模式识别的理论与方法

Pattern Recognition

裴继红

Chapter 5 (Part 3):
Support Vector Machines

(Sections 5.6-5.12)

本讲内容

- 感知器算法回顾
- 支撑矢量机

感知器回顾：准则函数

寻找解矢量 w 使对所有训练样本，下面不等式成立

$$\boldsymbol{w}^{t}\boldsymbol{y}_i > 0$$

感知器准则函数:

$$J_p(\mathbf{w}) = \sum_{\mathrm{y}\in\mathbf{Y}_E}\left(-\mathbf{w}^t\mathrm{y}\right)$$

上面准则函数中的 Y_E 是所有被矢量 $\boldsymbol{w}$ 误分的样本集合。
准则函数梯度为

$$\nabla J_p(\mathbf{w}) = \sum_{\mathrm{y}\in\mathbf{Y}_E}\left(-\mathrm{y}\right)$$

感知器回顾：学习规则

注意到 $\nabla J_p(\mathbf{w}) = \sum_{\mathrm{y}\in \mathbf{Y}_E} (-\mathrm{y})$

因此，权矢量的更新规则变为:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \eta(k) \sum_{y\in \mathbf{Y}_E} \mathrm{y}$$

其中，Y_{E}是被矢量 $\boldsymbol{w}(k)$ 错分的样本集合。

学习规则总结如下:

新的权矢量 **=** 当前权矢量
+ 被当前权矢量误分的所有样本之和×学习因子

感知器学习算法-1：批处理感知器算法

算法的实现步骤：

1. 初始化：权矢量 $\boldsymbol{w}$，学习步长$\eta(.)$，误差阈值θ，令 $\boldsymbol{k}=0$

2. do $\boldsymbol{k}=\boldsymbol{k}+1$

3. $$\mathbf{w}(k+1)=\mathbf{w}(k)+\eta(k)\sum_{y\in Y_k} y$$

4. while $\left|\eta(k)\sum_{y\in Y_k} y\right|>\theta$

5. Return $\boldsymbol{w}$

6. 结束

感知器学习算法-2：
——固定增量单样本感知器算法

算法的实现步骤:（训练样本已经过规范化）

1. 初始化：权矢量 $\boldsymbol{w}$，学习步长$\eta(.)$，误差阈值θ，令$\boldsymbol{k}=0$
2. do $\boldsymbol{k}=(\boldsymbol{k}+1) \bmod n$ （n为样本总数）
3. $\mathbf{w}(k+1)=\mathbf{w}(k)-y(k)\cdot\left(\operatorname{sign}[\mathbf{w}^t y(k)]-1\right)/2$
4. while 存在 $\boldsymbol{w}^t \boldsymbol{y}(\boldsymbol{k})<0$, $k=1,2,\ldots n$
5. Return $\boldsymbol{w}$
6. 结束

- 算法的特点

感知器的对偶算法

由权矢量的更新规则:

$$\mathbf{w}(k+1)=\mathbf{w}(k)+\eta(k)\sum_{y\in\mathbf{Y}_E}\mathrm{y}$$

其中，Y_E是被矢量 $w(k)$ 错分的样本集合。

令 $w(0)=y_1$, 即将权向量初始值选为第一个样本的向量的值; $\eta(k)=1$。
则上述学习过程结束后，解可以等效为如下形式:

$$\mathbf{w}(k)=\sum_{i=1}^{n}\alpha_i\mathbf{y}_i \qquad \mathbf{w}=\frac{\mathbf{w}(k)}{\|\mathbf{w}(k)\|}$$

其中，α_i 是在整个学习过程中，样本矢量 y_i 被错分的次数。

感知器学习算法-3：
——单样本感知器对偶算法

算法的实现步骤:（训练样本已经过规范化）

1. 初始化： $\boldsymbol{w} = \boldsymbol{w}(0) = \mathbf{y}(\mathbf{1})$，$\alpha_i=0$，i=1,2,…,n， 误差阈值$\theta$， 令 $\boldsymbol{k}=0$
2. do $\boldsymbol{k}=$（$\boldsymbol{k}+1$）mod n (n为样本总数)
3. $$\alpha_i = \alpha_i - \left(\mathrm{sign}[\mathbf{w}^t y_i(k)] - 1\right)/2$$
   $$\mathbf{w}(k) = \sum_{i=1}^{n} \alpha_i \mathbf{y}_i \qquad \mathbf{w} = \mathbf{w}(k) / \|\mathbf{w}(k)\|$$
4. while 存在 $\boldsymbol{w}^t \boldsymbol{y}(\boldsymbol{k}) < 0$, $k=1,2,\ldots n$
5. Return $\boldsymbol{w}$
6. 结束

感知器学习算法-3：对偶算法

算法的实现步骤：

1. 初始化：$\boldsymbol{w}(0)=\mathbf{y}_1$， $\alpha_i=0$，i=1,2,…,n，误差阈值θ，令 $\boldsymbol{k}=0$
2. do $\boldsymbol{k}=\boldsymbol{k}+1$
3. $$a(k+1)=a(k)+\eta(k)\sum_{y\in Y_k}y$$
4. while
5. Return $\boldsymbol{a}$ $\quad \left|\eta(k)\sum_{y\in Y_k}y\right|>\theta$
6. 结束

- 算法的特点

支撑向量机：SVM

Support Vector Machines

支撑向量机(SVM)

- SVM是由V. Vapnik等学者在1992年根据统计学习理论推导出的一个优秀的分类器。

- SVM当前已成为最重要的线性分类器之一，被广泛应用于目标检测识别、基于内容的图像检索、文本识别、手写体识别、生物统计学、语音识别等领域.

V. Vapnik

判别函数回顾

- 在2.4节中指出: 分类器用于将一个特征向量 x 指定到类 w_i， 若

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \qquad \text{对所有} \quad j \neq i$$

- 对两类情况, $$g(\mathbf{x}) \equiv g_1(\mathbf{x}) - g_2(\mathbf{x})$$

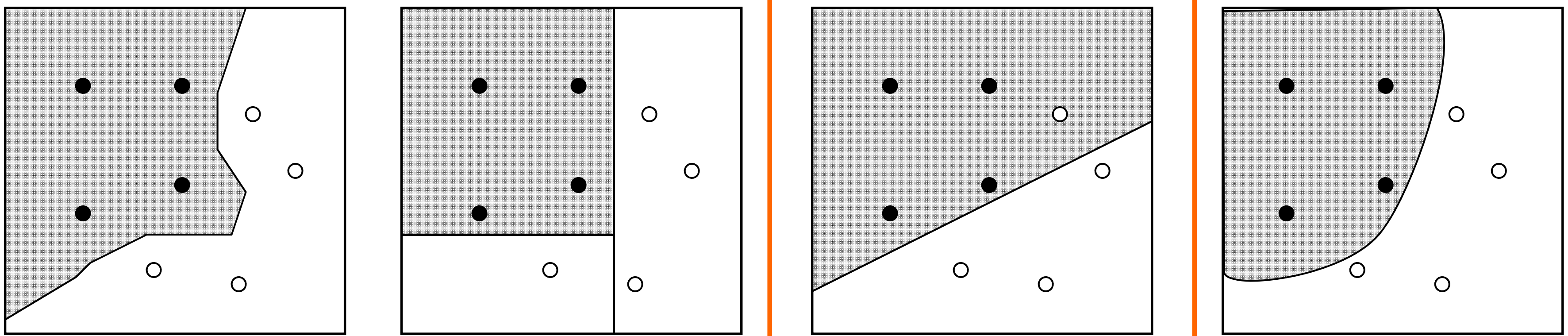
若 $g(\mathbf{x}) > 0$，决策为 ω_1 ；否则，决策为 ω_2

- 例如:
  - 最小错误率分类器

$$g(\mathbf{x}) \equiv p(\omega_1 \mid \mathbf{x}) - p(\omega_2 \mid \mathbf{x})$$

判别函数

- 可以是 *x* 的任意函数, 例如:



Nearest Neighbor | Decision Tree | Linear Functions | Nonlinear Functions

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

线性判别函数

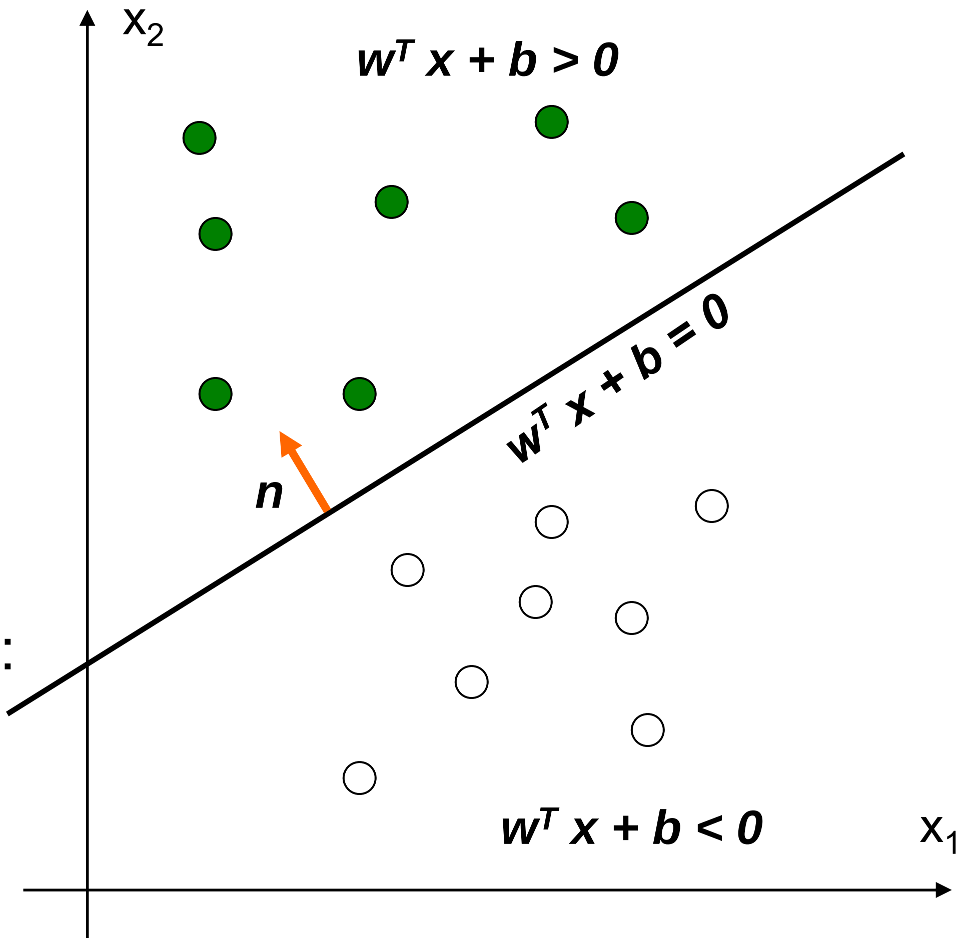
- g(x) 是一个线性函数:

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$

- 线形判别函数是特征空间的一个超平面

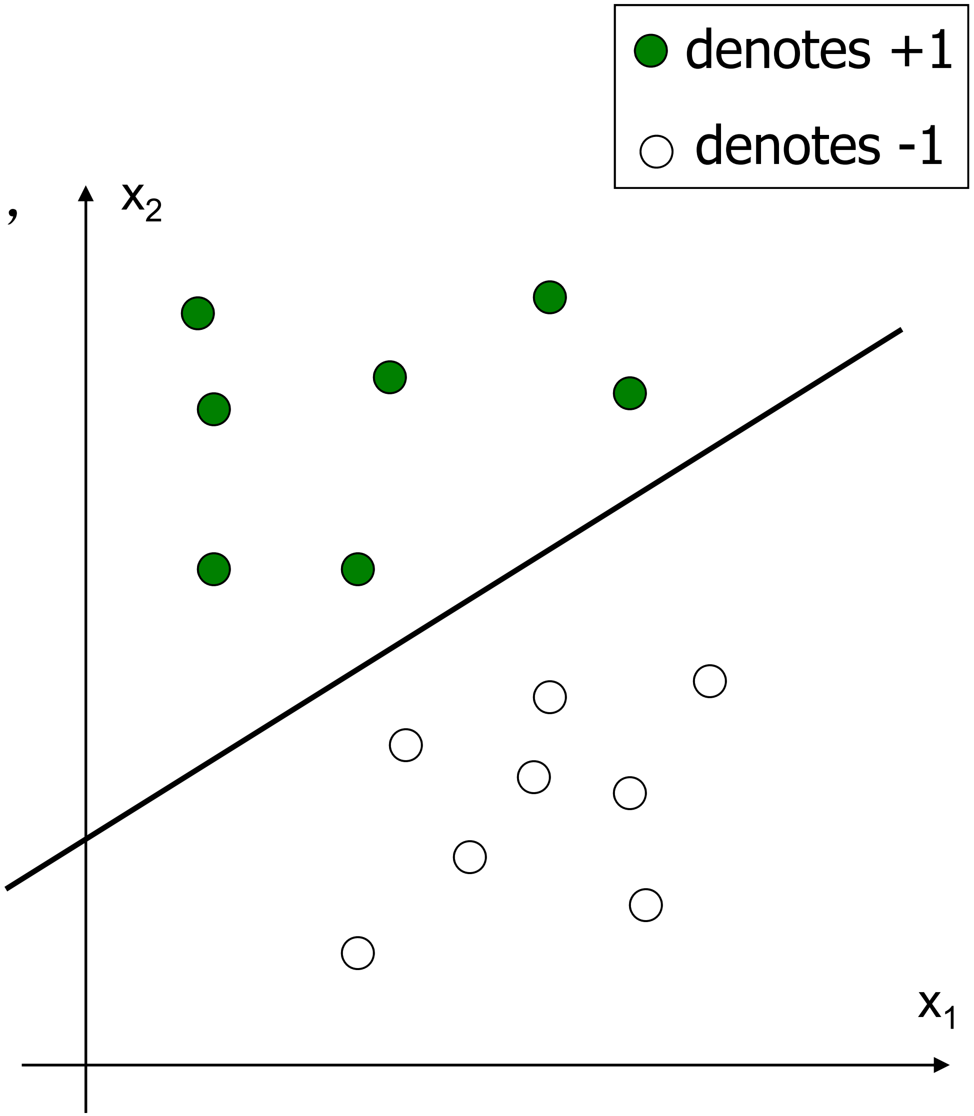
- 超平面的法线量(单位向量) :

