模式识别

课程团队：谢凤英 崔林艳 张浩鹏
邹征夏 李洪珏 李家军
单位：宇航学院

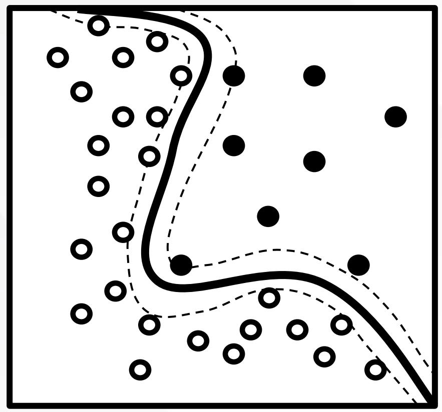
第四章 非线性分类器

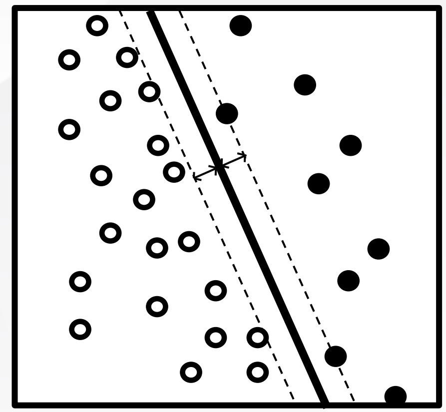
CONTENTS PAGE

非线性分类器

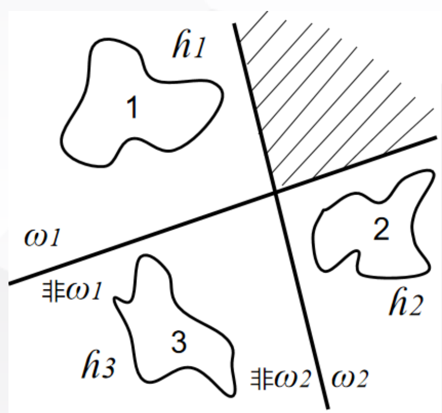
在很多情况下，类别之间的分类边界并不是线性的，需要用更复杂的非线性函数来描述分类。如

两类问题

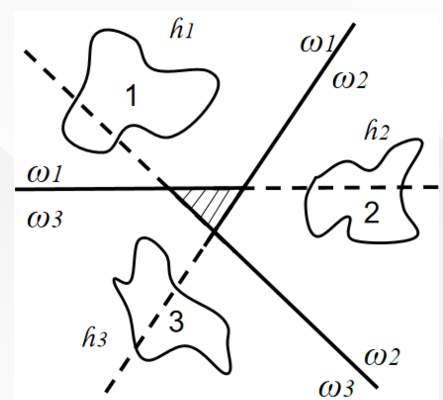




多类问题



“类与类的非”分类
需c-1个线性分类器

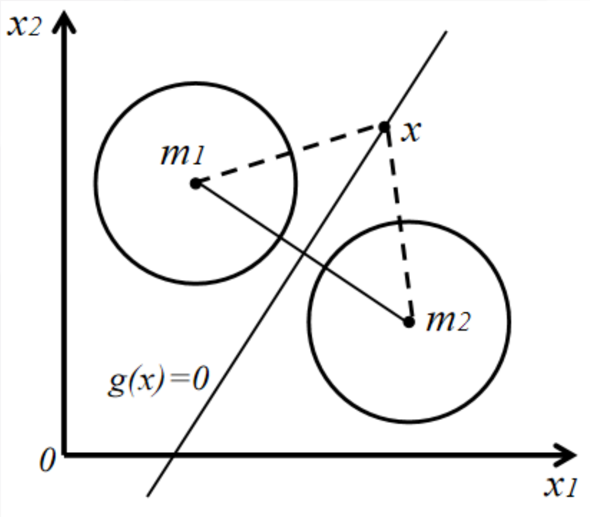


两两线性分类
需c*(c-1)/2个分类器

4.1 最小距离分类器

1、回顾两类单线性分类器

- 垂直平分 / 最小距离分类器
- 基于两类样本均值点作垂直平分线



2、最小距离分类器形式

- 判别函数：

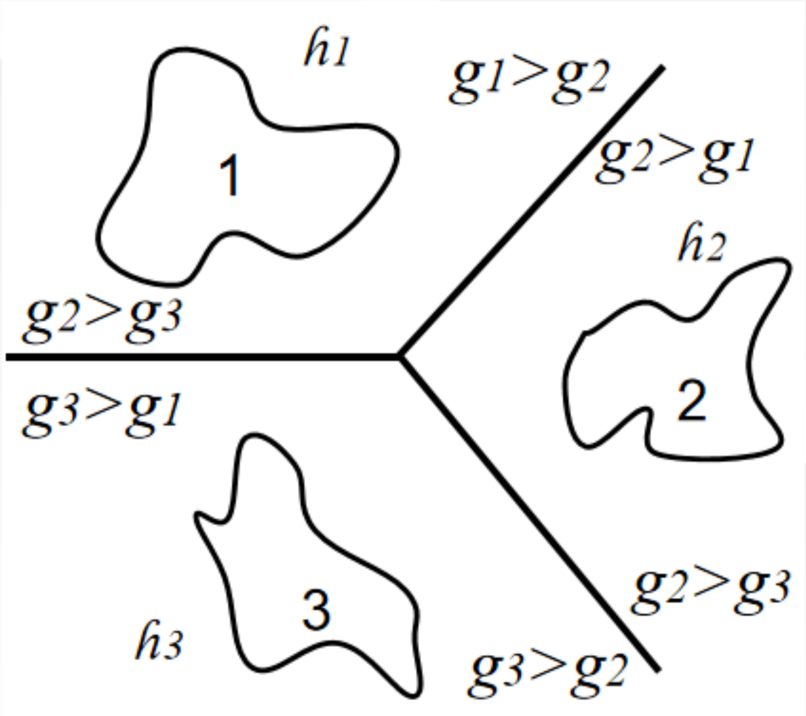
$$G_1(\mathbf{x}) = d_1(\mathbf{x}) = \|\mathbf{x} - m_1\|$$
$$G_2(\mathbf{x}) = d_2(\mathbf{x}) = \|\mathbf{x} - m_2\|$$

- 决策规则

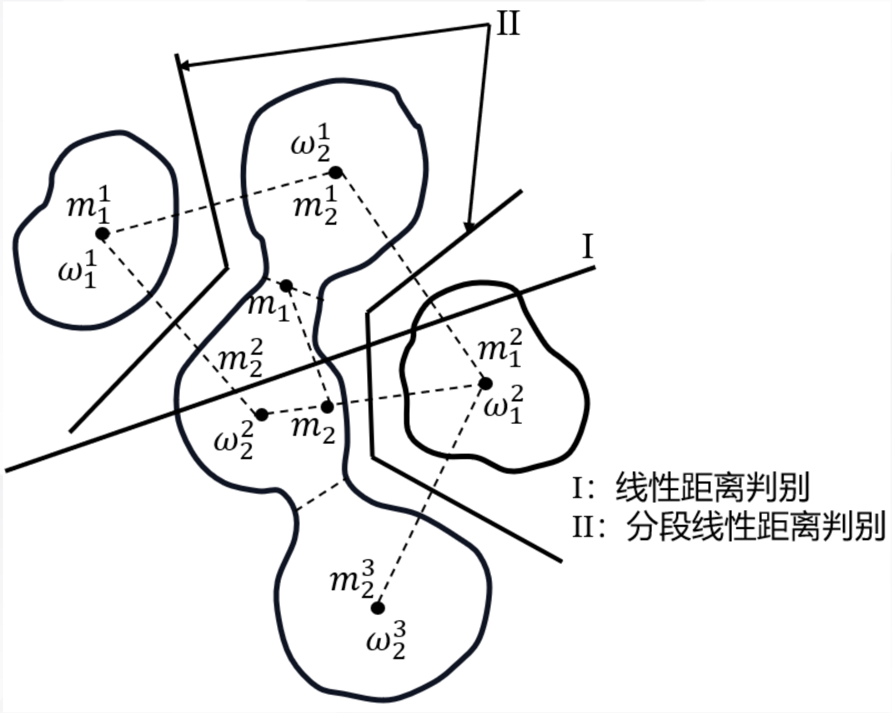
对于未知样本x，若$d_1(\mathbf{x}) < d_2(\mathbf{x})$，则$\mathbf{x}$决策为$\omega_1$类；若$d_1(\mathrm{x}) > d_2(\mathrm{x})$，则$\mathbf{x}$决策为$\omega_2$类。

4.1 最小距离分类器

3、使用最小距离分类器解决多类问题



解决C类单峰问题



解决两类多峰问题

4.1 最小距离分类器

举例： 已知各类及其子类， 设计分段最小距离分类器

1）先求各子类均值：

m_{ij}（ω_i类的第j子类）

2）定义各子类判别函数：

$$\mathrm{G}_i(\mathrm{x}) = min\left\|\mathrm{x} - m_{ij}\right\|$$

3）决策规则：

对于未知样本**x**，若$\mathrm{G}_k(\mathbf{x}) = min\mathrm{G}_i(\mathbf{x})$，则x决策为$\omega_k$

4.1 最小距离分类器

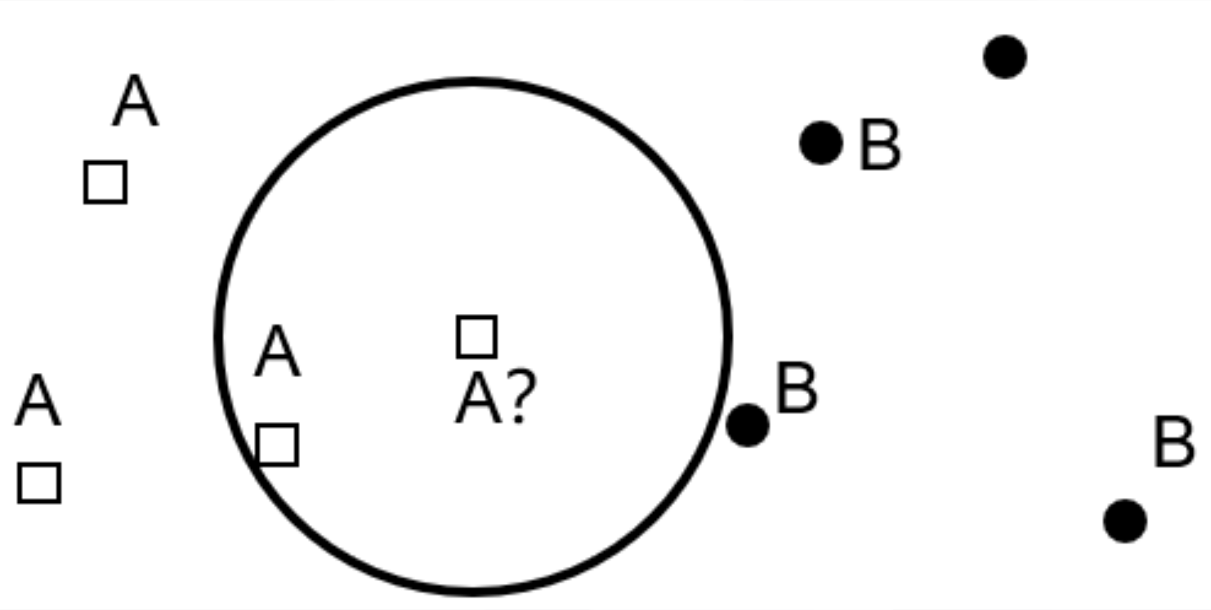
4、最小距离分类器特点

- 解决两类多峰或多类问题的分段线性分类器
- 可以解决几乎所有分类问题，但要已知各类子类
- 概念直观简单，但未经优化
- 分类器设计简单容易
- （无重叠区或空白区）

