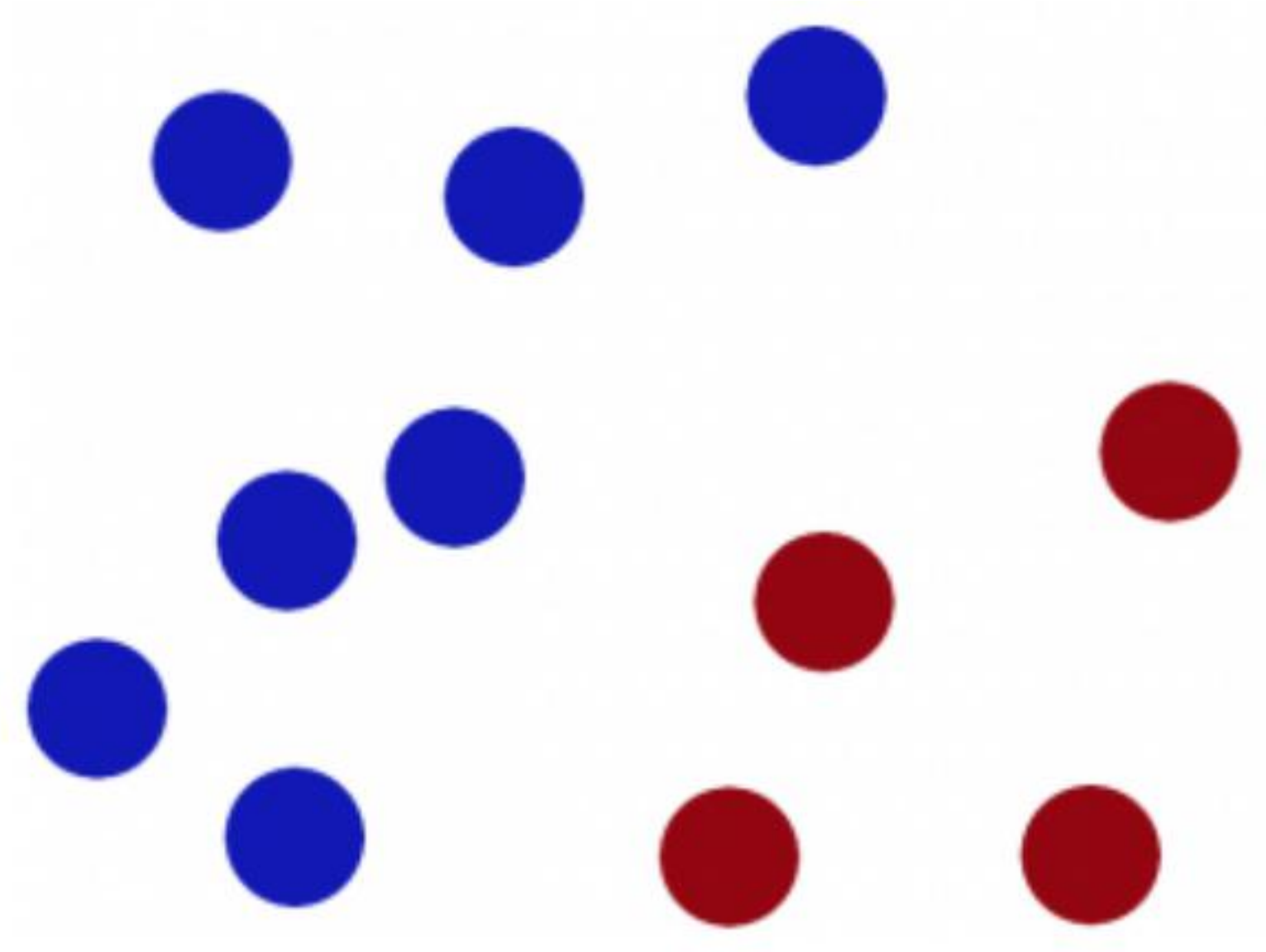
人工智能之机器学习

