第4章：支持向量机学习

蒋良孝 |  中国地质大学（武汉）

机器学习与数据挖掘团队

ljiang@cug.edu.cn

http://www.escience.cn/people/jlx/

本章内容

一、最大边缘超平面

二、线性支持向量机

三、非线性支持向量机

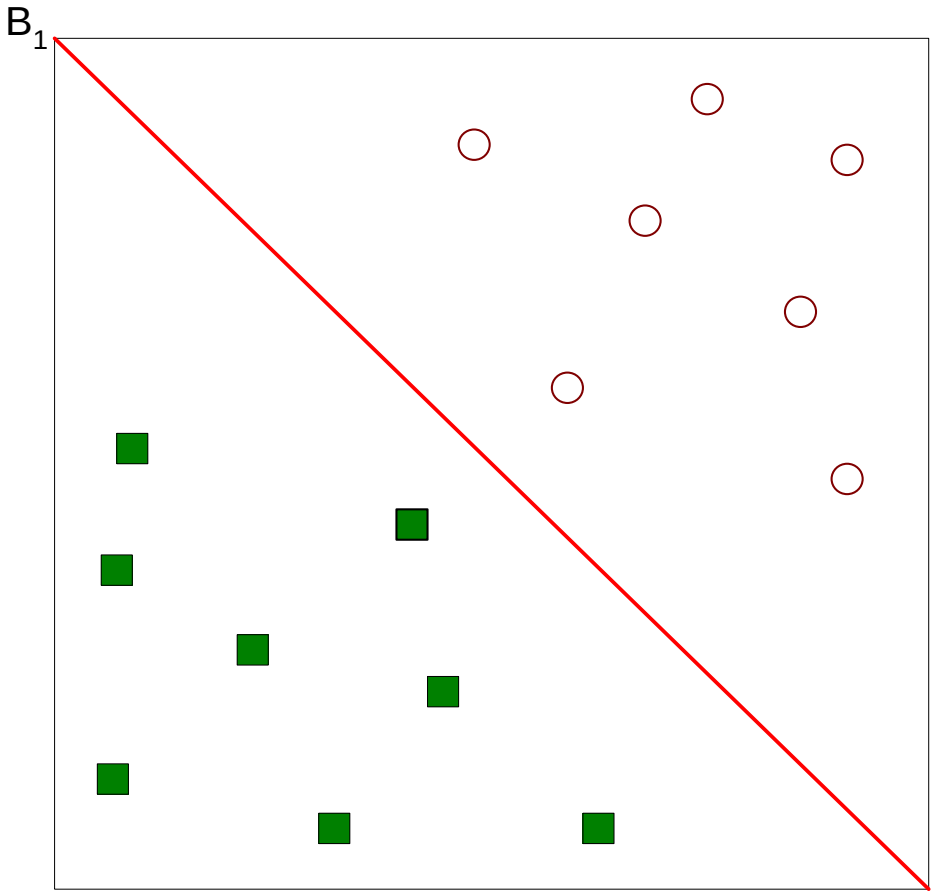
一、最大边缘超平面

- 支持向量机(Support Vector Machine, SVM)不仅具有坚实的统计学理论基础，还可以很好地应用于高维数据、避免维度灾难问题，已经成为一种倍受关注的机器学习分类技术。