4.2 近邻法分类器

1、最近邻法

基本思想：对于一个新样本，把它逐一与已知样本进行比较，找出距离新样本最近的已知样本，并以该样本的类别作为新样本的类别。



1、最近邻法

假定有 c 个类别的模式识别问题，每类有标明类别的样本 N_i 个，$i=1,\cdots,\mathrm{c}$。定义两个样本之间的距离度量 $\delta(\mathbf{x}_i,\mathbf{x}_j)$，比如可以采用欧式距离 $\delta(\mathbf{x}_i,\mathbf{x}_j)=\|\mathbf{x}_i-\mathbf{x}_j\|$，则可以规定类 ω_i 的判别函数为：

$$g_i(\mathbf{x})=\min_k \delta\left(\mathbf{x},\mathbf{x}_i^k\right), k=1,\cdots,N_i$$

其中，i 表示 ω_i 类，x_i^k 表示 ω_i 类 N_i 个样本中的第 k 个。

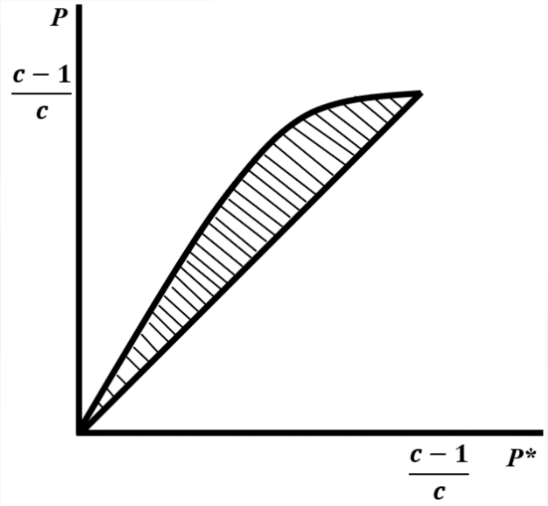
决策规则为：

若 $g_j(\mathbf{x})=\min_i g_i(\mathbf{x}), i=1,\cdots,c$，则决策 $\mathbf{x}\in\omega_j$。

4.2 近邻法分类器

1、最近邻法

研究表明，在已知样本数量足够的情况下，假设最近邻决策的错误率为p，类别数为c，则有



$$p^* \le p \le p^*\left(2 - \frac{c}{c-1}p^*\right)$$

其中，p^*为贝叶斯错误率（即理论最优错误率）。

结论：最近邻法的渐进错误率最坏不会超过两倍的贝叶斯错误率，而最好则有可能接近或达到贝叶斯错误率。

问题：在很多情况下，把决策建立在一个最近的样本上有一定风险，尤其是当数据分布复杂或数据中噪声严重时。

2、K-近邻法（k-Nearest Neighbor）

➢ k-近邻算法的原理

在N个已知样本中，找出未知样本$\mathbf{x}$的k个近邻。设这k个样本中，来自ω_1类的样本有k_1个，来自ω_2类的样本有k_2个，以此类推，来自ω_c类的样本有k_c个，则我们可以定义判别函数为：

$$g_i(\mathbf{x}) = k_i, i = 1, \cdots, c$$

决策规则为：若$g_j(\mathbf{x}) = \max\limits_i k_i$，则决策$\mathrm{x} \in \omega_j$。

2、K-近邻法

➢ k-近邻算法的一般步骤

1）首先确定k值（就是指k-近邻方法中k的大小，代表对于一个待分类的数据点，要寻找它的k个邻居）。

2）根据事先确定的距离度量公式（如：欧氏距离），得出待分类数据点和所有已知类别的样本点中，距离最近k个样本。

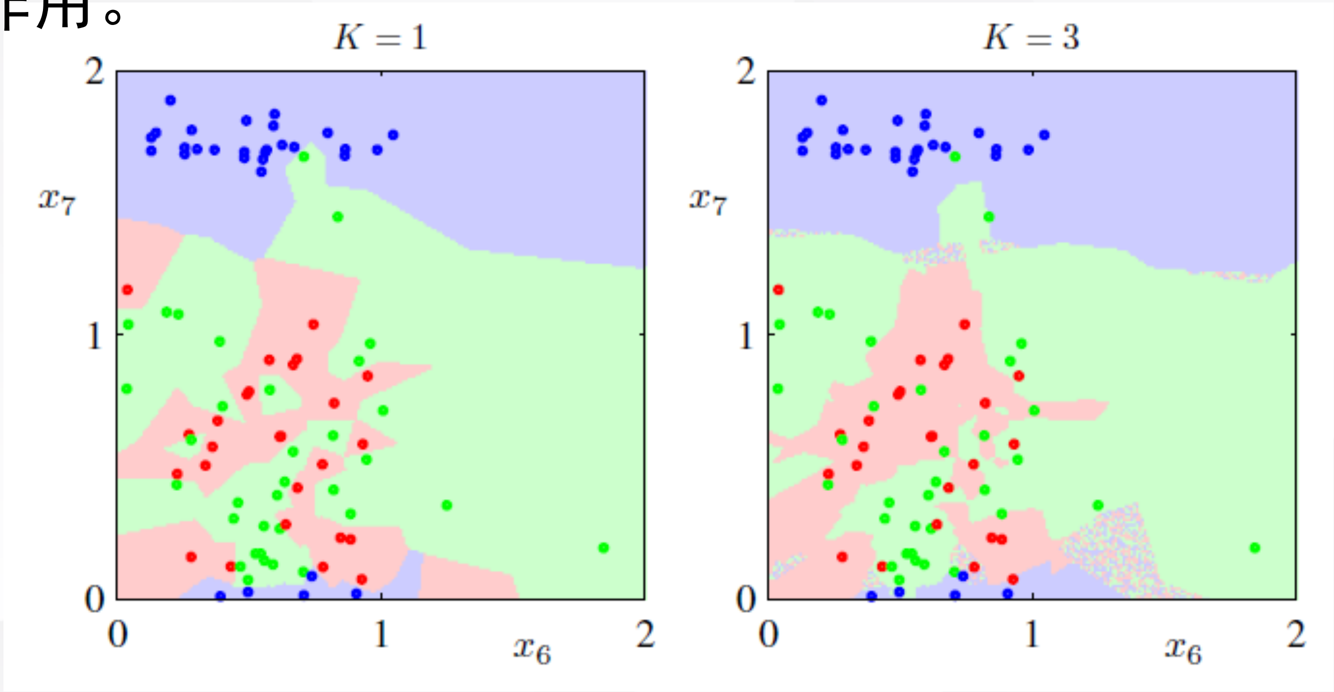
3）统计这k个样本点中各个类别的数量，并且判定该待分类数据点属于类别数量最高的那一类。

4.2 近邻法分类器

2、 K-近邻法

➢ 最近邻和k-近邻算法的对比实例

最近邻法对孤立点敏感，k-近邻对孤立点不敏感，可以起到平滑作用。



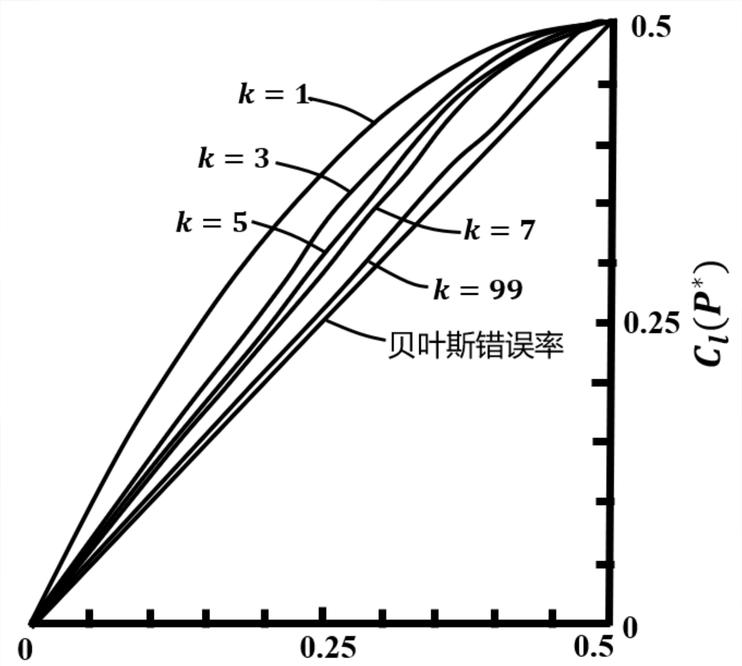
最近邻　　　　K-近邻（K=3）

4.2 近邻法分类器

2、 K-近邻法

➢ 近邻法错误率上下界

当 $k=1$ 时，k-近邻法就是最近邻算法。而随着 k 的增加，k-近邻法的渐进错误率逐渐降低，当趋近无穷大时，接近贝叶斯错误率。



3、近邻法的三要素

（1）k值的选择

- 选择的值越小，模型复杂度越高，容易发生过拟合。极端情况$k=1$，如果恰好遇到噪声，就会完全错误。
- 随着 k 值增大，模型泛化能力也增大，但丢失的信息也增多。k值的增大就意味着整体的模型变得简单。
- 设想$k=N$，则任意新输入样例的分类就等于训练样例中样本数最多的分类，此时无论输入样本是什么，都只是简单的预测它属于在训练样本中最多的类。

3、近邻法的三要素

（2）距离量度的方式

用L_p系列函数作为量度方法，设：

$$\mathbf{x}_i, \mathbf{x}_j \in R^n$$

$$\mathbf{x}_i = (x_i^1, x_i^2 \cdots, x_i^n), \mathbf{x}_j = (x_j^1, x_j^2 \cdots, x_j^n)$$

$\mathbf{x}_i, \mathbf{x}_j$的距离$L_p$定义为

$$L_p(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{l=1}^{n} \left|x_i^l - x_j^l\right|^p\right)^{\frac{1}{p}}, p \geq 1$$

3、近邻法的三要素

（2）距离量度的方式

当$p=2$时，是欧式距离(Euclidean distance)：

$$L_2(\mathbf{x}_i,\mathbf{x}_j)=\left(\sum_{l=1}^{n}\left|x_i^l-x_j^l\right|^2\right)^{\frac{1}{2}}$$