$$\mathbf{n} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$$



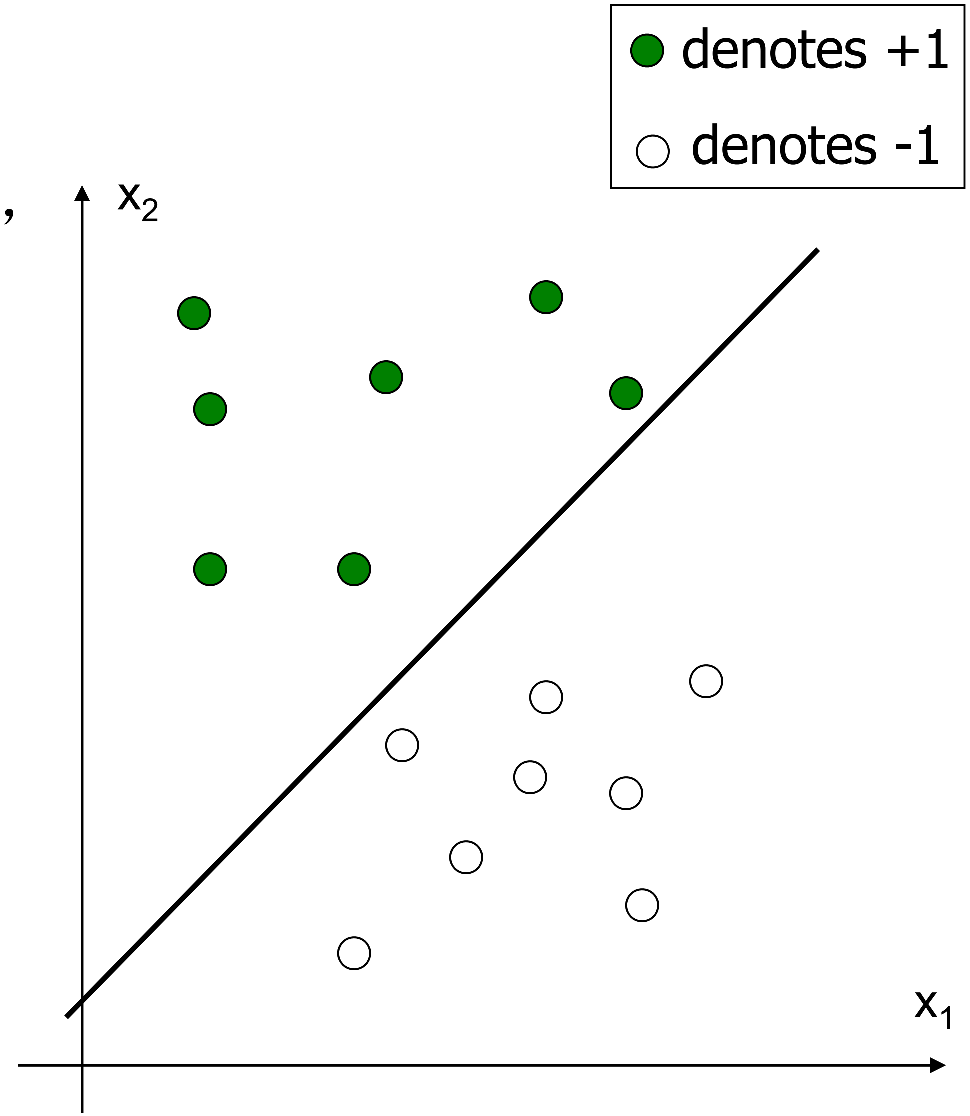
线性判别函数

- 在满足最小错误率的条件下，如何确定一个线性判别函数对这些点进行分类?
- 存在有无穷多个满足条件的可用线性判别函数!



线性判别函数

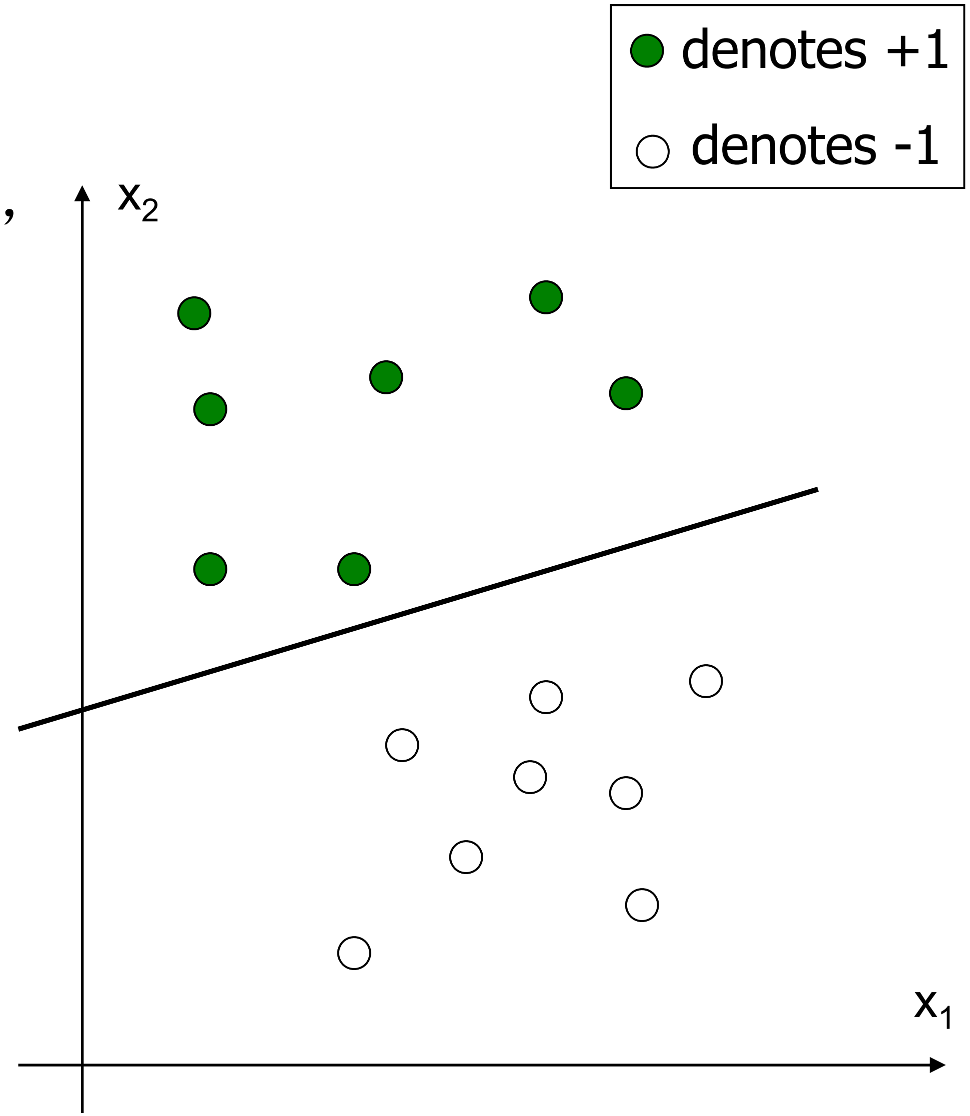
- 在满足最小错误率的条件下，如何确定一个线性判别函数对这些点进行分类?
- 存在有无穷多个满足条件的可用线性判别函数!



线性判别函数

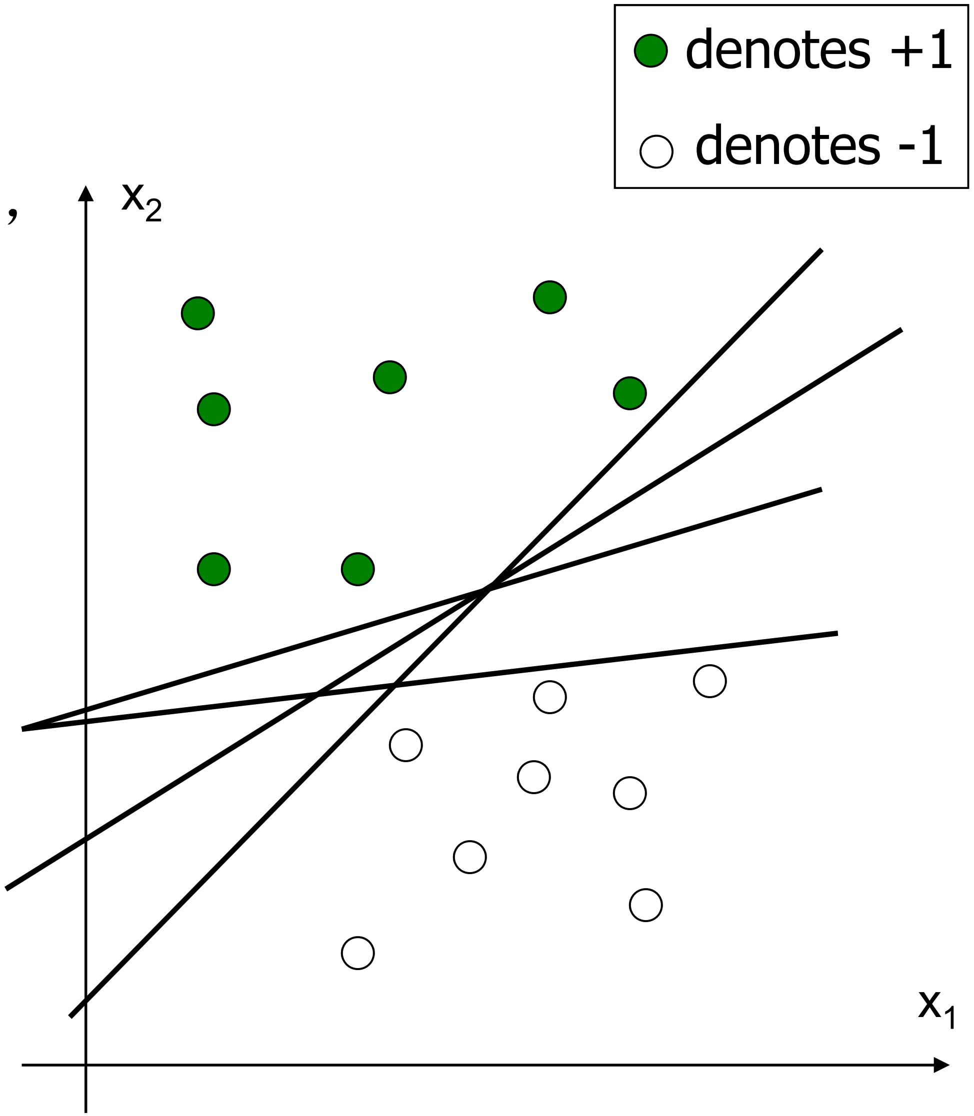
- 在满足最小错误率的条件下，如何确定一个线性判别函数对这些点进行分类?