- 为了解释SVM的基本思想，我们首先介绍一下最大边缘超平面(Maximal Margin Hyperplane)。

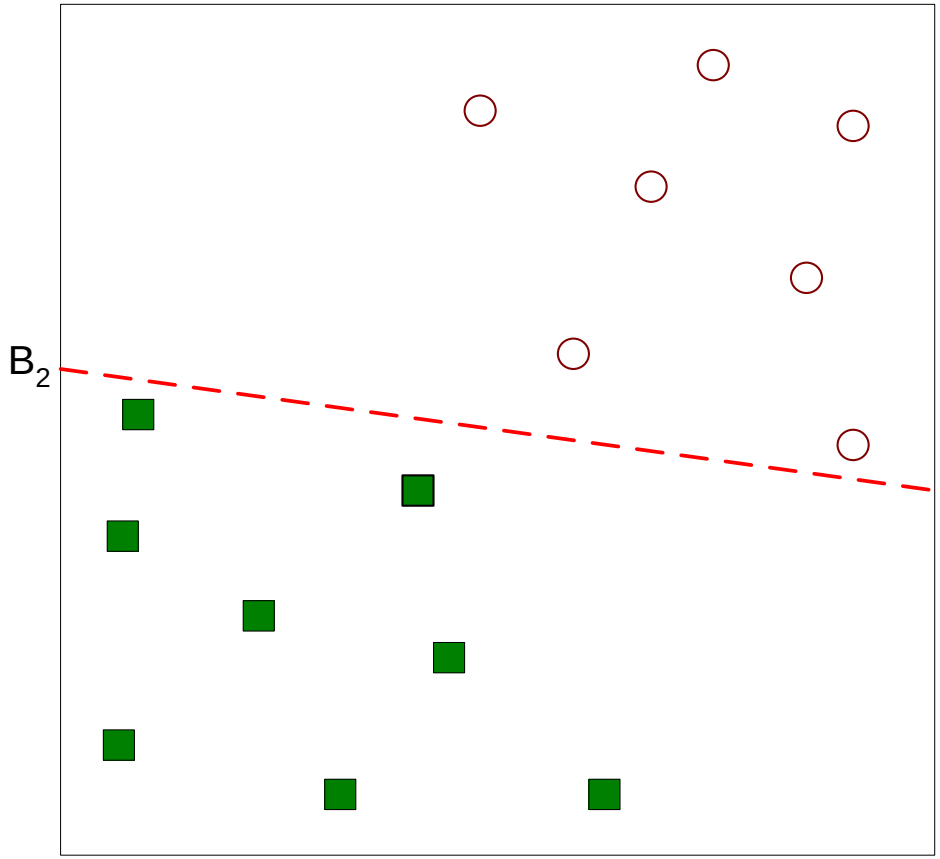
一、最大边缘超平面

给定训练数据集，线性分类器最基本的想法是：在样本空间中寻找一个超平面，将不同类别的样本分开。



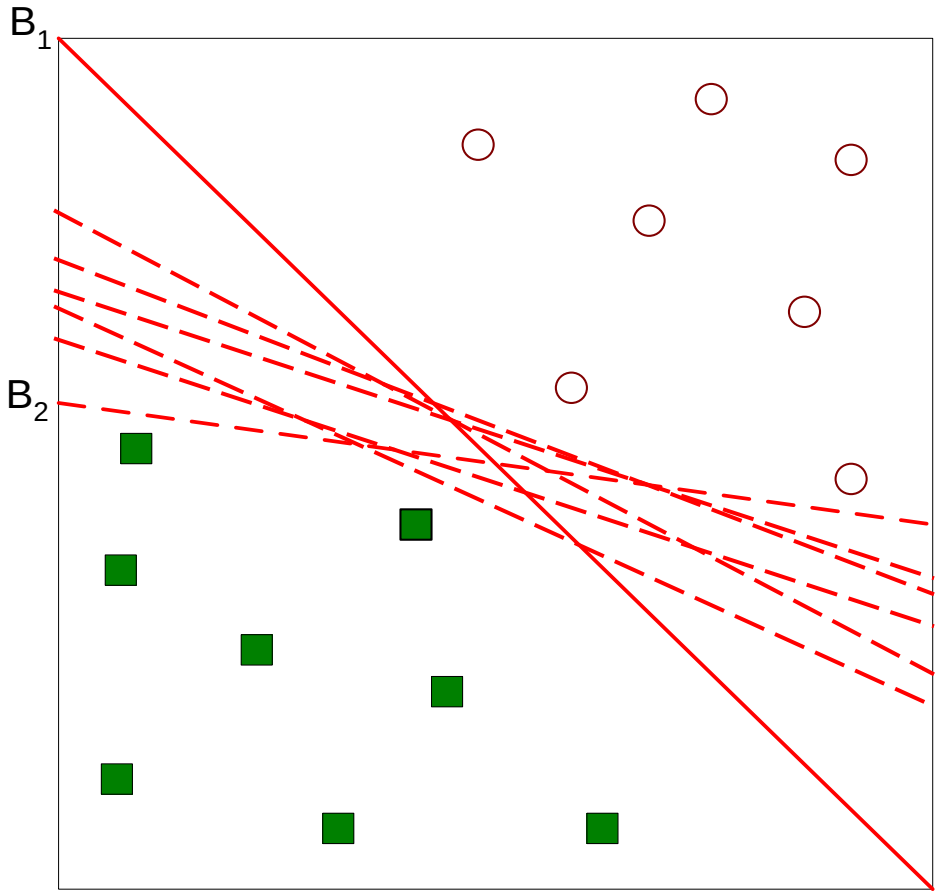
一、最大边缘超平面

给定训练数据集，线性分类器最基本的想法是：在样本空间中寻找一个超平面，将不同类别的样本分开。