当$p=1$时，是曼哈顿距离(Manhattan distance)：

$$L_1(\mathbf{x}_i,\mathbf{x}_j)=\sum_{l=1}^{n}\left|x_i^l-x_j^l\right|$$

当$p=\infty$时，是各个坐标距离的最大值:

$$L_\infty(\mathbf{x}_i,\mathbf{x}_j)=\max_{l}\left|x_i^l-x_j^l\right|$$

3、近邻法的三要素

（3）分类决策规则

假设现在已经找到了k个近邻的点。最直观的确定分类的方法就是“多数表决”，即把新样本分到k个点所属分类最多的类。

4.2 近邻法分类器

4、近邻法的特点

- 可以解决几乎所有分类问题
- 概念直观简单未经优化，但错误率并不高
- 分类器设计容易
- 运算量大，需要设计快速算法

4.2 近邻法分类器

5、kd树

- k-近邻算法是机器学习中最简单的算法之一，如果训练样本过大，则传统的遍历全样本寻找k近邻的方式将导致性能的急剧下降。为了优化效率，不同的训练数据存储结构被纳入到实现方式之中。
- kd树（K-Dimension Tree）是一种对k维空间中的实例点进行存储以便对其进行快速检索的树形数据结构。
- kd树是二叉树，表示对k维空间的一个划分（partition)。

https://zhuanlan.zhihu.com/p/53826008

4.2 近邻法分类器

5、kd树

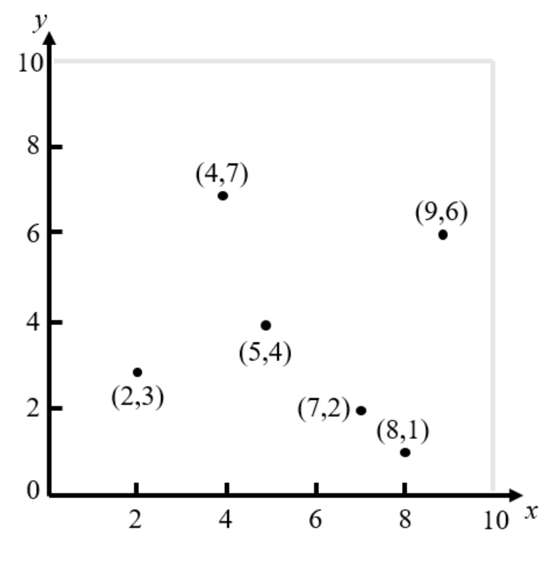
➢ kd树的构造

构造kd树相当于不断地用垂直于坐标轴的超平面将k维空间切分，构成一系列的k维超矩形区域。kd树的每个节点对应于一个k维超矩形区域。

通常，依次选择坐标轴对空间划分，选择训练样本点在选定坐标轴上的中位数为切分点，这样得到的kd树是平衡的。

例如：

$$T=\{(2,3)^{\mathrm{T}},(5,4)^{\mathrm{T}},(9,6)^{\mathrm{T}},(4,7)^{\mathrm{T}},(8,1)^{\mathrm{T}},(7,2)^{\mathrm{T}}\}$$



4.2 近邻法分类器

5、kd树

➢ kd树的构造

例如：$T=\left\{(2,3)^{\mathrm{T}},(5,4)^{\mathrm{T}},(9,6)^{\mathrm{T}},(4,7)^{\mathrm{T}},(8,1)^{\mathrm{T}},(7,2)^{\mathrm{T}}\right\}$

切分域选择：1）K个维度，每个维度完成一次切分，则完成了一个轮次的切分。2）每次切分，计算待切分空间内的点未被切分维度上的方差，找方差最大的维度作为此次的切分域。方差较大，表明在该维度上的点的分散度较高，按该维度切分分辨率比较高。

计算第一次切分时两个维度上的方差，则选择X轴切分

$$\bar{x}=(2+5+9+4+8+7)/6=5.83, D(x)=\sum_{k=1}^{6}(x_k-\bar{x})^2=34.83$$

$$\bar{y}=(3+4+6+7+1+2)/6=3.83, D(y)=\sum_{k=1}^{6}(y_k-\bar{y})^2=26.71$$

4.2 近邻法分类器

5、kd树

➢ kd树的构造

例如：$T=\{(2,3)^{\mathrm{T}},(5,4)^{\mathrm{T}},(9,6)^{\mathrm{T}},(4,7)^{\mathrm{T}},(8,1)^{\mathrm{T}},(7,2)^{\mathrm{T}}\}$

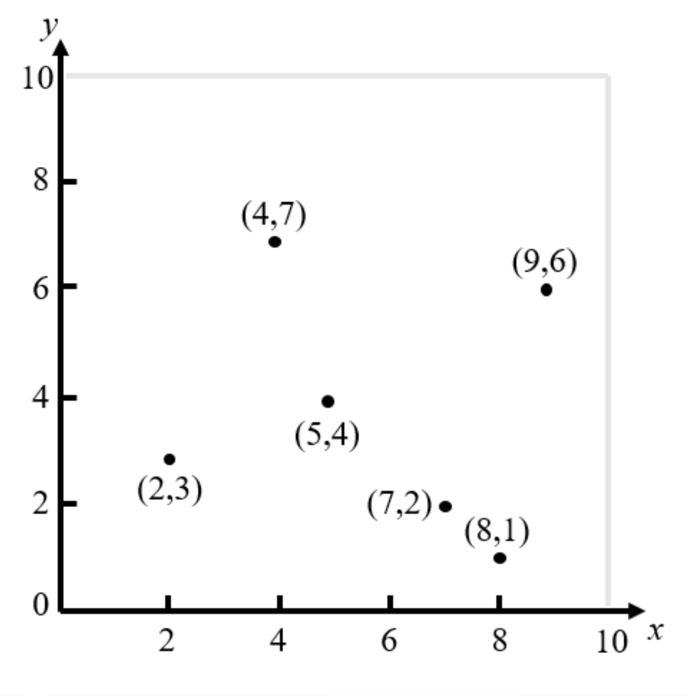
计算第一次切分时两个维度上的方差，则选择X轴切分

$$\bar{x}=(2+5+9+4+8+7)/6=5.83$$

$$D(x)=\sum_{k=1}^{6}(x_k-\bar{x})^2=34.83$$

$$\bar{y}=(3+4+6+7+1+2)/6=3.83$$

$$D(y)=\sum_{k=1}^{6}(y_k-\bar{y})^2=26.71$$



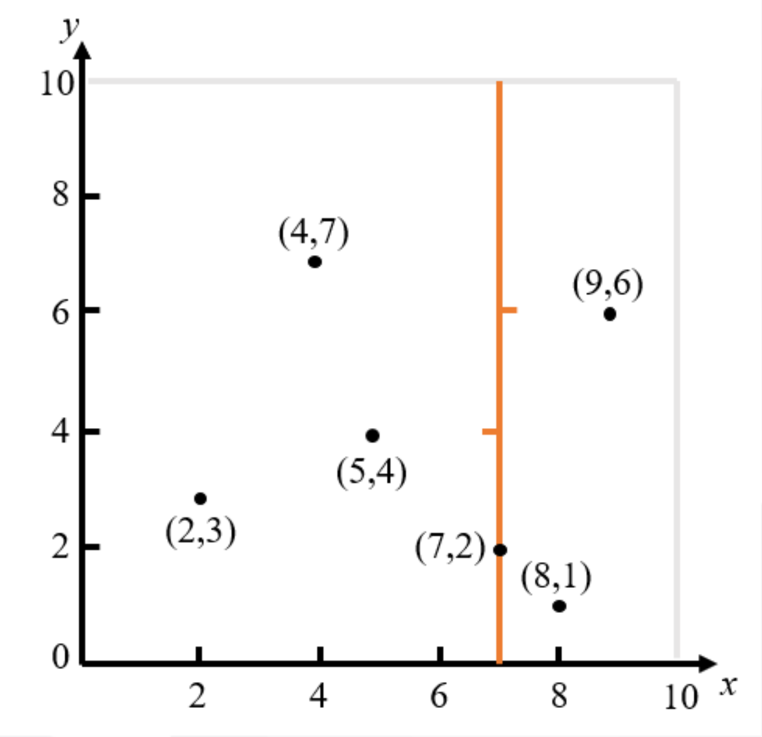
5、kd树

➢ kd树的构造

例如：$T=\{(2,3)^{\mathrm{T}},(5,4)^{\mathrm{T}},(9,6)^{\mathrm{T}},(4,7)^{\mathrm{T}},(8,1)^{\mathrm{T}},(7,2)^{\mathrm{T}}\}$

切分点选择： 取待切分平面上的所有数据点的中位数作为切分点，也有计算平均值作为切分点。

我们计算中位数，确定切分域和切分点后，可得第一次切分。

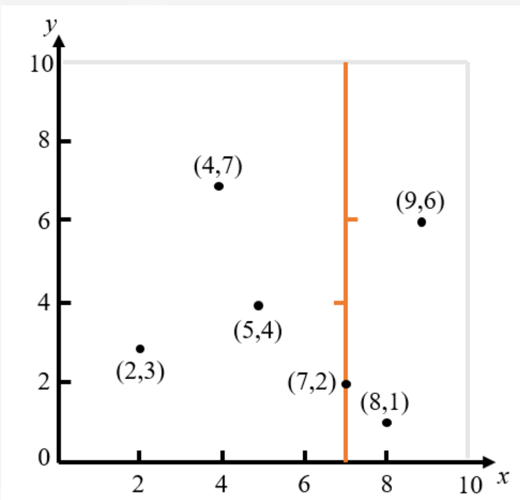


4.2 近邻法分类器

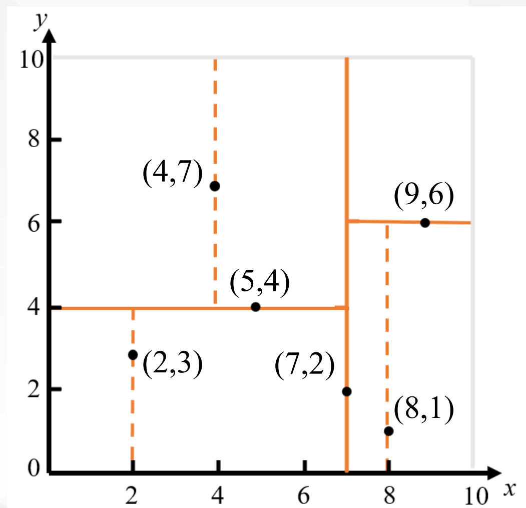
5、kd树

➢ kd树的构造

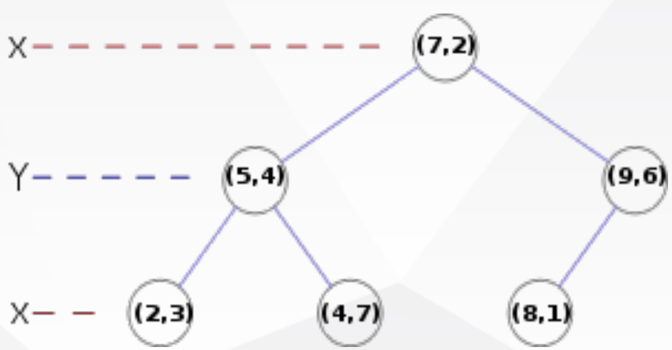
例如：$T=\left\{(2,3)^{\mathrm{T}},(5,4)^{\mathrm{T}},(9,6)^{\mathrm{T}},(4,7)^{\mathrm{T}},(8,1)^{\mathrm{T}},(7,2)^{\mathrm{T}}\right\}$



第一轮第一次，中位数7将空间氛围左右两个子矩形



第一轮第二次切分以及第二轮切分



形成的kd树

4.2 近邻法分类器

5、kd树

➢ 查找

以最近邻为例，包含以下两个步骤：

1）寻找近似点。按维度切分顺序进行搜索，寻找最近邻的叶子节点作为目标数据的近似最近点。

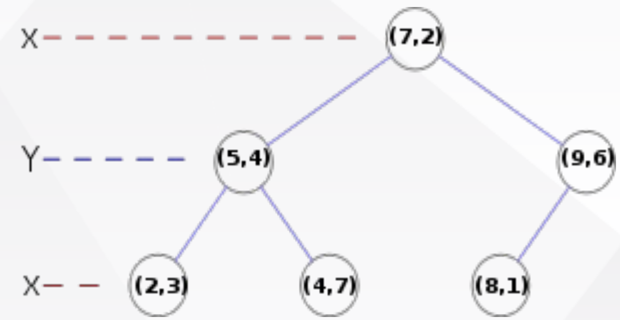
2）回溯。以目标数据和最近邻的近似点的距离沿树根部进行回溯和迭代。

4.2 近邻法分类器

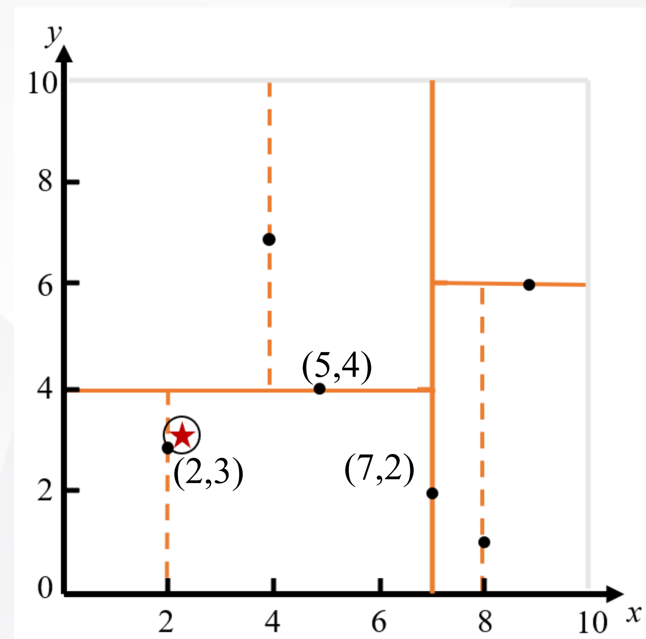
5、kd树

➢ 查找

例如查找点（2.1, 3.1）



计算近似最近点。与（7, 2）比较，2.1小于7，向左搜索；下一个点为(5, 4)，3.1小于4，向左搜索，最后定位（2, 3）是近似最近点，距离为0.141。将（2, 3）到（2.1, 3.1）的距离为半径，以（2, 3）为圆心作圆。上述路径是(7, 2)->(5, 4)->(2, 3)。

