), \phi(z) \rangle = \langle Ax, Az \rangle = x^T A^T A z := x^T B z,$$

where $B := A^T A$ is a real symmetric positive semi-definite matrix.

Kernel

- **Example**: Suppose we first specify the kernel as

$$K(x,z) := \langle x,z\rangle^2 = (x^T z)^2, \qquad x = (x_1,\dots,x_n)^T, \qquad z := (z_1,\dots,z_n)^T \in R^n.$$

$$K(x,z) = (x^T z)^2 = \left(\sum_{i=1}^n x_i z_i\right)^2 = \left(\sum_{i=1}^n x_i z_i\right)\cdot\left(\sum_{j=1}^n x_j z_j\right)$$

$$= \sum_{i=1}^n \sum_{j=1}^n x_i x_j z_i z_j = \sum_{(i,j)=(1,1)}^{(n,n)} (x_i x_j)(z_i z_j)$$

- Feature map:

$$\phi(x) = (x_i x_j)_{(i,j)=(1,1)}^{(n,n)}, \qquad x = (x_1,\dots,x_n)^T \in R^n.$$

Kernel

- More generally, we may specify the kernel as

$$K(x,z) := \left(\langle x,z\rangle + c\right)^2 = \left(x^T z + c\right)^2, \qquad c>0, \quad x=(x_1,\dots,x_n)^T, \qquad z := (z_1,\dots,z_n)^T \in R^n.$$

$$K(x,z) = \left(x^T z + c\right)^2 = \left(\sum_{i=1}^n x_i z_i + c\right)^2 = \left(\sum_{i=1}^n x_i z_i + c\right)\cdot\left(\sum_{j=1}^n x_j z_j + c\right)$$

$$= \sum_{i=1}^n \sum_{j=1}^n x_i x_j z_i z_j + 2c\sum_{i=1}^n x_i z_i + c^2 = \sum_{i=1}^n \sum_{j=1}^n x_i x_j z_i z_j = \sum_{(i,j)=(1,1)}^{(n,n)} (x_i x_j)(z_i z_j)$$

- Feature map:

$$\phi(x) = \left(c, \left(\sqrt{2c}\,x_i\right)_{i=1}^n, \left(x_i x_j\right)_{(i,j)=(1,1)}^{(n,n)}\right), \qquad x=(x_1,\dots,x_n)^T \in R^n$$

Kernels

- Polynomial (degree d)

$$K(x,z) := \left(\langle x,z\rangle + c\right)^d = \left(x^T z + c\right)^d, \qquad d \geq 2.$$

In these cases, the decision boundary in the input space corresponding to a hyperplane in these feature spaces is a polynomial curve of degree d, so these kernel are frequently called **polynomial kernels**.

- Two layer neural network

$$K(\mathbf{x},\mathbf{x}') = \tanh\left(\upsilon(\mathbf{x}\cdot\mathbf{x}') + \Theta\right)$$

- Radial basis functions (Gauss)

$$K(\mathbf{x},\mathbf{x}') = e^{\frac{-\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}}$$

Maximal Margin Classifier in the Feature Space

- Let $X \subseteq R^n, \qquad Y := \{+1,-1\}.$
- training examples: $S := \{(x_i, y_i)\}_{i=1}^n \subseteq X \times Y$
- kernel: $K(x,z) := \langle \phi(x), \phi(z) \rangle, \qquad x, z \in X.$

maximize $$\sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle \phi(x_i), \phi(x_j) \rangle = \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

subject to $$\sum_{i=1}^n \alpha_i y_i = 0 \quad \text{and} \quad \alpha_i \geq 0 \ \text{for all} \ i$$

Maximal Margin Classifier in the Feature Space

- Define $I_{sv} := \{i : \alpha_i^* > 0\}$
- optimal weight: $w^* = \sum_{i=1}^{n} \alpha_i^* y_i \phi(x_i) = \sum_{i \in I_{sv}} \alpha_i^* y_i \phi(x_i)$
- optimal discriminant function:

$$f^*(x) = \sum_{i \in I_{sv}} \alpha_i^* y_i \langle \phi(x_i), \phi(x) \rangle + b^* = \sum_{i \in I_{sv}} \alpha_i^* y_i K(x_i, x) + b^*$$

where

$$b^* = y_k - \sum_{i \in I_{sv}} \alpha_i^* y_i \langle \phi(x_i), \phi(x_k) \rangle = y_k - \sum_{i \in I_{sv}} \alpha_i^* y_i K(x_i, x_k), \qquad \alpha_i^* > 0$$

- No need to calculate the any features to form the final discriminant function.
- Kernel is just good enough.

1-norm Soft Margin Classifier in the Feature Space

maximize $$\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \langle \phi(x_i), \phi(x_j)\rangle = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i, x_j)$$

subject to $$\sum_{i=1}^{n}\alpha_i y_i = 0 \quad \text{and} \quad 0 \le \alpha_i \le C \quad \text{for all} \quad i$$

- Define $I_{sv} := \{i \in \underline{l} : \alpha_i^* > 0\}$.
- optimal weight: $w^* = \sum_{i=1}^{l}\alpha_i^* y_i \phi(x_i) = \sum_{i \in I_{sv}}\alpha_i^* y_i \phi(x_i)$
- optimal discriminant function:

$$f^*(x) = \sum_{i \in I_{sv}}\alpha_i^* y_i \langle \phi(x_i), \phi(x)\rangle + b^* = \sum_{i \in I_{sv}}\alpha_i^* y_i K(x_i, x) + b^*$$

where

$$b^* = y_k - \sum_{i \in I_{sv}}\alpha_i^* y_i \langle \phi(x_i), \phi(x_k)\rangle = y_k - \sum_{i \in I_{sv}}\alpha_i^* y_i K(x_i, x_k), \qquad 0 < \alpha_k^* < C.$$

1-norm Soft Margin Classifier in the Feature Space

- The nonlinear classifier by using the support vector machine can be designed as a feedforward network with a single layer of nonlinear units.
- The curse of dimensionality for the nonlinear classification problem is bypassed by focusing on the dual problem for performing the constrained optimization problem.

Implementation Techniques

- **Sequential Minimal Optimization Technique**

- J. Platt, Fast training of support vector machines using sequential minimal optimization, In B. Schokopf, C. J. C. Burges and A. J. Smola, editors, Advances in Kernel Methods - Support vector Learning, Cambridge, MA, MIT Press, 1999, 185-208.

- J. C. Platt, Sequential minimal optimization: A fast algorithm for training support vector machines, Technical Report MSR-TR-98-14, Microsoft Research, 1998.

The end

Thank you
for your attention

Examples of Animals








Examples of Plants