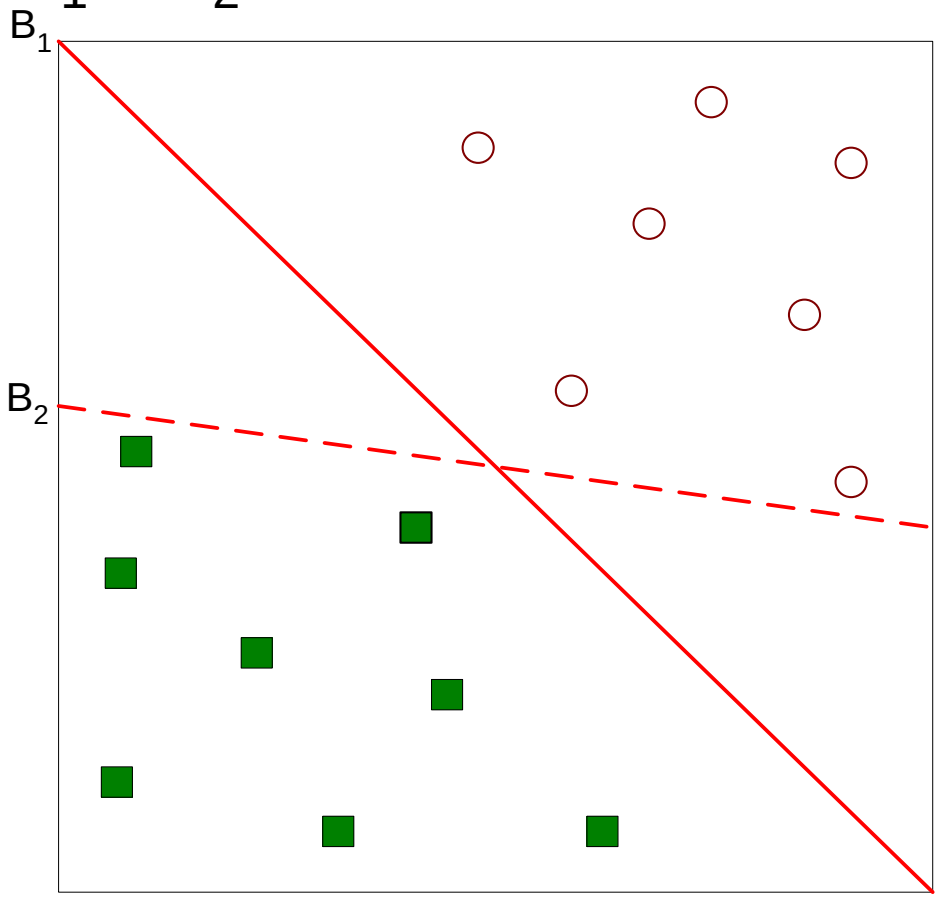
一、最大边缘超平面

其实能将训练样本分开的超平面可能有很多，分类器必须从这些超平面中选择哪个来表示它的决策边界？



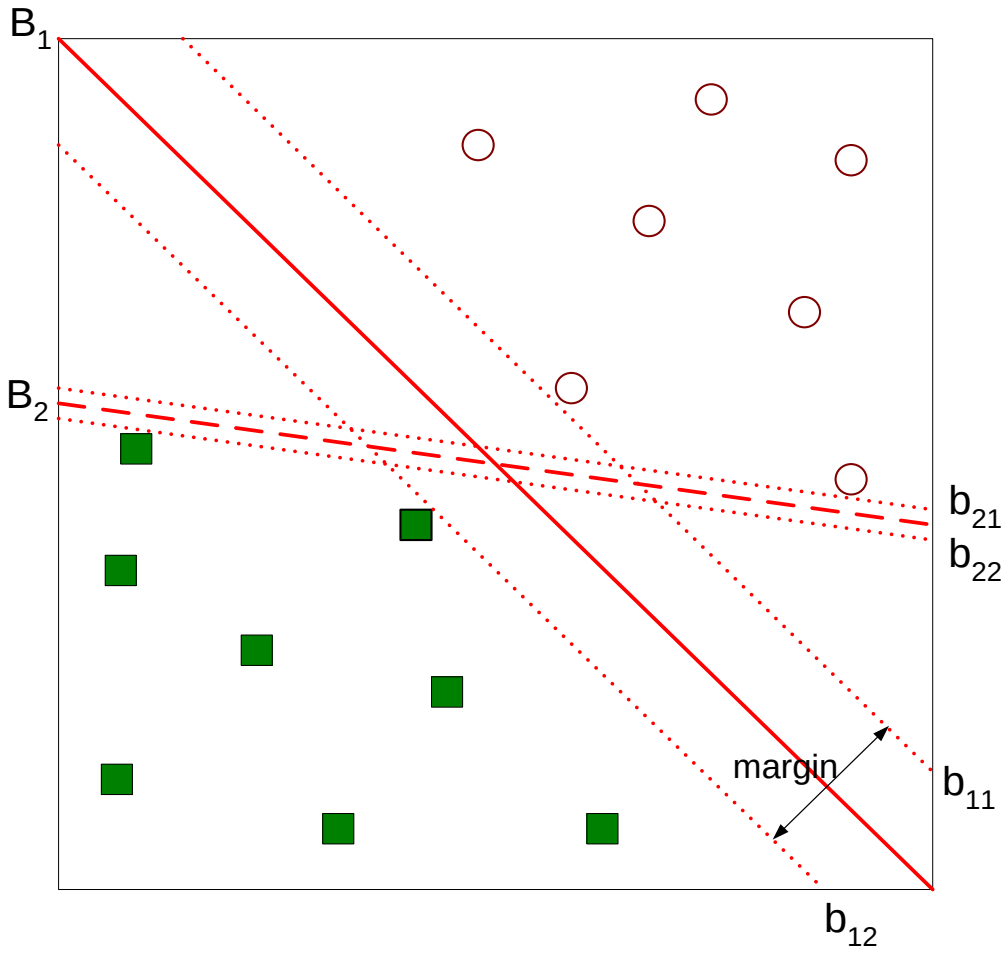
一、最大边缘超平面

为了更好地解释不同的超平面对泛化误差的影响，考虑两个超平面B_1和B_2。



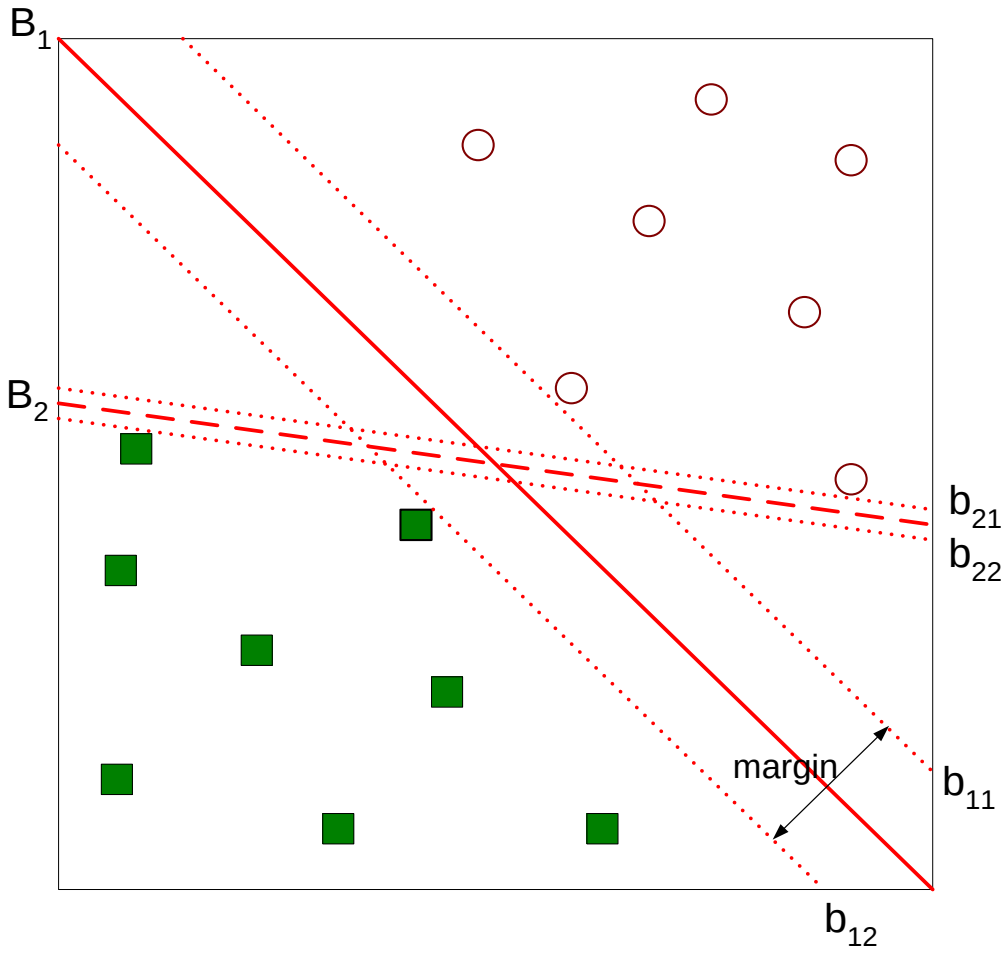
一、最大边缘超平面

每个超平面B_i都对应着一对平行的超平面：b_{i1}和b_{i2}，它们之间的间隔称为超平面的边缘(margin)。



一、最大边缘超平面

图中，超平面B_1的边缘明显大于B_2的边缘。因此，在这个例子中，B_1就是训练样本的最大边缘超平面。



二、线性支持向量机

- 直觉上，决策边界的边缘较小，决策边界任何轻微的扰动都可能对分类结果产生较大的影响。也就是说，具有较大边缘的决策边界比那些具有较小边缘的决策边界具有更好的泛化误差。