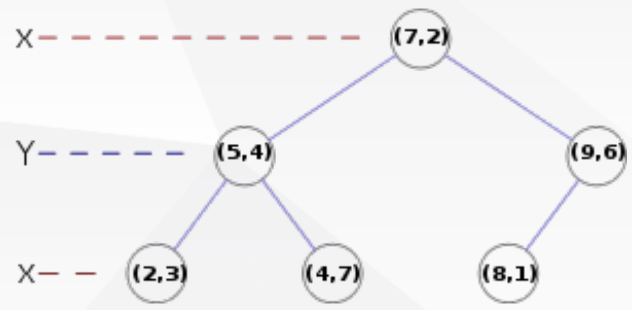


回溯。先计算该点与（5, 4）的距离，大于0.141，被(5, 4) 切分的另一个子平面与（2, 3）点为圆心的圆无交集。再回溯（7, 2）点，与其右子平面无交集，回溯结束，确认最近邻点为（2, 3）。

4.2 近邻法分类器

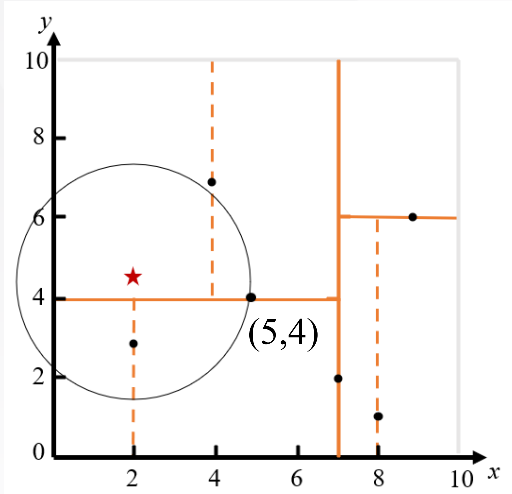
5、kd树

➢ 查找

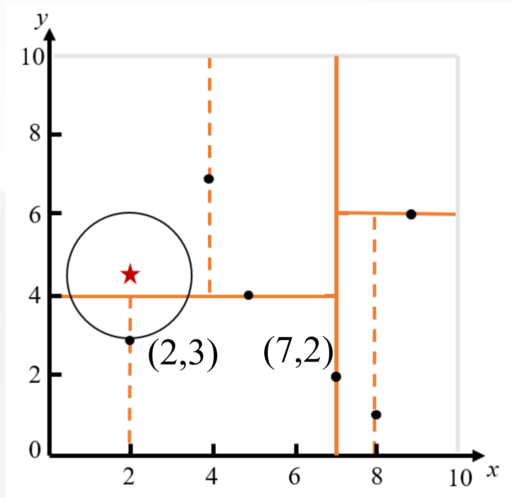


再如查找点（2, 4.5）

1）查找路径为(7, 2)->(5, 4)->(4, 7)，近似最近点落在叶子节点（4, 7），距离为3.20，作圆。



2）回溯。其（5, 4）与目标点的距离为3.04，小于3.20，则更新（5, 4）为最近近似点，以3.04做圆；此圆与（5, 4）所切分的上下两个平面相交，需要检查（5, 4）的另外一个子树的叶子节点（2, 3）。（2, 3）的距离为1.5，小于3.04，更新（2, 3）为近似最近点；最后回溯至（7, 2），确认与（7, 2）切分的右子平面无关；回溯结束，（2, 3）为其最近点。



6、剪辑法

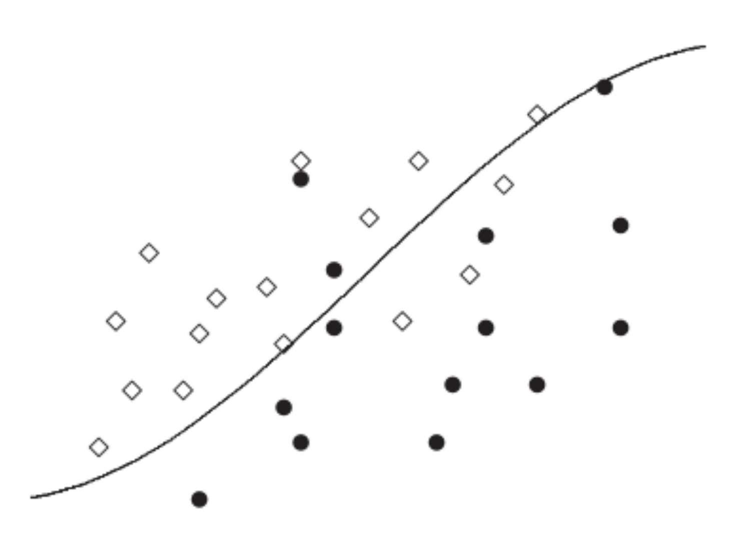
K近邻算法进行分类时，对于一个待分类的样本，需要计算其与训练集中所有样本的距离，并选择距离最小的前k个来进行分类决策。随着训练样本数的增大，K近邻算法的计算成本急剧增大。

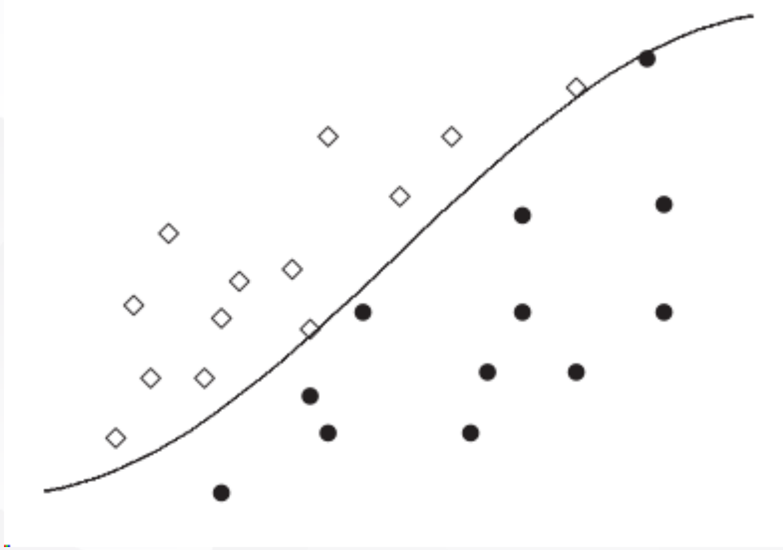
有两种减少训练集样本数的方法：**剪辑方法**和**压缩方法**。

- **剪辑法：**设法将交界区的已知样本去掉，决策时就不会受到这些样本的影响，使近邻法的决策面更接近最优分类面。
- **压缩法：**设法找出各类样本中最有利于用来与其他类区分的代表性样本，进而把很多训练样本都去掉，从而简化决策过程中的计算。

6、剪辑法

- 剪辑方法通过对训练集的处理达到去除被错分的训练样本的目的。
- 基本的剪辑方法如下：给定训练集R和分类标准η，设S是被分类规则错分的样本集，将这些样本从训练集中除去。重复这个过程直到满足停止规则。





4.4 支持向量机

- 支持向量机(Suppot Vector Machines，SVM) 被提出于1964年，20世纪90年代后得到快速发展，并衍生出一系列改进和扩展算法。
- 在解决小样本情况下的机器学习问题和高维、非线性问题中表现出较为优异的效果。
- SVM是基于线性可分的最优分类面提出的，最优分类面的定义，保证了在样本一定的情况下，两类样本间的距离最大。