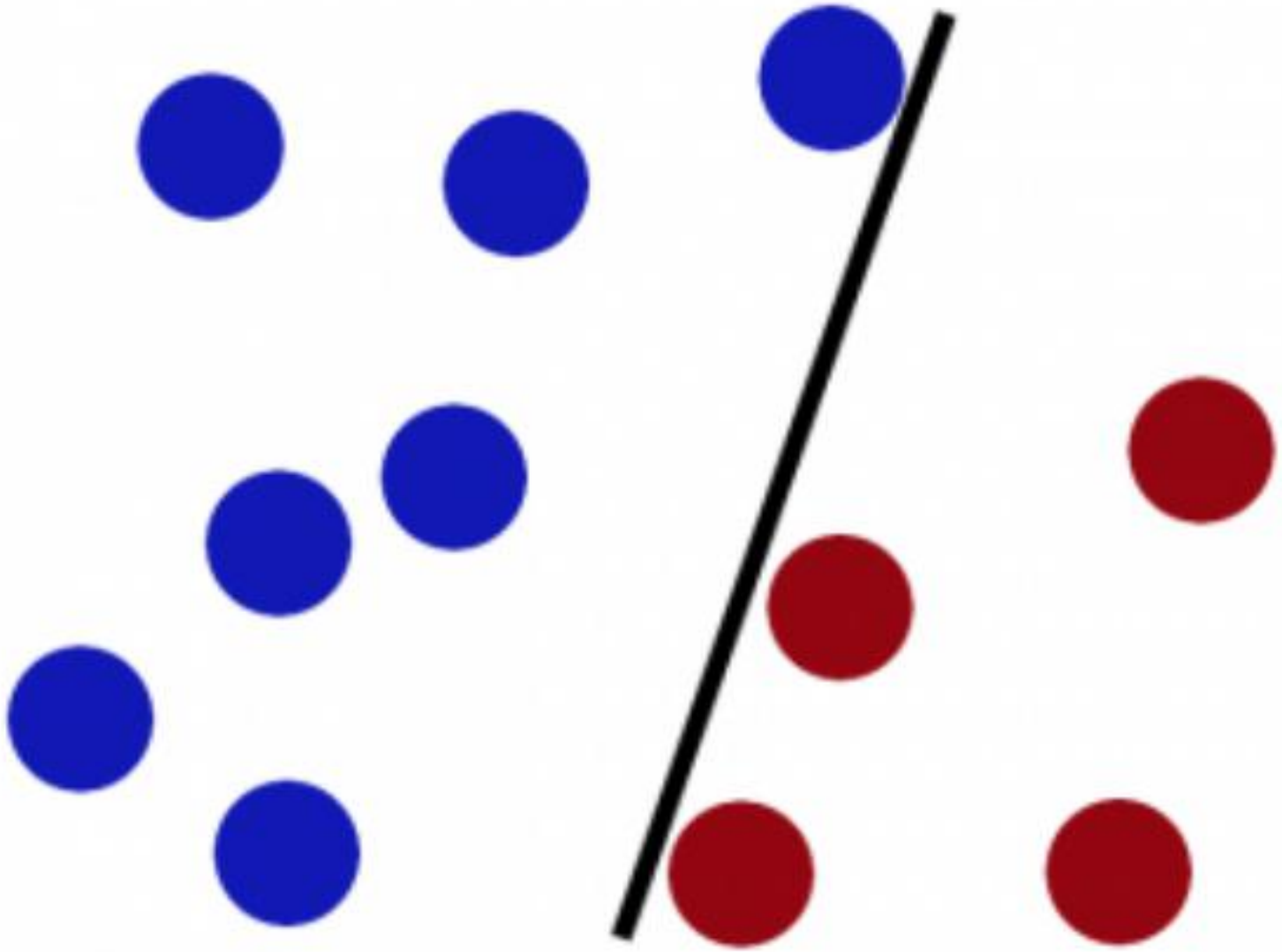
支持向量机（SVM）

主讲人：李老师