- 因此，根据结构风险最小化原理，需要设计最大化决策边界的边缘的线性分类器，以确保最坏情况下的泛化误差最小。线性支持向量机(linear SVM) 就是这样的分类器。

二、线性支持向量机

- 给定训练数据集$D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_n,y_n)\},y\in\{-1,+1\}$

  线性分类器的决策边界可以写成如下线性方程：

$$\boldsymbol{w}^{\top}\boldsymbol{x}+b=0$$

  其中$\boldsymbol{w}=(w_1;w_2;\cdots;w_m)$为法向量，决定了决策边界的方向；$b$为位移量，决定了决策边界与原点之间的距离。显然，决策边界由参数$\boldsymbol{w}$和b确定。

- 假设决策边界能将训练样本正确分类，即对于任意样本点$(\boldsymbol{x}_i,y_i)\in D$：若$y_i=+1$，则有$\boldsymbol{w}^{\top}\boldsymbol{x}_i+b>0$；若$y_i=-1$则有$\boldsymbol{w}^{\top}\boldsymbol{x}_i+b<0$。那么通过调整决策边界的参数$\boldsymbol{w}$和$b$总可以得到：

$$\begin{cases}\boldsymbol{w}^{\top}\boldsymbol{x}_i+b\geq+1, & y_i=+1;\\ \boldsymbol{w}^{\top}\boldsymbol{x}_i+b\leq-1, & y_i=-1.\end{cases}$$

二、线性支持向量机

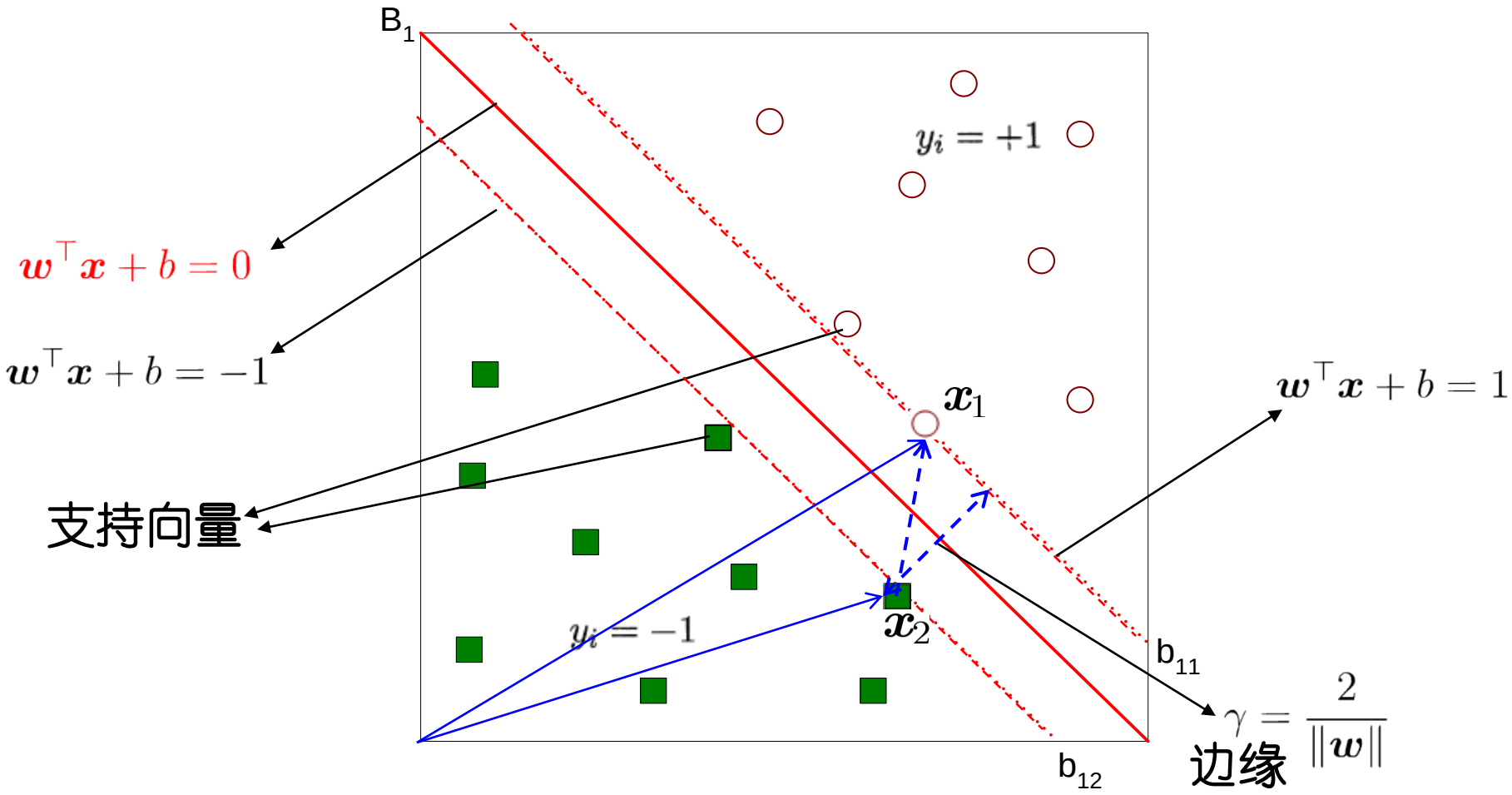
- 距离决策边界最近的训练样本点使得上式中的等号成立，因此被称为“支持向量”(support vector)。
- 两个异类支持向量到决策边界的距离之和被称为决策边界的“边缘”(margin)，刚好等于超平面b_{i1}和b_{i2}之间的间隔γ。
- 为了计算边缘γ，令$\boldsymbol{x}_1$是b_{i1}上的一个样本点，$\boldsymbol{x}_2$是b_{i2}上的一个样本点，将$\boldsymbol{x}_1$和$\boldsymbol{x}_2$分别代入上式，得到$b_{i1}:\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_1+b=1$ $b_{i2}:\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_2+b=-1$