- 存在有无穷多个满足条件的可用线性判别函数!



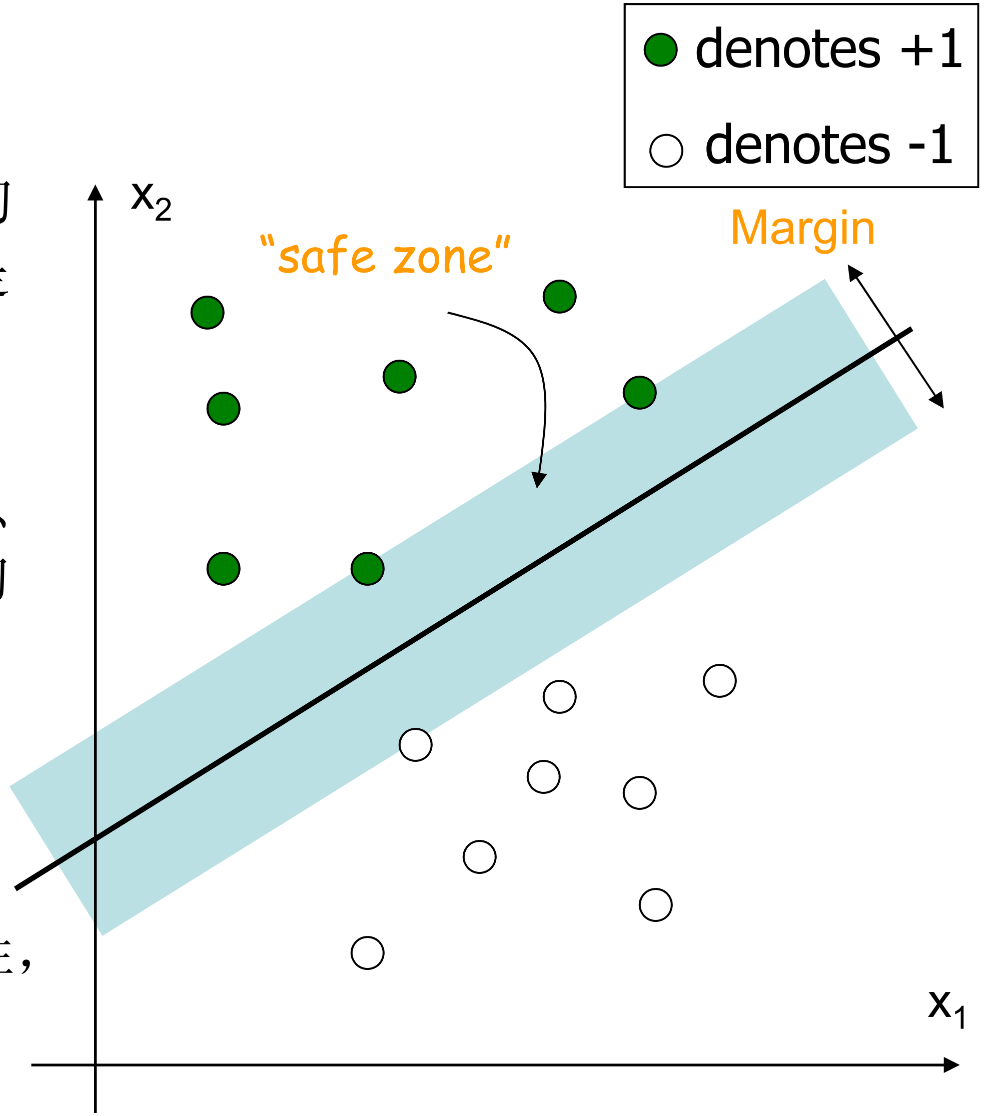
线性判别函数

- 在满足最小错误率的条件下，如何确定一个线性判别函数对这些点进行分类?
- 存在有无穷多个满足条件的可用线性判别函数!
- 那一个是最好的?



大边界线性分类器：线性可分情况

- 具有最大边界（margin）的线性判别函数（分类器）是最好的！
- 边界定义为在碰到数据点以前可以增减的边界线之间的宽度。
- 为什么是最好的?
  - 对噪声、野值点具有鲁棒性，从而具有强壮的推广能力



大边界线性分类器：线性可分情况

- 给定数据点集合:

$$\{(\mathbf{x}_i, y_i)\},\ i = 1, 2, \cdots, n$$

这里

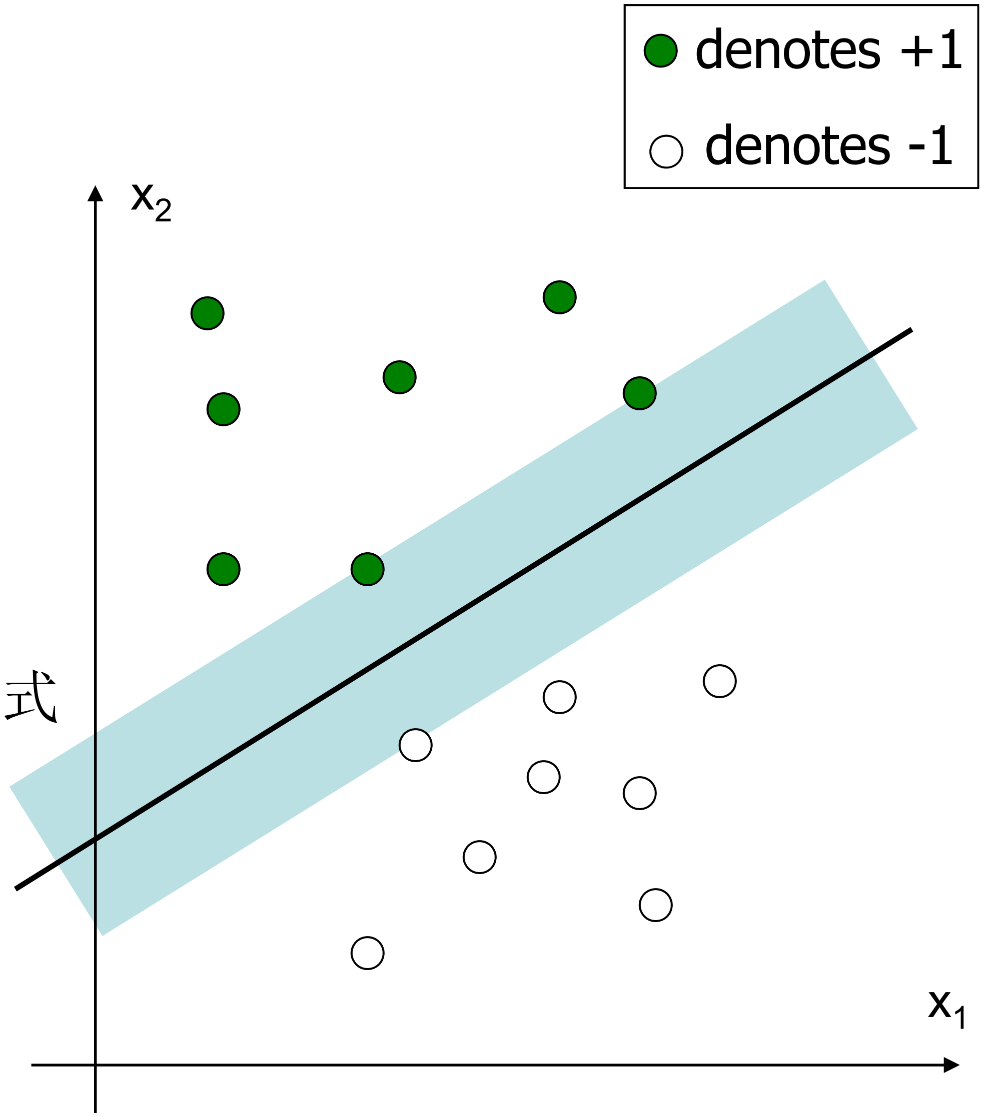
$$y_i = +1,\ \text{若 } \mathbf{w}^T\mathbf{x}_i + b > 0$$