参考书：南京大学周志华教授

https://blog.csdn.net/u014472643/article/details/79612204 对偶

https://blog.csdn.net/sinat_20177327/article/details/79729551 松弛约束条件

https://blog.csdn.net/yjn03151111/article/details/46746839 对软间隔问题的解释

https://blog.csdn.net/qq_35992440/article/details/80987664 拉格朗日乘子法与对偶算法在svm中的应用

。。。。。。

4.4 支持向量机

1、寻找最优分类面

2、用拉格朗日方程求解对偶问题

3、核函数

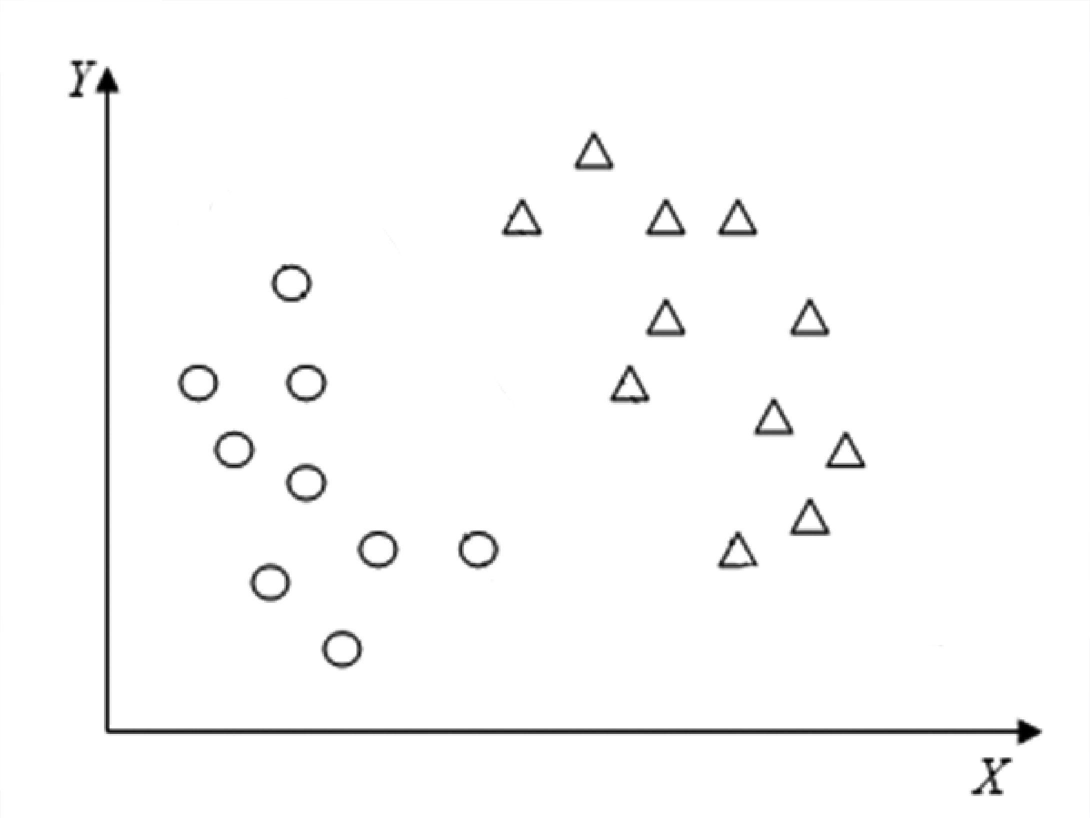
4、核技巧

5、软间隔

6、SVM解决多分类问题

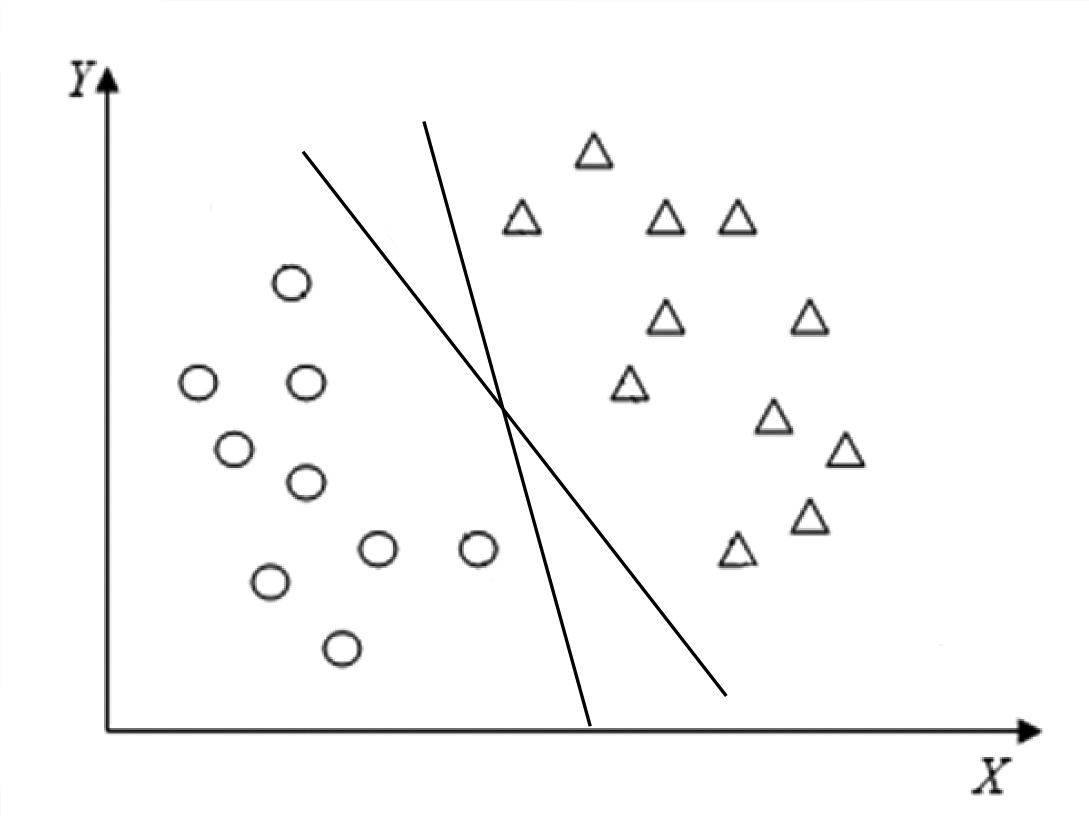
4.4 支持向量机

1、寻找最优分类面



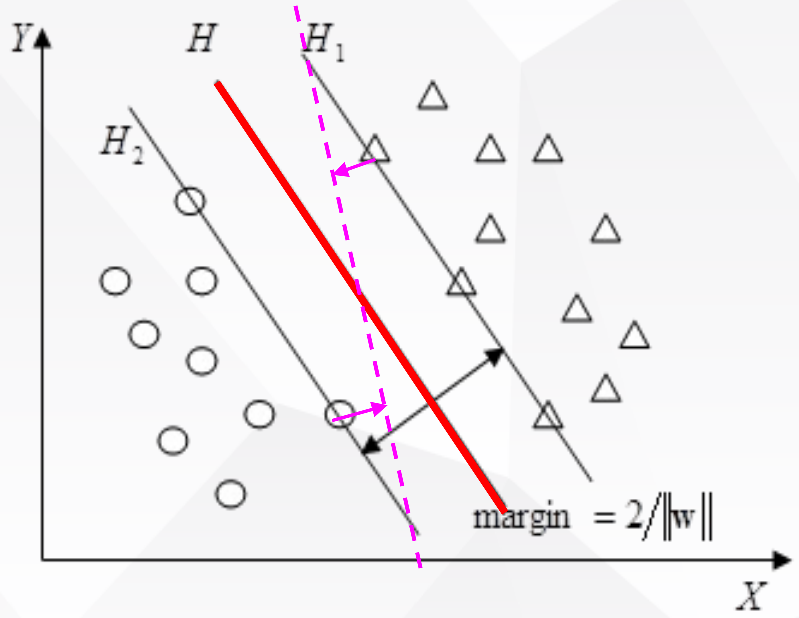
4.4 支持向量机

1、寻找最优分类面



4.4 支持向量机

1、寻找最优分类面



4.4 支持向量机

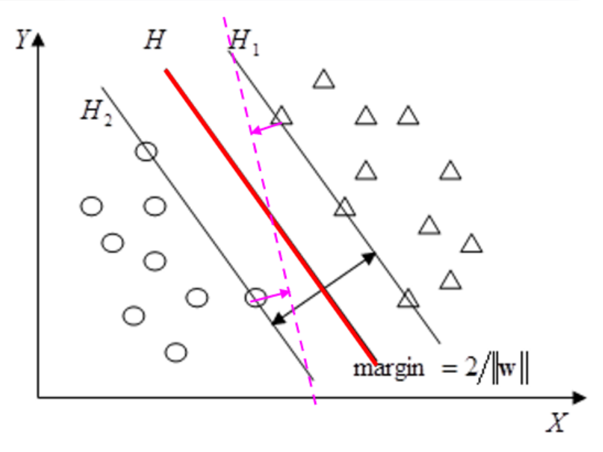
1、寻找最优分类面

一个样本有d个特征，用$\mathbf{x}=(x_1,x_2,\cdots,x_d)$表示。给定训练集 $D=\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\cdots,(\mathbf{x}_n,y_n)\}$，$y_i\in\{-1,+1\}$,$\mathbf{x}_i\in \mathrm{R}^d$

在d维样本空间中，划分超平面可通过如下线性方程来描述：

$$f(\mathrm{x})=\mathbf{w}^T\mathbf{x}+b=0$$

其中$\mathbf{w}=(w_1,w_2,\cdots,w_d)$为法向量，决定了超平面的方向，$b$为位移项，决定了超平面与原点之前的距离。



样本空间中任意点$\mathbf{x}=(x_1,x_2,\cdots,x_d)$到超平面$(\mathbf{w},b)$的距离为：

$$\mathrm{r}=\frac{|\mathbf{w}^T\mathbf{x}+b|}{\|\mathbf{w}\|}$$

4.4 支持向量机

1、寻找最优分类面

假设超平面$(\mathbf{w}, b)$能将样本正确分类，则对于$(\mathbf{x}_i, y_i)$有：

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

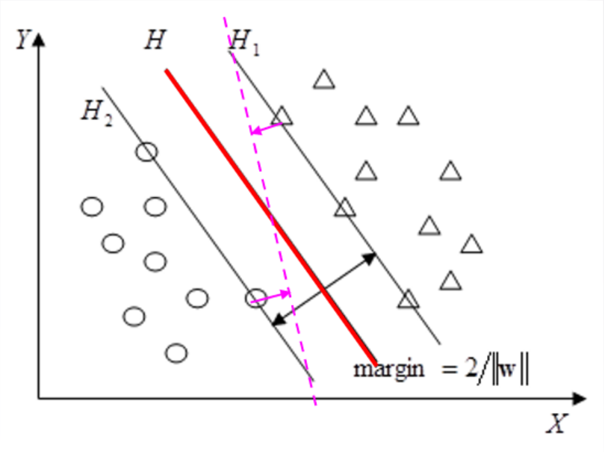
2）若y_i=-1，则有$\mathbf{w}^T\mathbf{x}_i + b < 0$

将判别函数进行归一化，使两类所有样本都满足$|f(\mathbf{x})| \geq 1$：

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

$$\Downarrow$$

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad i = 1,2,\cdots,n.$$



则距分类面最近的样本使等号成立$|f(\mathbf{x})| = 1$，被称为支持向量。

两异类支持向量到超平面的距离之和为 $r = \frac{2}{\|\mathbf{w}\|}$，此即为间隔(margin)。

4.4 支持向量机

1、寻找最优分类面

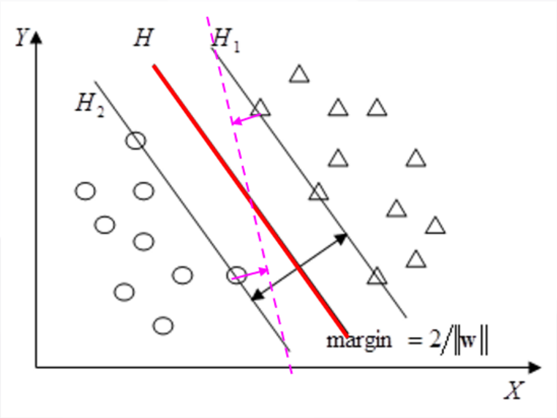
最好的划分超平面就是使间隔最大的超平面。要找到具有最大间隔（maximum margin）的划分超平面，就是要找到$\mathbf{w}$和b满足：

$$\max_{\mathbf{w},b} \frac{2}{\|\mathbf{w}\|}$$
$$s.t.\ y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad i = 1,2,\cdots,n.$$

使$\frac{2}{\|\mathbf{w}\|}$最大，等价于$\|\mathbf{w}\|$或$\|\mathbf{w}\|^2$最小。则上式可重新写成（支持向量机的基本型）：

（公式1）

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$
$$s.t.\ y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad i = 1,2,\cdots,n.$$



不等式约束优化问题

$$\text{minimize} \quad f(\boldsymbol{x})$$

$$subject\ to \quad g_i(\boldsymbol{x}) \le 0$$

$$i = 1,2,\dots,m$$

构造拉格朗日乘子

$$L(\boldsymbol{x}, \boldsymbol{\alpha}) = f(\boldsymbol{x}) + \sum_i \alpha_i g_i(\boldsymbol{x})$$

（Karush-Kuhn-Tucker, KKT）一阶必要条件

$$\frac{\partial L}{\partial \boldsymbol{x}} = \nabla f(\boldsymbol{x}) + \sum_i \alpha_i \nabla g_i(\boldsymbol{x}) = 0$$

$$\frac{\partial L}{\partial \alpha_i} = g_i(\boldsymbol{x}) \le 0$$

$$\alpha_i g_i(\boldsymbol{x}) = 0$$

2、用拉格朗日方程求解对偶问题

推导过程：

拉格朗日方程： $$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{n} \alpha_i\left(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\right)$$

（公式2）

$$\max_{\alpha_i \geq 0} L(\mathbf{w}, b, \alpha) = \begin{cases} \frac{1}{2}\|\mathbf{w}\|^2, y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \\ +\infty, otherwise \end{cases}$$