  令两式相减得到：$\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x}_1-\boldsymbol{x}_2)=2$即$\boldsymbol{w}^{\mathrm{T}}\overrightarrow{\boldsymbol{x}_2\boldsymbol{x}_1}=2$

  向量$\overrightarrow{\boldsymbol{x}_2\boldsymbol{x}_1}$在$\boldsymbol{w}$上的投影乘以$\boldsymbol{w}$的模：$\|\boldsymbol{w}\|\times\gamma=2$

  即：$\gamma=\dfrac{2}{\|\boldsymbol{w}\|}$

二、线性支持向量机


B1
$y_i = +1$
$\boldsymbol{w}^\top \boldsymbol{x} + b = 0$
$\boldsymbol{w}^\top \boldsymbol{x} + b = -1$
$\boldsymbol{x}_1$
$\boldsymbol{w}^\top \boldsymbol{x} + b = 1$
支持向量
$y_i = -1$
$\boldsymbol{x}_2$
b11
$\gamma = \frac{2}{\|\boldsymbol{w}\|}$
边缘
b12

二、线性支持向量机

- 因此，线性支持向量机的学习就是要寻找满足约束条件的参数$\boldsymbol{w}$和b, 使得γ最大，即：

$$\arg\max_{\boldsymbol{w},b} \frac{2}{\|\boldsymbol{w}\|}$$

$$s.t. \begin{cases} \boldsymbol{w}^\top \boldsymbol{x}_i + b \geq +1, & y_i = +1; \\ \boldsymbol{w}^\top \boldsymbol{x}_i + b \leq -1, & y_i = -1. \end{cases}$$

等价为

$$\arg\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2$$

$$s.t. \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, i = 1, 2, \cdots, n.$$

二、线性支持向量机

- 由于目标函数是二次的，并且约束条件在参数 $\boldsymbol{w}$和b上是线性的，因此线性支持向量机的学习问题是一个凸二次优化问题，可以直接用现成的优化计算包求解，或者用更高效的拉格朗日乘子法求解。

二、线性支持向量机

- 拉格朗日乘子法
  - ✓ 第一步：引入拉格朗日乘子$\alpha_i \geq 0$得到拉格朗日函数：

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1)$$

  - ✓ 第二步：令$L(\boldsymbol{w}, b, \boldsymbol{\alpha})$对$\boldsymbol{w}$和$b$的偏导为零可得：

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

  - ✓ 第三步：回代拉格朗日函数得到对偶优化问题：

$$L(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x_i}^\top \boldsymbol{x_j}$$

二、线性支持向量机

- 因为对偶优化问题的二次项前面有个负号，而原始线性支持向量机学习的优化问题的二次项前面没有负号，这说明原来要优化的最小化问题已经转化成了最大化对偶优化问题。即：

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\top} \boldsymbol{x}_j$$

- 不过，如果再对上述最大问题稍作变形，就可等价为下面的最小化问题：

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\top} \boldsymbol{x}_j - \sum_{i=1}^{n} \alpha_i$$