$$y_i = -1,\ \text{若 } \mathbf{w}^T\mathbf{x}_i + b < 0$$

- 对*w*和*b*进行尺度变换, 则上式等价于

$$y_i = +1,\ \text{若 } \mathbf{w}^T\mathbf{x}_i + b \geq 1$$

$$y_i = -1,\ \text{若 } \mathbf{w}^T\mathbf{x}_i + b \leq -1$$



大边界线性分类器：线性可分情况

- 已知

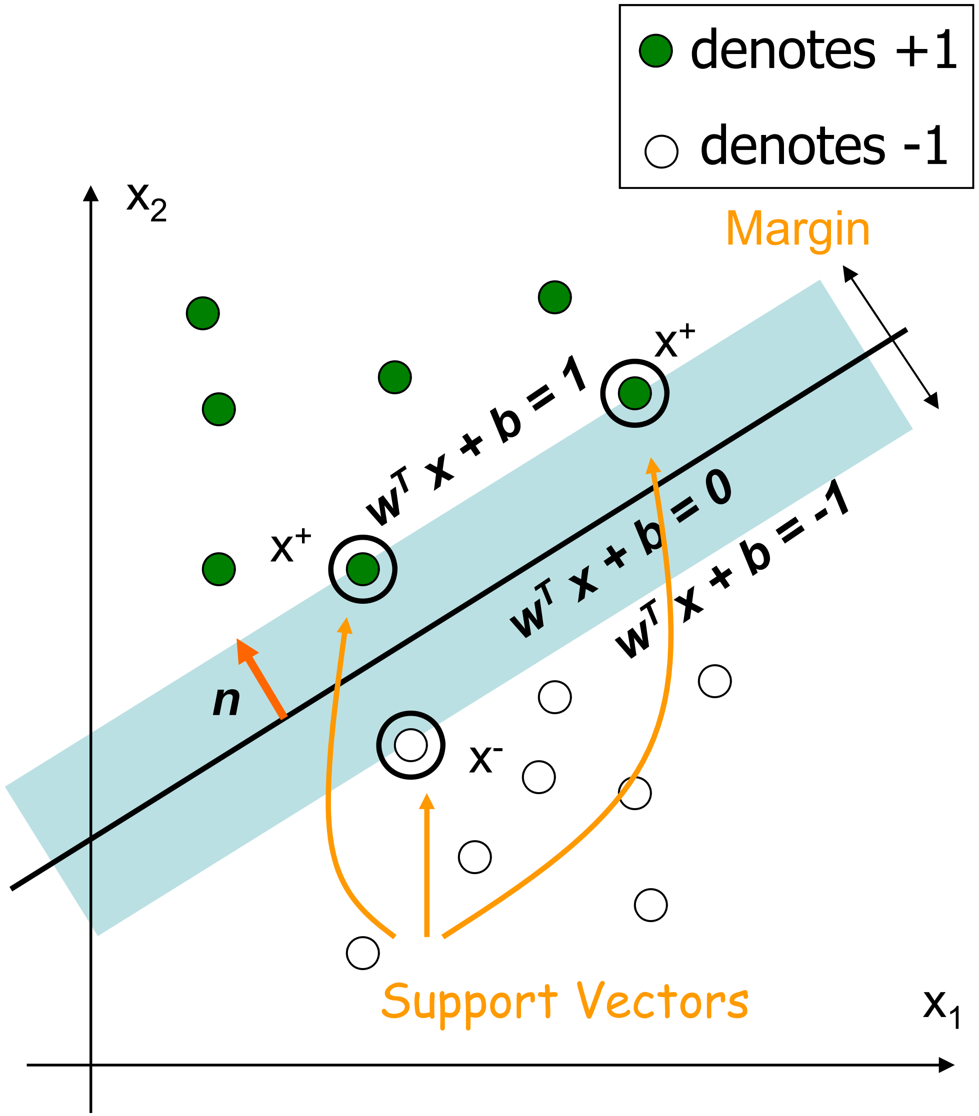
$$\mathbf{w}^T\mathbf{x}^+ + b = 1$$

$$\mathbf{w}^T\mathbf{x}^- + b = -1$$

- 则边界宽度为:

$$M = (\mathbf{x}^+ - \mathbf{x}^-)\cdot\mathbf{n}$$

$$= (\mathbf{x}^+ - \mathbf{x}^-)\cdot\frac{\mathbf{w}}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$$