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

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

（公式1）

$$\min_{\mathbf{w},b} \max_{\alpha_i \geq 0} L(\mathbf{w}, b, \alpha)$$

$$\max_{\alpha_i \geq 0} \min_{\mathbf{w},b} L(\mathbf{w}, b, \alpha)$$

2、用拉格朗日方程求解对偶问题

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

（公式2）

$$\min_{\mathbf{w},b} \max_{\alpha_i \geq 0} \mathrm{L}(\mathbf{w}, b, \alpha)$$

$$\Downarrow$$

$$\max_{\alpha_i \geq 0} \min_{\mathbf{w},b} \mathrm{L}(\mathbf{w}, b, \alpha)$$

对$\mathbf{w}, b$求偏导等于0：

$$\begin{cases} \dfrac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 \\ \dfrac{\partial L}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases}$$

（公式3）

把公式3代入拉格朗日方程：

$$Q(\alpha) = \min_{\mathbf{w},b} \mathrm{L}(\mathbf{w}, b, \alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \,.\, i, j = 1,2,\cdots,n$$

$$\text{其中} \sum_{i=1}^{n} \alpha_i y_i = 0, \alpha_i \geq 0, \qquad i = 1,2,\cdots,n.$$

2、用拉格朗日方程求解对偶问题

最后问题变成：（公式4）

$$\begin{cases} \max\limits_{\alpha}\left(\sum\limits_{i=1}^{n}\alpha_i-\frac{1}{2}\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j\right) \\ s.t.(1)\sum\limits_{i=1}^{n}\alpha_i y_i=0,(2)\alpha_i\geq 0, i=1,2,\cdots,n. \end{cases}$$

解出α_i

根据公式3求出$\mathbf{w}$ 对 $\mathbf{w}$ 求偏导等于0：$\frac{\partial L}{\partial \mathbf{w}}=\mathbf{w}-\sum\limits_{i=1}^{n}\alpha_i y_i \mathbf{x}_i=0$（公式3）

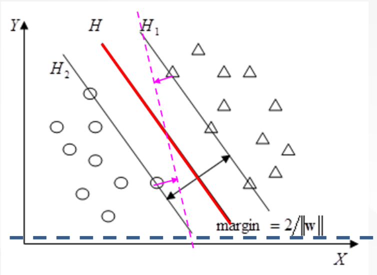
用支持向量求出b 对于支持向量$\mathbf{x}_i$，有 $y_i(\mathbf{w}^T\mathbf{x}_i+b)=1$

最终得到分类模型：$f(\mathbf{x})=\mathbf{w}^T\mathbf{x}+b=\sum_{i=1}^{n}a_i y_i \mathbf{x}_i^T\mathbf{x}+b$

（公式5）

4.4 支持向量机

2、用拉格朗日方程求解对偶问题



支持向量机模型：（公式1） $\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$ s.t. $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad i = 1,2,\cdots,n.$

拉格朗日方程：（公式2） $\mathrm{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{n} \alpha_i\left(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\right)$

$\min_{\mathbf{w},b} \max_{\alpha_i \geq 0} \mathrm{L}(\mathbf{w}, b, \alpha)$

$\max_{\alpha_i \geq 0} \min_{\mathbf{w},b} \mathrm{L}(\mathbf{w}, b, \alpha)$

最终分类模型：（公式5） $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = \sum_{i=1}^{n} a_i y_i \mathbf{x}_i^T\mathbf{x} + b$

推导过程需要满足KKT条件 (Karush-Kuhn-Tucker)

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

人设定的

对任意训练样本$(\mathbf{x}_i, y_i)$，总有a_i=0或y_if($\mathbf{x}_i$)=1。若 $a_i = 0$，则该样本不会在最终模型中出现，不会对$f(\mathbf{x})$有任何影响；若$a_i > 0$，则必有y_if($\mathbf{x}_i$)=1，所对应的样本点位于最大间隔的边界上，是一个支持向量。因此支持向量机有一个重要性质：训练完成后，大部分的训练样本都不需要保留，最终模型仅与支持向量有关。

2、用拉格朗日方程求解对偶问题

求解过程

A、求解α_i

公式4
$$\begin{cases}\max\limits_{\alpha}\left(\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j\right)\\ s.t.(1)\sum_{i=1}^{n}\alpha_i y_i=0,(2)\alpha_i\geq 0,i=1,2,\cdots,n.\end{cases}$$

二次规划问题，可使用通用的二次规划算法来求解。

SMO（Sequential Minimal Optimization）算法：

先固定α_i之外的所有参数，然后求α_i上的极值，由于存在约束$\sum_{i=1}^{n}\alpha_i y_i=0$，若固定$\alpha_i$之外的其他变量，则$\alpha_i$可由其他变量导出，于是，SMO每次选择两个变量$\alpha_i$和$\alpha_j$，并固定其他参数，这样在参数初始化后，SMO不断执行如下两个步骤直至收敛：

1）选取一对需更新的变量α_i和α_j；

2）固定α_i和α_j以外的参数，求解公式4获得更新后的α_i和α_j。

2、用拉格朗日方程求解对偶问题

求解过程

B、求解法向量w

根据以下公式求解 $\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0$

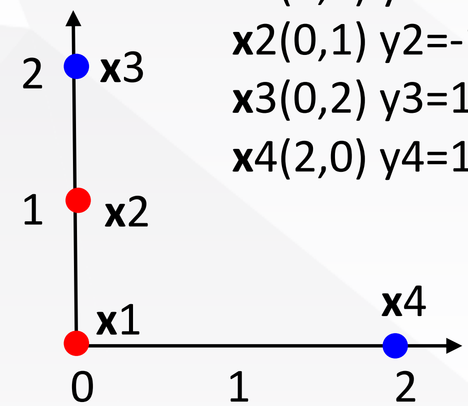
C、求解偏移项b

对于支持向量$\mathbf{x}_i$，有 $y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$。可以使用所有支持向量求解的平均值作为$b$：

$$b_i = \frac{1}{y_i} - \mathbf{w}^T\mathbf{x}_i \qquad \mathrm{b} = \frac{1}{m}\sum_{s\in S}\left(\frac{1}{y_s} - \sum_{i\in S} a_i y_i \mathbf{x}_i^T \mathbf{x}_s\right)$$

$S = \{i|a_i > 0, i = 1,2,\cdots,m\}$为所有支持向量的下标集

2、用拉格朗日方程求解对偶问题



$\mathbf{x}1(0,0)$ y1=-1
$\mathbf{x}2(0,1)$ y2=-1
$\mathbf{x}3(0,2)$ y3=1
$\mathbf{x}4(2,0)$ y4=1

例：输入4个训练样本点，求解其线性SVM最大间隔分类超平面。

求解α_i：

$$Q(\alpha)=\sum_{i=1}^{4}\alpha_i-\frac{1}{2}\sum_{i=1}^{4}\sum_{j=1}^{4}a_ia_jy_iy_j\mathbf{x}_i^T\mathbf{x}_j$$

$$=(\alpha_1+\alpha_2+\alpha_3+\alpha_4)-\frac{1}{2}(\alpha_2^2-4\alpha_2a_3+4\alpha_3^2+4\alpha_4^2)$$

用Matlab中的二次规划求解：

$\max Q(\alpha)$满足：

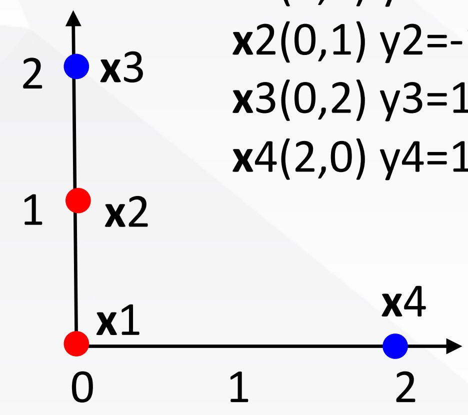
（1）$\sum_{i=1}^{n}\alpha_iy_i=\alpha_3+\alpha_4-\alpha_1-\alpha_2=0$

（2）$\alpha_1,\alpha_2,\alpha_3,\alpha_4\geq 0$

得到：$\alpha_1=0,\alpha_2=4,\alpha_3=3,\alpha_4=1$，非支持向量x1的$\alpha$值为0。

2、用拉格朗日方程求解对偶问题

例：输入四个训练样本点，求解其线性SVM最大间隔分类超平面。



x1(0,0) y1=-1
x2(0,1) y2=-1
x3(0,2) y3=1
x4(2,0) y4=1

求解α_i： $Q(\alpha)=\sum_{i=1}^{4}\alpha_i-\frac{1}{2}\sum_{i=1}^{4}\sum_{j=1}^{4}a_i a_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j$

$$=(\alpha_1+\alpha_2+\alpha_3+\alpha_4)-\frac{1}{2}(\alpha_2^2-4\alpha_2 a_3+4\alpha_3^2+4\alpha_4^2)$$

$$\boxed{\begin{array}{c}\alpha_1=0,\alpha_2=4,\\ \alpha_3=3,\alpha_4=1\end{array}}$$

求解**w**: $\mathbf{w}-\sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i=0 \Rightarrow \mathbf{w}=\sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i$

$$\mathbf{w}=-4\cdot[0,1]^T+3\cdot[0,2]^T+[2,0]^T=[2,2]^T$$

求解b:使用所有支持向量求解的平均值

$$b_i=\frac{1}{y_i}-\mathbf{w}^T\mathbf{x}_i \Rightarrow b=\frac{(-1-2)+(1-4)+(1-4)}{3}=-3$$

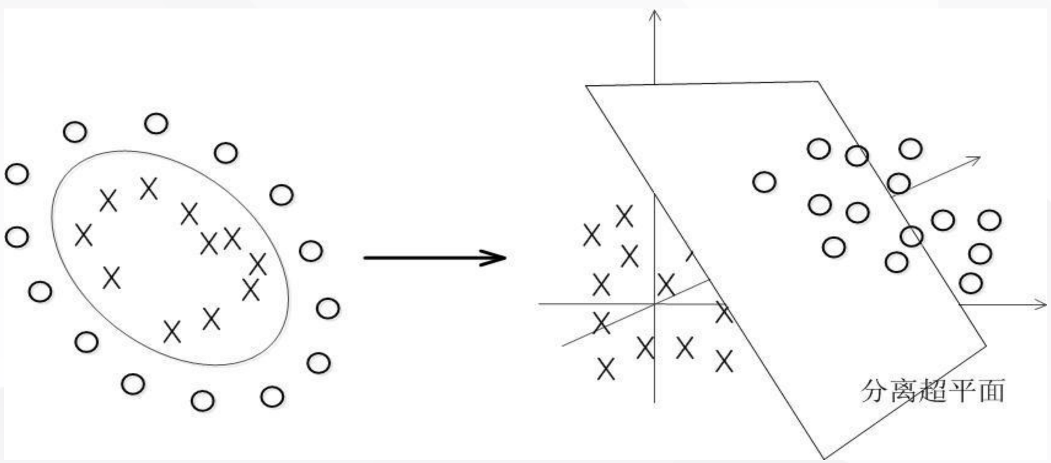
即决策平面为： $f(\mathbf{x})=\mathbf{w}^T\mathbf{x}+b=[2,2]^T\mathbf{x}-3$

4.4 支持向量机

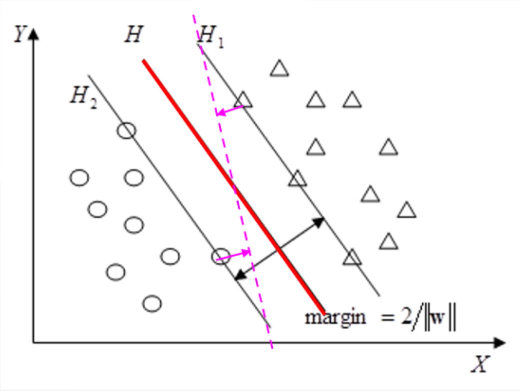
3、核函数

事实上，大部分的数据是线性不可分的。此时，可以基于原本的数据，适当增加数据的维度，将数据映射到一个新的空间(一般称之为特征空间)，形成新的样本数据，使其能够线性可分。