二、线性支持向量机

- 再结合拉格朗日函数的约束条件，就可得到原始问题的最终优化问题：

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^{\top}\boldsymbol{x}_j - \sum_{i=1}^{n}\alpha_i$$

$$s.t. \quad \sum_{i=1}^{n}\alpha_i y_i = 0,\ \alpha_i \geq 0,\ i = 1, 2, \cdots, n.$$

二、线性支持向量机

- 解出α后，求出$\boldsymbol{w}$和b即可得到线性SVM的最终模型:$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i^\top \boldsymbol{x} + b$
- 因为线性SVM还需满足不等式约束$y_i f(\boldsymbol{x_i}) \geq 1$，同时$\alpha_i \geq 0$，所以可以把不等式约束变换成等式约束，这种变换得到拉格朗日乘子约束，也称作为Karush-Kuhn-Tucker (KKT) $\alpha_i(y_i f(\boldsymbol{x}_i) - 1) = 0$
- KKT条件表明，除非训练样本满足 $y_i f(\boldsymbol{x}_i) = 1$，否则必有拉格朗日乘子$\alpha_i = 0$。那些$\alpha_i > 0$的训练样本，即$y_i f(\boldsymbol{x}_i) = 1$的训练样本都落在最大边缘的边界$b_{i1}$和$b_{i2}$上，都是支持向量。可见，最终支持向量机模型的参数$\boldsymbol{w}$和$b$仅仅依赖于这些支持向量。

二、线性支持向量机

- 由于对偶优化问题是一个二次规划问题，可使用通用的二次规划算法来求解；然而，该问题的规模正比于训练样本的数目，对于大型数据集，会造成很大的开销。为了避开这个障碍，人们通过利用问题本身的特性，提出了很多高效的算法，比如：SMO序列最小化优化算法 (Sequential Minimal Optimization) 就是其中一个著名的代表，具体算法可参阅[Platt ,1998]：

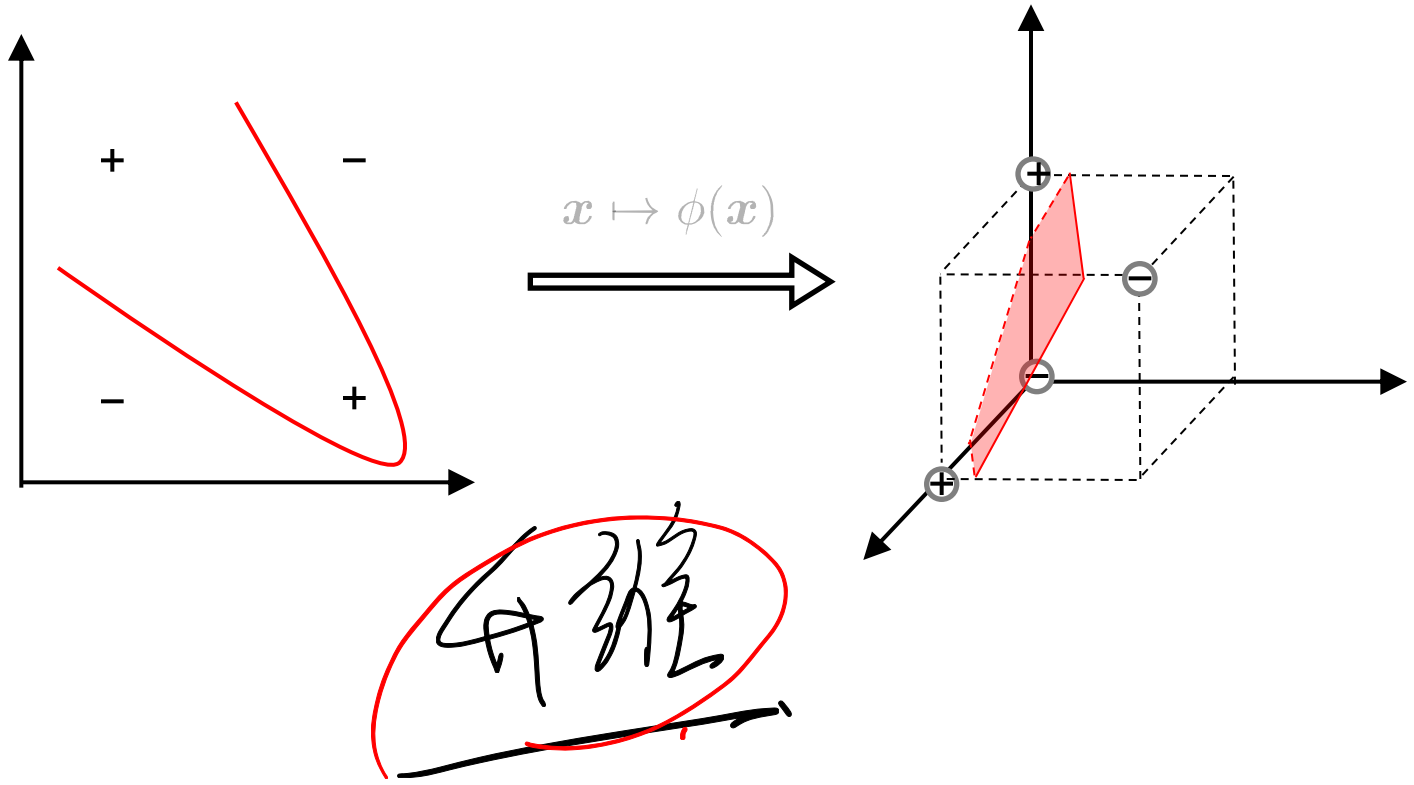
J. Platt: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In B. Schoelkopf and C. Burges and A. Smola, editors, Advances in Kernel Methods - Support Vector Learning, 1998.