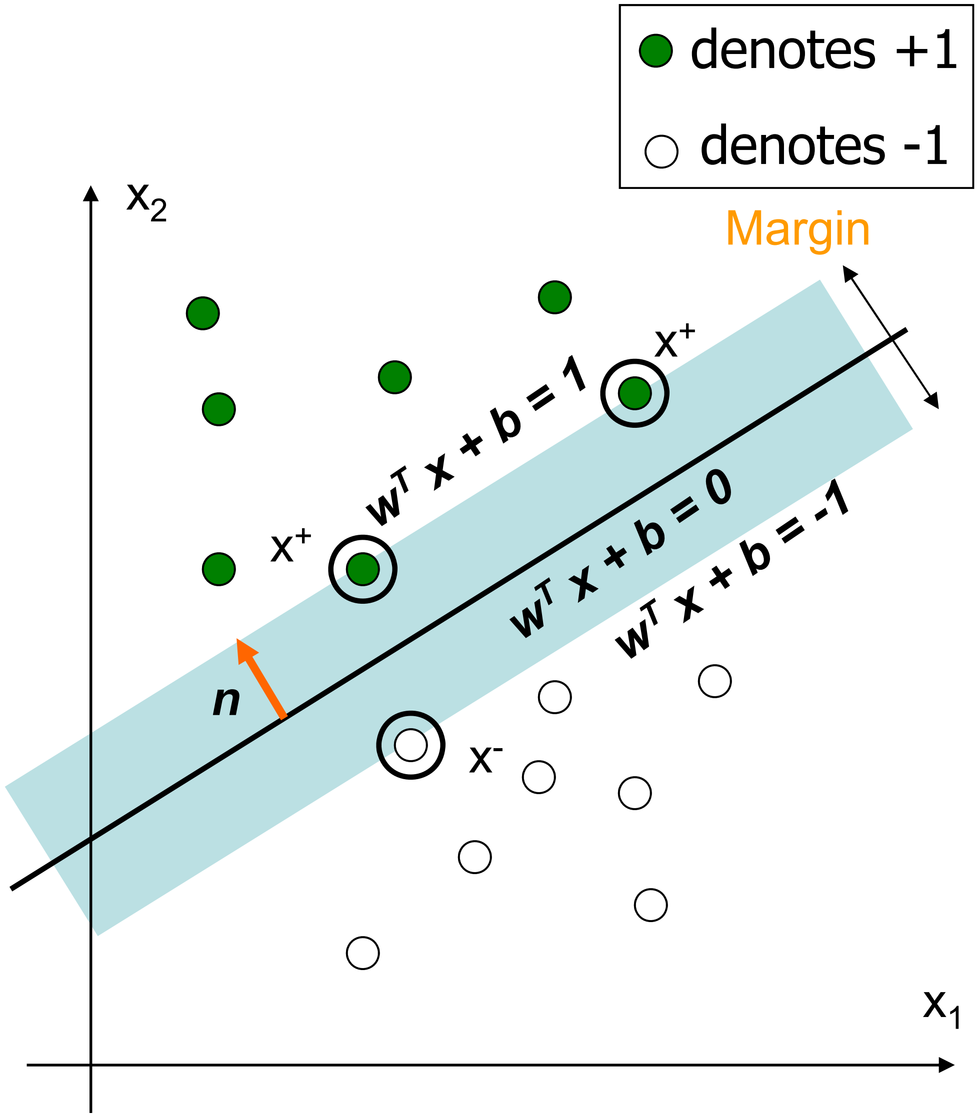
大边界线性分类器：线性可分情况

- 希望的目标:

$$\text{maximize} \quad \frac{2}{\|\mathbf{w}\|}$$

且

$$y_i = +1, \ \text{若}\ \mathbf{w}^T\mathbf{x}_i + b \geq 1$$
$$y_i = -1, \ \text{若}\ \mathbf{w}^T\mathbf{x}_i + b \leq -1$$



大边界线性分类器：线性可分情况

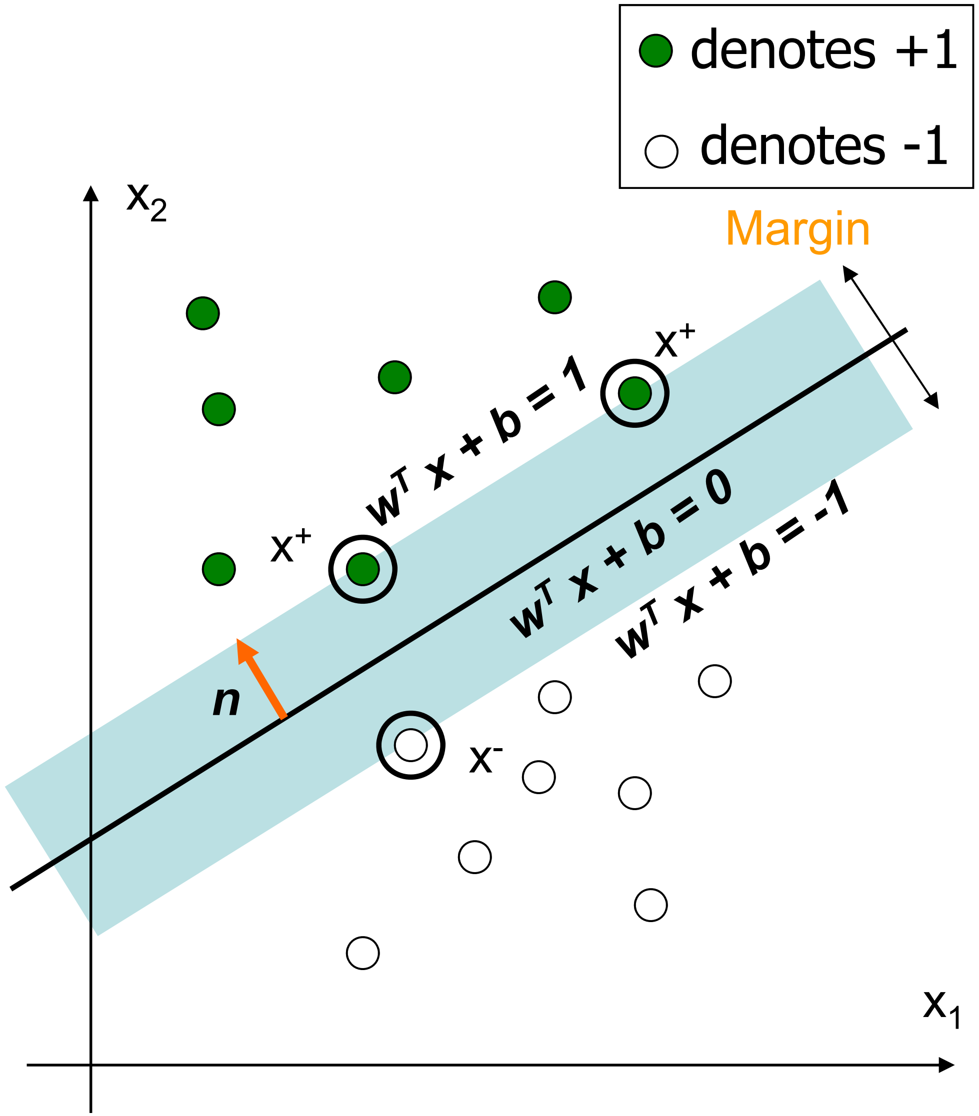
- 希望的目标等价为:

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

且

$$y_i = +1, \ \text{若} \ \mathbf{w}^T\mathbf{x}_i + b \geq 1$$

$$y_i = -1, \ \text{若} \ \mathbf{w}^T\mathbf{x}_i + b \leq -1$$



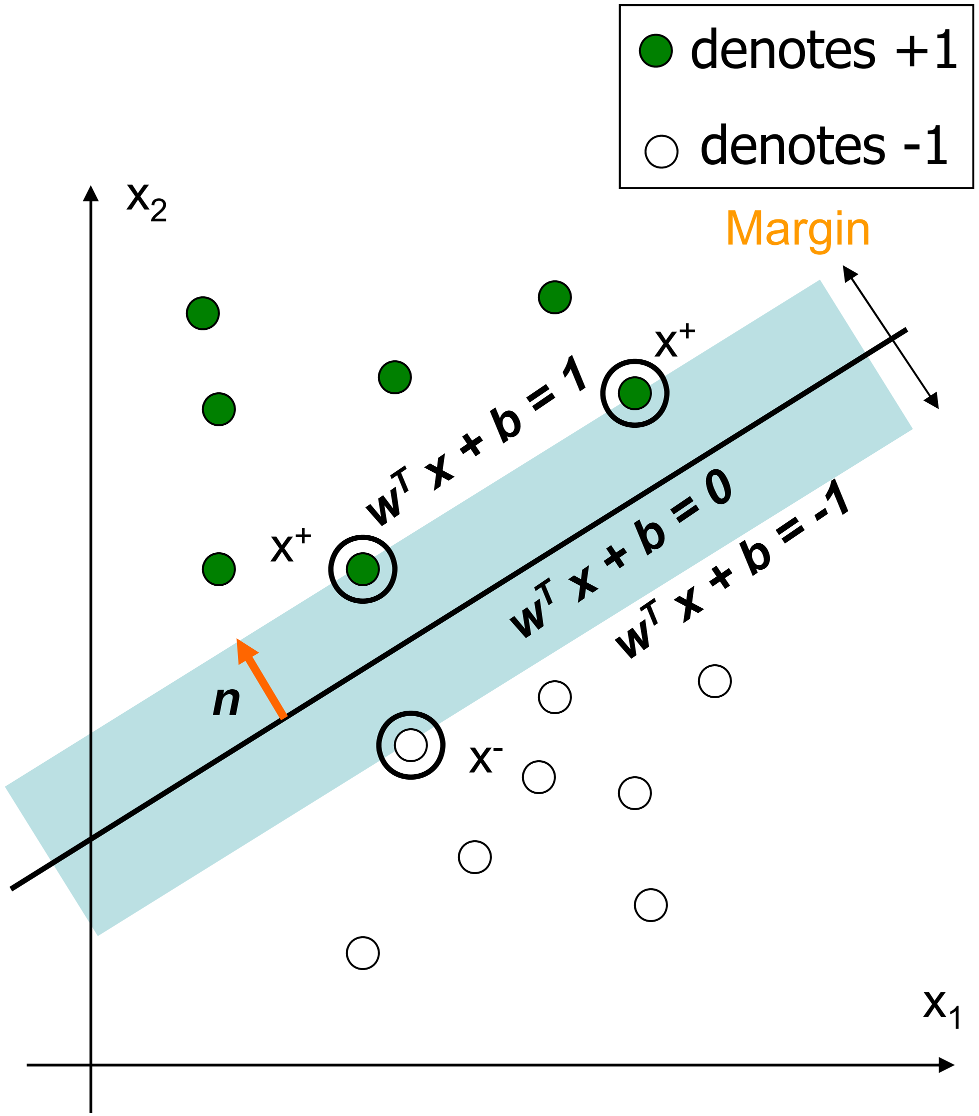
大边界线性分类器：线性可分情况

- 希望的目标可表示为:

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

且

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$



该优化问题的数学描述

具有线性约束的二次规划

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

$$\text{s.t.} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$

Lagrangian（拉格朗日）优化函数

$$\text{minimize} \quad L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n}\alpha_i\left(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1\right)$$

$$\text{s.t.} \quad \alpha_i \geq 0$$

解该优化问题

$$\text{minimize} \quad L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 \right)$$

$$\text{s.t.} \quad \alpha_i \geq 0$$

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L_p}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

解该优化问题

$$\text{minimize} \quad L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 \right)$$

$$\text{s.t.} \quad \alpha_i \geq 0$$

Lagrangian对偶优化函数

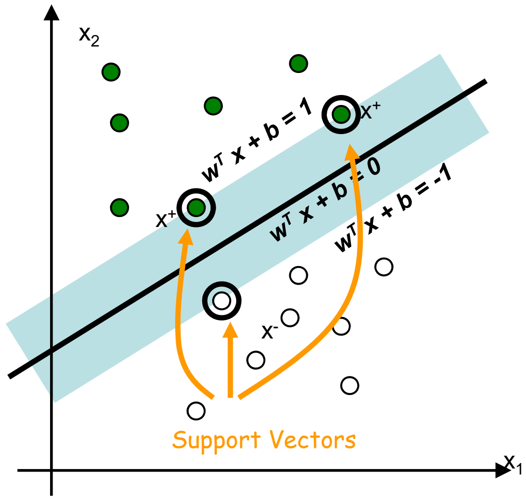
$$\text{maximize} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{s.t.} \quad \alpha_i \geq 0 \text{ , and } \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

解该优化问题

- 由 KKT 条件, 可知:

$$\alpha_i \left(y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 \right) = 0$$



- 这样, 只有支撑向量具有 $\alpha_i \neq 0$

- 因此，解具有如下形式:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = \sum_{i \in \mathrm{SV}} \alpha_i y_i \mathbf{x}_i$$

再由 $y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 = 0$, 可得到$b$

这里 $\mathbf{x}_i$ 是支撑向量support vector

该优化问题解的分析

- 线性判别函数为:

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = \sum_{i \in \mathrm{SV}} \alpha_i \mathbf{x}_i^T \mathbf{x} + b$$

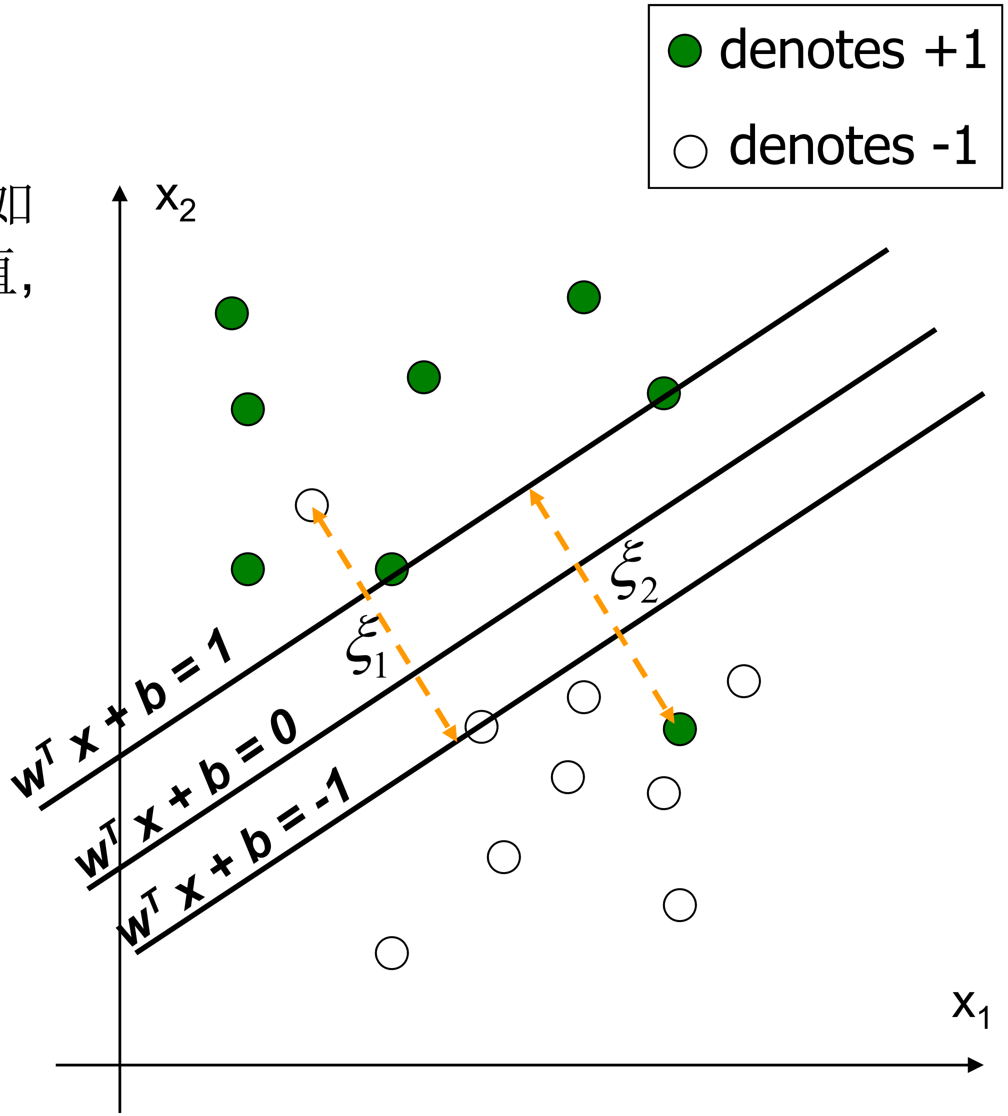
- 注意，上面的函数值只依赖于待测点 ***x*** 和支撑向量 $\boldsymbol{x}_i$ 之间的点积运算

- 回顾解前面解优化问题时，也涉及到计算所有训练样本点的点对之间的点积运算 $\boldsymbol{x}_i^T\boldsymbol{x}_j$

大边界线性分类器：非线性可分情况

- 如果数据点非线性可分会如何? (如出现噪声数据, 野值, 等)

- 可以加入松弛变量（Slack variables） ξ_i，以允许对难划分或噪声数据点产生误分类



大边界线性分类器：非线性可分情况

- 优化模型表示为:

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

这样

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

- 参数 C 可以被看成用于控制过拟合（over-fitting）的一种方式

大边界线性分类器：非线性可分情况

- 表示为Lagrangian 对偶问题的优化模型

$$\text{maximize} \ \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

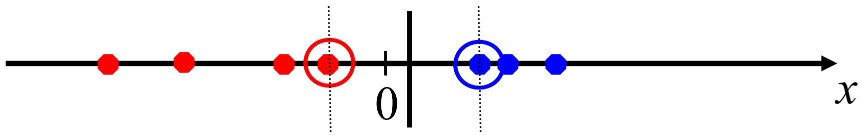
这样

$$0 \le \alpha_i \le C$$

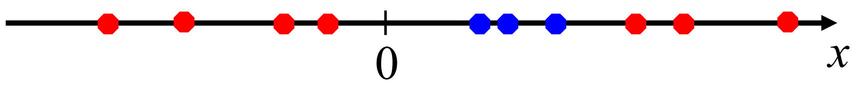
$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