如果原始空间是有限维，那么一定存在一个高维特征空间使样本可分。



数据升维



理想的线性可分

4.4 支持向量机

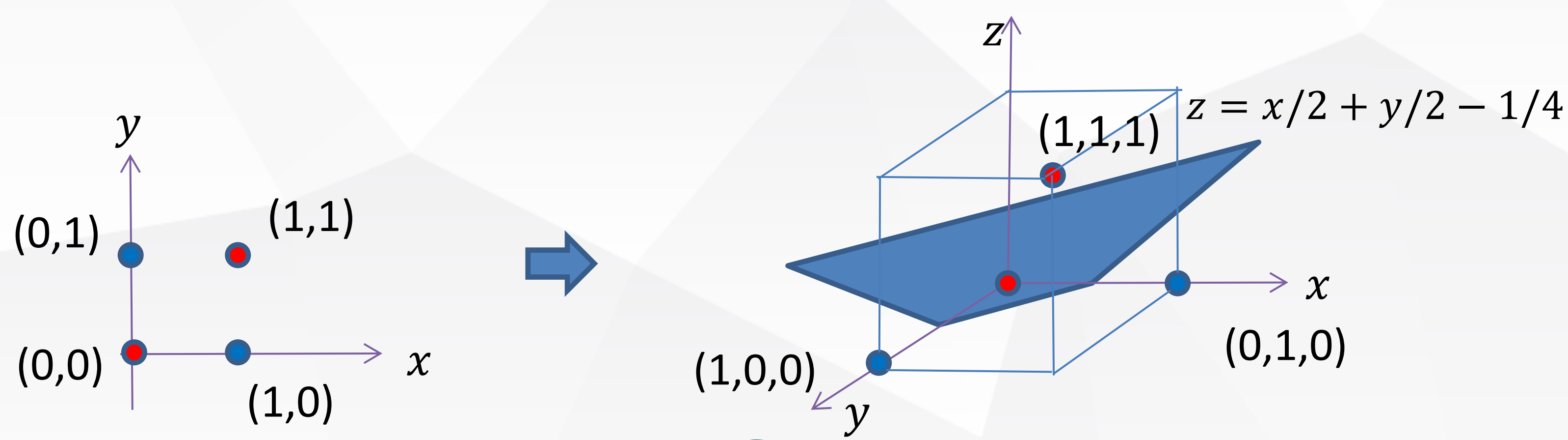
3、核函数

例如：经典的XOR问题，如何对{(0,0),(1,1)}和{(1,0),(0,1)}进行分类？

在二维平面，无法对其进行线性分类。我们为数据增加第三个维度，并设置其值为x_3=x_1*x_2，这样就得到了新的一组输入数据，**x**1(0,0,0)、**x**2(1,1,1)、**x**3(1,0,0)、**x**4(0,1,0)。

对新的数据，得到分类超平面为：$z = x/2 + y/2 - 1/4$

$$f(\mathbf{x}) = [-1,-1,2]\mathbf{x} + 1/2$$



3、核函数

回顾一下线性可分时支持向量机的推导

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = 0$$

基本型

$$\begin{cases} \min\limits_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \\ s.t.\ y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad i = 1,2,\cdots,n. \end{cases}$$

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

$$\begin{cases} Q(\alpha) = \sum\limits_{i=1}^{n}\alpha_i - \frac{1}{2}\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}a_i a_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j \\ \text{其中} \sum\limits_{i=1}^{n}\alpha_i y_i = 0, \alpha_i \geq 0, \qquad i = 1,2,\cdots,n. \end{cases}$$

3、核函数

令**x**是原空间的特征向量，$\phi(\mathbf{x})$是将**x**映射后的特征向量，则在特征空间中划分超平面所对应的模型可表示为：

$$f(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b$$（w和b是模型参数）

类似地，有：
$$\begin{cases} \min\limits_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \\ s.t.\ y_i(\mathbf{w}^T\phi(\mathbf{x}_i)+b) \geq 1, \quad i = 1,2,\cdots,n. \end{cases}$$

其对偶问题是：
$$\begin{cases} \max\limits_{\alpha}\left(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)\right) \\ s.t. \sum_{i=1}^{n}\alpha_i y_i = 0, \alpha_i \geq 0, \quad i = 1,2,\cdots,n. \end{cases}$$

上式中$\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$是样本$\mathbf{x}_i$和$\mathbf{x}_j$映射到特征空间后的内积。

3、核函数

由于特征空间维数可能很高，甚至可能是无穷，直接计算$\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$通常是困难的。为了避开这个障碍，可以设想这样一个函数：

$$\kappa(\mathbf{x}_i,\mathbf{x}_j)=\langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\rangle=\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$$

即向量$\mathbf{x}_i$和$\mathbf{x}_j$在特征空间的内积等于它们在原始样本空间中通过函数$\kappa(\cdot,\cdot)$计算的结果。

例如，定义二维向量$\mathbf{x}(x_1,x_2)$的高维映射：

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

则向量$\mathbf{x}_1(\alpha_1,\alpha_2)$和$\mathbf{x}_2(\beta_1,\beta_2)$进行高维映射后求内积得：

$$\langle\phi(\mathbf{x}_1),\phi(\mathbf{x}_2)\rangle=2\alpha_1\beta_1+\alpha_1^2\beta_1^2+2\alpha_2\beta_2+\alpha_2^2\beta_2^2+2\alpha_1\alpha_2\beta_1\beta_2+1$$

定义函数：$\kappa(\mathbf{x}_1,\mathbf{x}_2)=(\langle\mathbf{x}_1,\mathbf{x}_2\rangle+1)^2$

则将$\mathbf{x}_1$和$\mathbf{x}_2$代入得：$2\alpha_1\beta_1+\alpha_1^2\beta_1^2+2\alpha_2\beta_2+\alpha_2^2\beta_2^2+2\alpha_1\alpha_2\beta_1\beta_2+1$

4.4 支持向量机

3、核函数

有了这样的函数，我们就不必计算高维空间的内积。

$$
\begin{cases}
\max\limits_{\alpha}\left(\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)\right) \\
s.t.\sum_{i=1}^{n}\alpha_i y_i=0,\alpha_i\geq 0, i=1,2,\cdots,n.
\end{cases}
$$

而是通过该函数在低维空间完成计算

$$
\begin{cases}
\max\limits_{\alpha}\left(\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \kappa(\mathbf{x}_i,\mathbf{x}_j)\right) \\
s.t.\sum_{i=1}^{n}\alpha_i y_i=0,\alpha_i\geq 0, \qquad i=1,2,\cdots,n.
\end{cases}
$$

4.4 支持向量机

3、核函数

求解后可得决策规则：

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

$$=\sum_{i=1}^{n} \alpha_i y_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + b$$

$$=\sum_{i=1}^{n} \alpha_i y_i \kappa(\mathbf{x}_i, \mathbf{x}) + b$$

这里的$\kappa(\cdot,\cdot)$就是核函数（kernel function）。上式显示出模型最优解可通过训练样本的核函数展开，这一展开也称为支持向量展开（support vector expansion）。

线性可分时分类模型：

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = \sum_{i=1}^{n} a_i y_i \mathrm{x}_i^T \mathbf{x} + b$$

3、核函数

在不知道特征映射的形式时，我们并不知道什么样的核函数是合适的，核函数仅是隐式地定义了这个特征空间。因此核函数选择成为支持向量机的最大变数。若核函数选择不合适，意味着将样本映射到一个不合适的特征空间，导致性能不佳。

常用的核函数：

① 线性核：$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$

② 多项式核：$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)^d$， d为多项式的次数， d =1时退化为线性核。

③ 高斯核(σ>0)：$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$

④ 拉普拉斯核(σ>0)：$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\sigma}\right)$

⑤ sigmoid核(β>0,θ>0)：$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta \mathbf{x}_i^T \mathbf{x}_j + \theta)$

4、核技巧

当把输入样本做中心点平移，则 b 可以消掉，决策超平面可简化为：

$$f(\mathbf{x})=\sum_{i=1}^{n}\alpha_i y_i \kappa(\mathbf{x}_i,\mathbf{x})+b \Rightarrow \sum_{i=1}^{n}\alpha_i y_i \kappa(\mathbf{x}_i,\mathbf{x})=0$$

就模式分类的输出空间而言，只需指定相应的α值就可得到决策超平面，无需显式计算出法向量**w**。可以利用如下定义的核矩阵参与运算：

$$K=\{\kappa(\mathbf{x}_i,\mathbf{x}_j)\}_{i,j=1}^{n}$$

核矩阵也称为**Gram**矩阵，是一个非负的对称矩阵。

优化目标函数变为：

$$\begin{cases}\max\limits_{\alpha} Q(\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K[i,j]\\ s.t.\sum_{i=1}^{n}\alpha_i y_i=0,\alpha_i\geq 0,\qquad i=1,2,\cdots,n.\end{cases}$$

4、核技巧

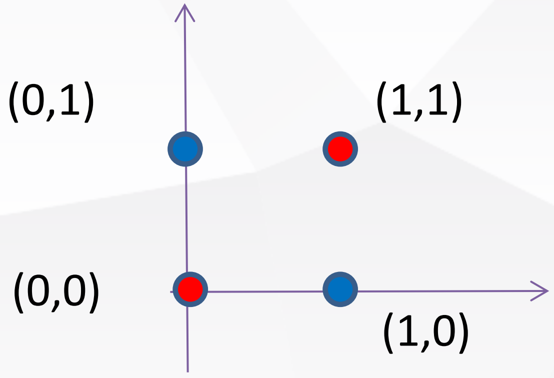
例：有如下训练样本和期望的输出响应，用核函数$\kappa(\mathbf{x}_1,\mathbf{x}_2)=(<\mathbf{x}_1,\mathbf{x}_2>+1)^2$进行分类

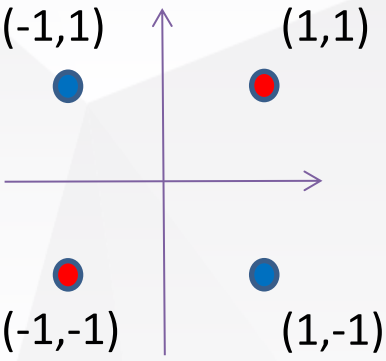
输入样本**x**	期望输出
(0,0)	-1
(0,1)	+1
(1,0)	+1
(1,1)	-1

预处理，对输入样本进行减均值除方差的归一化操作。

$$y=\frac{\mathrm{x}-\mu}{\sigma}$$

预处理结果	期望输出
(-1,-1)	-1
(-1,1)	+1
(1,-1)	+1
(1,1)	-1





4、核技巧

$$
\begin{cases}
\max\limits_{\alpha} Q(\alpha) = \sum\limits_{i=1}^{n} \alpha_i - \frac{1}{2}\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} \alpha_i\alpha_j y_i y_j K[i,j] \\
s.t. \sum\limits_{i=1}^{n} \alpha_i y_i = 0, \alpha_i \geq 0,
\end{cases}
$$

由核矩阵定义计算得到：

$\kappa(\mathbf{x}_1,\mathbf{x}_2)=(<\mathbf{x}_1,\mathbf{x}_2>+1)^2$

$$
K = \begin{matrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{matrix}
$$

预处理结果	期望输出
(-1,-1)	-1
(-1,1)	+1
(1,-1)	+1
(1,1)	-1

目标函数为：

$$
Q(\alpha) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2}(9\alpha_1^2 - 2\alpha_1 a_2 - 2\alpha_1 a_3 + 2\alpha_1 a_4 + 9\alpha_2^2 + 2\alpha_2 a_3 - 2\alpha_2 a_4 + 9\alpha_3^2 - 2\alpha_3 a_4 + 9\alpha_4^2)
$$