三、非线性支持向量机

- 线性SVM假定训练样本是线性可分的，即存在一个线性的决策边界能将所有的训练样本正确分类。
- 然而在实际应用中，在原始的样本空间内也许并不存在这样的决策边界。
- 对于这样的问题，可将样本从原始空间映射到一个更高维的特征空间，使得样本在映射后的特征空间内线性可分。例如在下图中，如果将原始的二维空间映射到一个合适的三维空间，就能找到一个合适的划分超平面。幸运的是，如果原始空间是有限维，即属性数目有限，那么一定存在一个更高维的特征空间使得样本线性可分。

三、非线性支持向量机

异或问题的非线性映射



三、非线性支持向量机

- 设样本$\boldsymbol{x}$映射后的向量为$\phi(\boldsymbol{x})$, 决策边界为$f(\boldsymbol{x})=\boldsymbol{w}^{\top}\phi(\boldsymbol{x})+b$

原始问题

$$\min_{\boldsymbol{w},b}\frac{1}{2}\|\boldsymbol{w}\|^2$$

$$s.t.\quad y_i(\boldsymbol{w}^{\top}\phi(\boldsymbol{x}_i)+b)\geq 1,\ i=1,2,\cdots,n.$$

对偶问题

$$\min_{\boldsymbol{\alpha}}\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\phi(\boldsymbol{x}_i)^{\top}\phi(\boldsymbol{x}_j)-\sum_{i=1}^{n}\alpha_i$$

$$s.t.\quad \sum_{i=1}^{n}\alpha_i y_i=0,\ \alpha_i\geq 0,\ i=1,2,\cdots,n.$$

预测模型

$$f(\boldsymbol{x})=\boldsymbol{w}^{\top}\phi(\boldsymbol{x})+b=\sum_{i=1}^{n}\alpha_i y_i\phi(\boldsymbol{x}_i)^{\top}\phi(\boldsymbol{x})+b$$

三、非线性支持向量机

- 求解上面的优化问题，涉及到计算$\phi(\boldsymbol{x}_i)^\top\phi(\boldsymbol{x}_j)$，这是样本$\boldsymbol{x}_i$与$\boldsymbol{x}_j$映射到特征空间之后的内积。由于特征空间维数可能很高，甚至可能是无穷维，因此直接计算$\phi(\boldsymbol{x}_i)^\top\phi(\boldsymbol{x}_j)$通常是非常困难的。为了避开这个障碍，一个基本的想法是：不显式地设计映射$\phi(\cdot)$，而是设计一个核函数：

$$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\phi(\boldsymbol{x}_i)^\top\phi(\boldsymbol{x}_j)$$

- $\boldsymbol{x}_i$和$\boldsymbol{x}_j$在高维特征空间的内积等于它们在原始样本空间中通过核函数$k(\cdot,\cdot)$计算的结果。有了这样的核函数，我们就不必直接去计算高维甚至无穷维特征空间中的内积了。

三、非线性支持向量机

- 那么，合适的核函数是否一定存在呢？什么样的函数才能做核函数呢？
- 根据Mercer定理，只要一个对称函数所对应的核矩阵半正定，那么它就能作为核函数来使用。
- 常用核函数包括：

名称	表达式	参数
线性核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j$	
多项式核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^\top \boldsymbol{x}_j)^d$	$d \geq 1$为多项式的次数
高斯核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\delta^2}\right)$	$\delta > 0$为高斯核的带宽(width)
拉普拉斯核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\delta}\right)$	$\delta > 0$
Sigmoid核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\beta \boldsymbol{x}_i^\top \boldsymbol{x}_j + \theta)$	tanh为双曲正切函数, $\beta > 0, \theta < 0$

- 于是，核函数的选择成为非线性SVM的最大变数，若核函数选择不合适，就意味着将样本映射到了一个不合适的特征空间，从而很可能导致非线性SVM性能不佳。