核SVMs

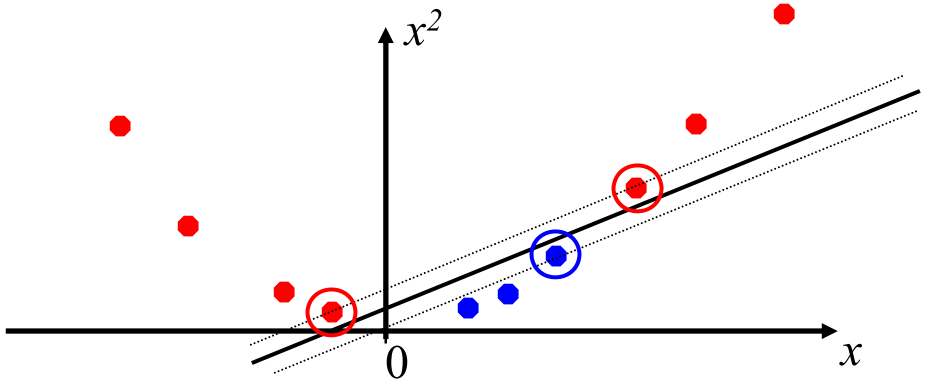
- 数据集合是线性可分时可以解决噪声大的问题:



- 但如果数据集合的分布如下情况时怎么办?

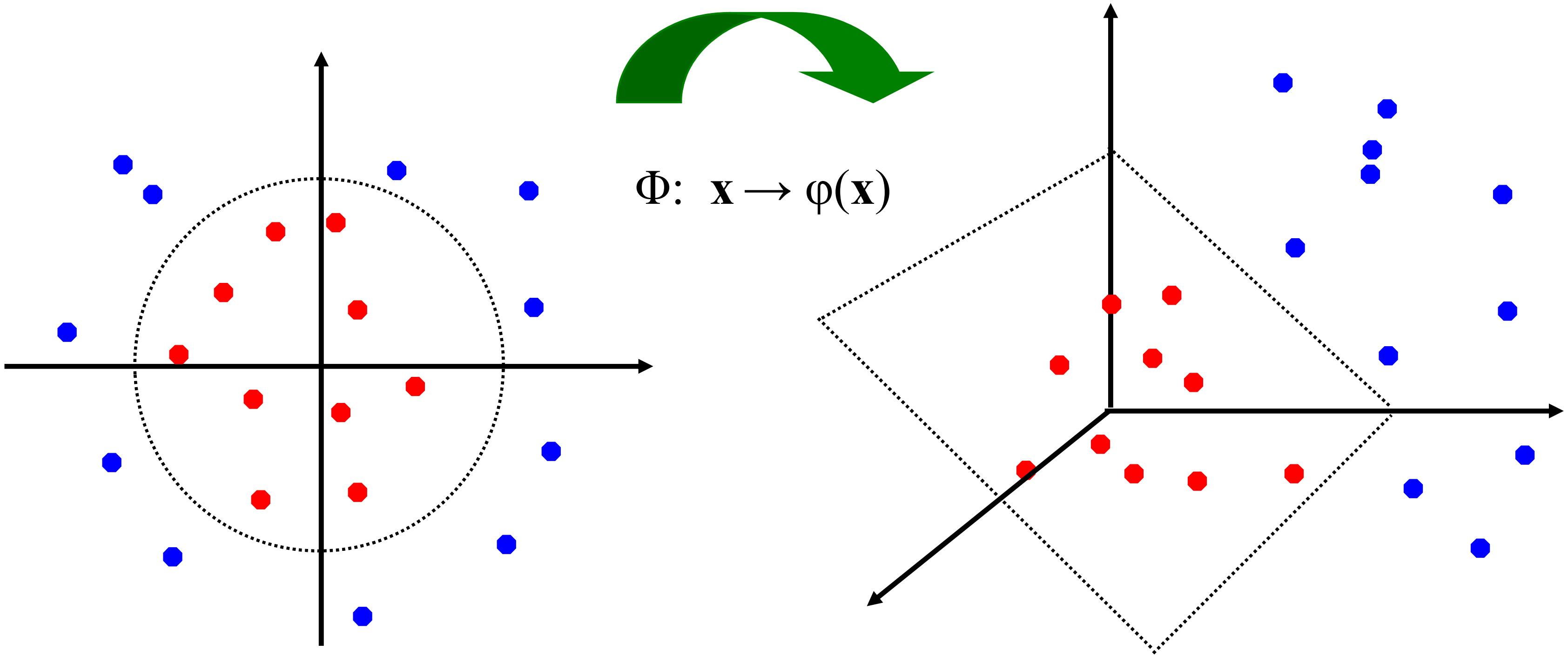


- 可映射到高维空间:



核SVMs：特征空间

- 中心思想：可以将原始输入空间映射到一些高维空间，在该空间中训练集是线性可分的：



核SVMs : 核技巧（The Kernel Trick）

- 利用这一映射, 判别函数现在为:

$$g(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b = \sum_{i\in\mathrm{SV}} \alpha_i \boxed{\phi(\mathbf{x}_i)^T\phi(\mathbf{x})} + b$$

- 不需要明白地搞清楚具体的映射，因为我们只需要使用训练样本和测试样本之间的点积运算.

- 核函数 *kernel function* 是具有如下特点的函数，它在扩展后的特征空间的样本点对的点积运算可以表示为:

$$K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$$

核SVMs : 核函数举例

- 例如:

2-维向量 $\mathbf{x}=[x_1 \ \ x_2]$;

令 $K(\mathbf{x}_i,\mathbf{x}_j)=(1+\mathbf{x}_i^T\mathbf{x}_j)^2$,

需要证明 $K(\mathbf{x}_i,\mathbf{x}_j) = \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$:

$$K(\mathbf{x}_i,\mathbf{x}_j)=(1+\mathbf{x}_i^T\mathbf{x}_j)^2,$$

$$= 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2} + x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$

$$= [1 \ \ x_{i1}^2 \ \ \sqrt{2}\, x_{i1}x_{i2} \ \ x_{i2}^2 \ \ \sqrt{2}x_{i1} \ \ \sqrt{2}x_{i2}]^T [1 \ \ x_{j1}^2 \ \ \sqrt{2}\, x_{j1}x_{j2} \ \ x_{j2}^2 \ \ \sqrt{2}x_{j1} \ \ \sqrt{2}x_{j2}]$$

$$= \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j),$$

这里 $\varphi(\mathbf{x}) = [1 \ \ x_1^2 \ \ \sqrt{2}\, x_1x_2 \ \ x_2^2 \ \ \sqrt{2}x_1 \ \ \sqrt{2}x_2]$

核SVMs：一些常用的核函数

- 一些通常使用的核函数:
  - 线性核: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
  - 多项式核: $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^p$
  - 高斯(径向基函数(RBF))核:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}\right)$$

  - Sigmoid核:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta_0 \mathbf{x}_i^T \mathbf{x}_j + \beta_1)$$

- 一般来说，那些满足*Mercer*条件的函数可以被称为核函数

核SVMs：最优化

- 表示为Lagrangian对偶问题的优化模型

$$\text{maximize} \quad \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

且

$$0 \le \alpha_i \le C$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

- 判别函数的解为

$$g(\mathbf{x}) = \sum_{i \in \mathrm{SV}} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b$$

- 采用的最优化技术是一样的

支撑向量机SVM: 算法

- 1. 选择一个 kernel function
- 2. 选取C的值
- 3. 解二次规划问题 (许多软件包有提供函数)
- 4. 由支撑向量构造判别函数