二次规划方法求解，得：$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \frac{1}{8}$

4、核技巧

$\kappa(\mathbf{x}_1,\mathbf{x}_2)=(<\mathbf{x}_1,\mathbf{x}_2>+1)^2$

$\alpha_1=\alpha_2=\alpha_3=\alpha_4=\frac{1}{8}$

预处理结果	期望输出
(-1,-1)	-1
(-1,1)	+1
(1,-1)	+1
(1,1)	-1

待决策样本$\mathbf{x}=(x_1,x_2)$

则其决策超平面为:

$$f(\mathbf{x})=\sum_{i=1}^{n}\alpha_i y_i\kappa(\mathbf{x}_i,\mathbf{x})+b \xrightarrow[\text{核技巧}]{\text{中心点平移}} \sum_{i=1}^{n}\alpha_i y_i\kappa(\mathbf{x}_i,\mathbf{x})=0$$

$$\sum_{i=1}^{4}\alpha_i y_i\,\kappa(\mathbf{x}_i,\mathbf{x})=-\frac{1}{8}(1-x_1-x_2)^2+\frac{1}{8}(1-x_1+x_2)^2+\frac{1}{8}(1+x_1-x_2)^2-\frac{1}{8}(1+x_1+x_2)^2=0$$

化简得决策面方程：$-x_1x_2=0$

5、软间隔

前面一直假定训练样本在样本空间或特征空间线性可分，即存在一个超平面将不同类的样本完全划分开。但现实中，很难确定合适的核函数使得训练样本在特征空间中线性可分。缓解该问题的办法是允许支持向量机出错。

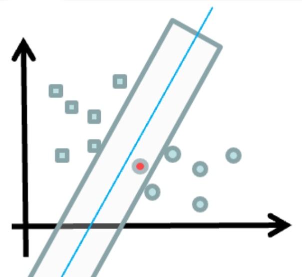
我们希望找到一个最优超平面，使其对整个训练集的平均分类误差达到最小。如果数据$(\mathbf{x}_i, y_i)$不满足：

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

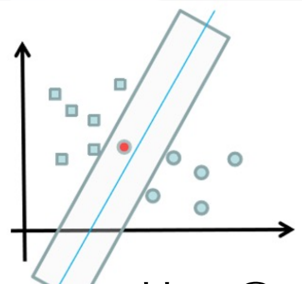
分为两种情况：

① 数据点落在分离区域之内，但在决策面正确的一侧

② 数据点落在决策面错误的一侧



情况①



情况②

5、软间隔

为处理不可分问题，引入松弛变量{ξ $_i$>=0}到分离超平面的定义中：

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, i = 1,2,\cdots,n$$

①当$0 < \xi_i \leq 1$时，数据点落入分离区域的内部，但在决策面正确的一侧

②当$\xi_i > 1$ 时，数据点落入决策面错误的一侧。

同时为松弛变量引入惩罚参数C>0。C值大时对误分类的惩罚增大，相反则减少。得到修改后的优化问题为:

$$\min_{\mathbf{w},b,\xi}\left(\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i\right) \quad s.t.\ y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1,2,\cdots,n.$$

目标函数包含了两层意思：最小化$\frac{1}{2}\|\mathbf{w}\|^2$以使分类间隔尽量大；最小化$C\sum_{i=1}^{n}\xi_i$使得错误分类的个数尽量少。

5、软间隔

修改后的优化问题:

$$\min_{\mathbf{w},b,\xi}\left(\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{n}\xi_i\right)$$

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

相应的Lagrange函数变为:

$$\mathrm{L}(\mathbf{w},b,\alpha,\xi,\mu)$$
$$=\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{n}\xi_i+\sum_{i=1}^{n}\alpha_i\big(1-\xi_i-y_i(\mathbf{w}^T\mathbf{x}_i+b)\big)-\sum_{i=1}^{n}\mu_i\xi_i$$

其中$\alpha_i\geq 0,\mu_i\geq 0$是拉格朗日乘子.

5、软间隔

与前面过程类似，对 $\mathbf{w}, b, \xi_i$ 求偏导等于0：

$$\begin{cases} \dfrac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 \\ \dfrac{\partial L}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0 \\ \dfrac{\partial L}{\partial \xi} = C - \alpha_i - \mu_i = 0 \end{cases}$$

代入Lagrange方程，得到软间隔最大规划：

$$\begin{cases} \max\limits_{\alpha} \left(\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \right) \\ s.t. (1) \sum_{i=1}^{n} \alpha_i y_i = 0, (2) 0 \le \alpha_i \le C, \qquad i = 1,2,\cdots,n. \end{cases}$$

仅比硬间隔最大规划多一个α上限C.

5、软间隔

上述推导满足KKT条件：

KKT条件：
$$
\begin{cases}
① \ \alpha_i \geq 0，\ \mu_i \geq 0 \\
② \ 1-\xi_i-y_i(\mathrm{w}^T\mathrm{x}_i+b) \leq 0 \\
③ \ \alpha_i\big(1-\xi_i-y_i(\mathrm{w}^T\mathrm{x}_i+b)\big)=0 \\
④ \ \xi_i \geq 0 \\
⑤ \ \mu_i\xi_i=0
\end{cases}
$$

对w, b, ξ_i求偏导等于0

$$
\begin{cases}
\dfrac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \displaystyle\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 \\
\dfrac{\partial L}{\partial b} = \displaystyle\sum_{i=1}^{n} \alpha_i y_i = 0 \\
\dfrac{\partial L}{\partial \xi} = C - \alpha_i - \mu_i = 0
\end{cases}
$$

对于任意训练样本$(\mathbf{x}_i, y_i)$，总有a_i=0或y_if($\mathbf{x}_i$)=1-ξ_i。

1）若 $a_i=0$，则该样本不会对$f(\mathbf{x})$有任何影响。

2）若$a_i>0$，则必有y_if($\mathbf{x}_i$)=1-ξ_i，即该样本是支持向量。此时若$\alpha_i<C$，则$\mu_i>0$，进而$\xi_i=0$，即该样本恰好在最大间隔边界上；$\alpha_i=C$，则$\mu_i=0$，此时若$\xi_i\leq$1则该样本落在最大间隔内，若$\xi_i>$ 1则该样本被错分。由此可见，软间隔支持向量机的模型仅与支持向量有关。

4.4 支持向量机

6、SVM解决多分类问题

SVM算法最初是为二值分类问题设计的，当处理多类问题时，就需要构造合适的多类分类器。构造SVM多类分类器的方法主要有两类：

（1）直接法。一次性将多个分类面的参数求解合并到一个最优化问题中，多目标函数优化， “一次性”实现多类分类。这种方法看似简单，但计算复杂度比较高，实现起来比较困难，只适合用于小型问题。

（2）间接法。主要是通过组合多个二分类器来实现多分类器的构造，常见的方法有one-against-one和one-against-all两种。

https://blog.csdn.net/baoyan2015/article/details/70265459

https://blog.csdn.net/weixin_42296976/article/details/81946047

6、SVM解决多分类问题

一对其余（one-versus-rest，OVR SVMs）：

训练时依次把某个类别的样本归为一类,其他剩余的样本归为另一类，这样c个类别的样本就构造出了c个SVM。分类时将未知样本分类为具有最大分类函数值的那类。

比如有5个类别，第一次就把类别1的样本定为正样本，其余2、3、4、5的样本合起来定为负样本，这样得到一个两类分类器；第二次把类别2 的样本定为正样本，把1、3、4、5的样本合起来定为负样本，得到一个分类器；如此下去，可以得到5个两类分类器（总是和类别的数目一致）。

优点：训练c个分类器，个数较少，分类速度相对较快。
问题：存在分类重叠现象、不可分类现象，“其余”那一类数据大，会导致“数据集偏斜”问题。

6、SVM解决多分类问题

一对一（one-against-one）：

按一对一训练，按一对一调用分类器分类。

第一个只回答“是第1类还是第2类”，第二个只回答“是第1类还是第3类”，第三个只回答“是第1类还是第4类”，如此下去，如果有c个类别，则有c(c-1)/2）个分类器。
分类时，调用c(c-1)/2）个分类器，投票多的为胜者。

这种方法有分类重叠的现象，但不会有不可分类现象。类别数多时，分类器数量太多，假如有1000个类别，则需约要500,000个分类器（类别数的平方量级）。

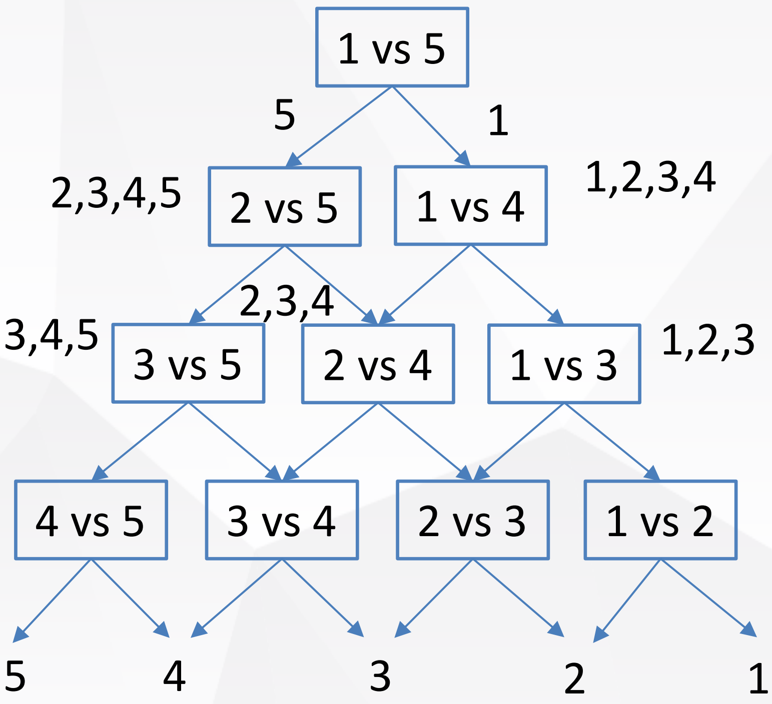
6、SVM解决多分类问题

有向无环图的分类方法DAG SVM（Directed Acyclic Graph）：

采用一对一方法训练，按有向无环图组织分类器，只需要调用c-1个分类器。

优点：分类速度快，且没有分类重叠和不可分类现象。

缺点：假如最一开始的分类器回答错误，则后面的分类器是无法纠正它的错误。



作业

第一次课作业

1、已知：

甲类样本4个：$[2, 2]^T$、 $[2, 3]^T$、 $[1, 2]^T$、 $[2, 1]^T$

乙类样本4个：$[-2, -2]^T$、 $[-3, -2]^T$、 $[-1, -2]^T$、 $[-2, -3]^T$

试用最近邻分类器对未知样本$[-1, -1]^T$ 和$[3, 2]^T$进行分类。

2、请自己查阅资料，了解球树（ball tree）算法，并对比球树算法与kd树算法的优缺点。

3、紧邻算法中有两种减少训练集样本数量的方法，即剪辑方法和压缩方法，请自己查阅资料，了解压缩法，并给出压缩法的基本步骤。

作业

第二次课作业

1、SVM推导过程中KKT条件是什么，并对其进行分析。

2、已知：

输入样本**x**	期望输出
$[0, 0]^{\mathrm{T}}$	-1
$[0, 1]^{\mathrm{T}}$	-1
$[0, 2]^{\mathrm{T}}$	+1
$[2, 0]^{\mathrm{T}}$	+1

且$\alpha_1 = 0, \alpha_2 = 4, \alpha_3 = 3, \alpha_4 = 1$

请求解**w** 和b，写出计算过程。