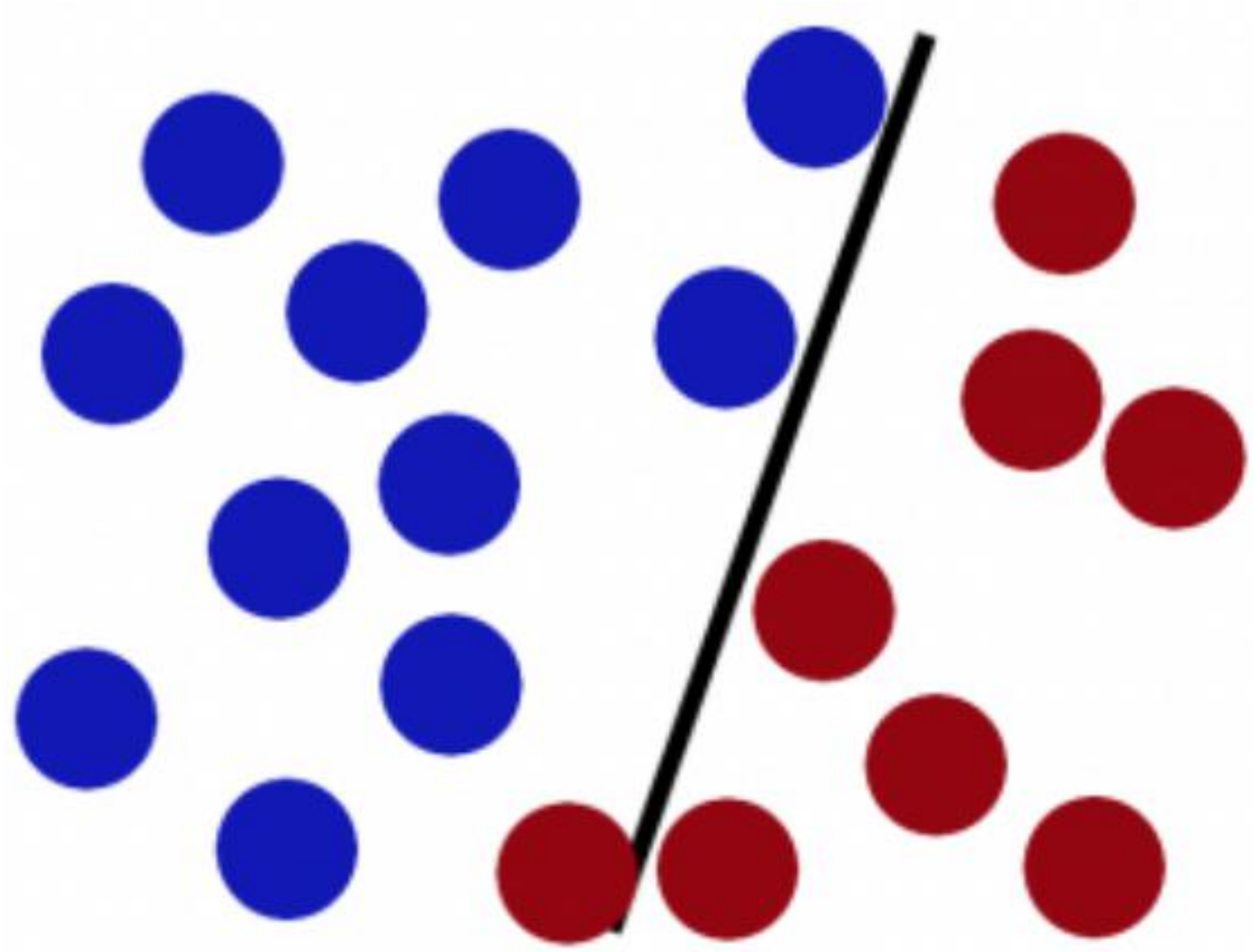
从一个故事出发



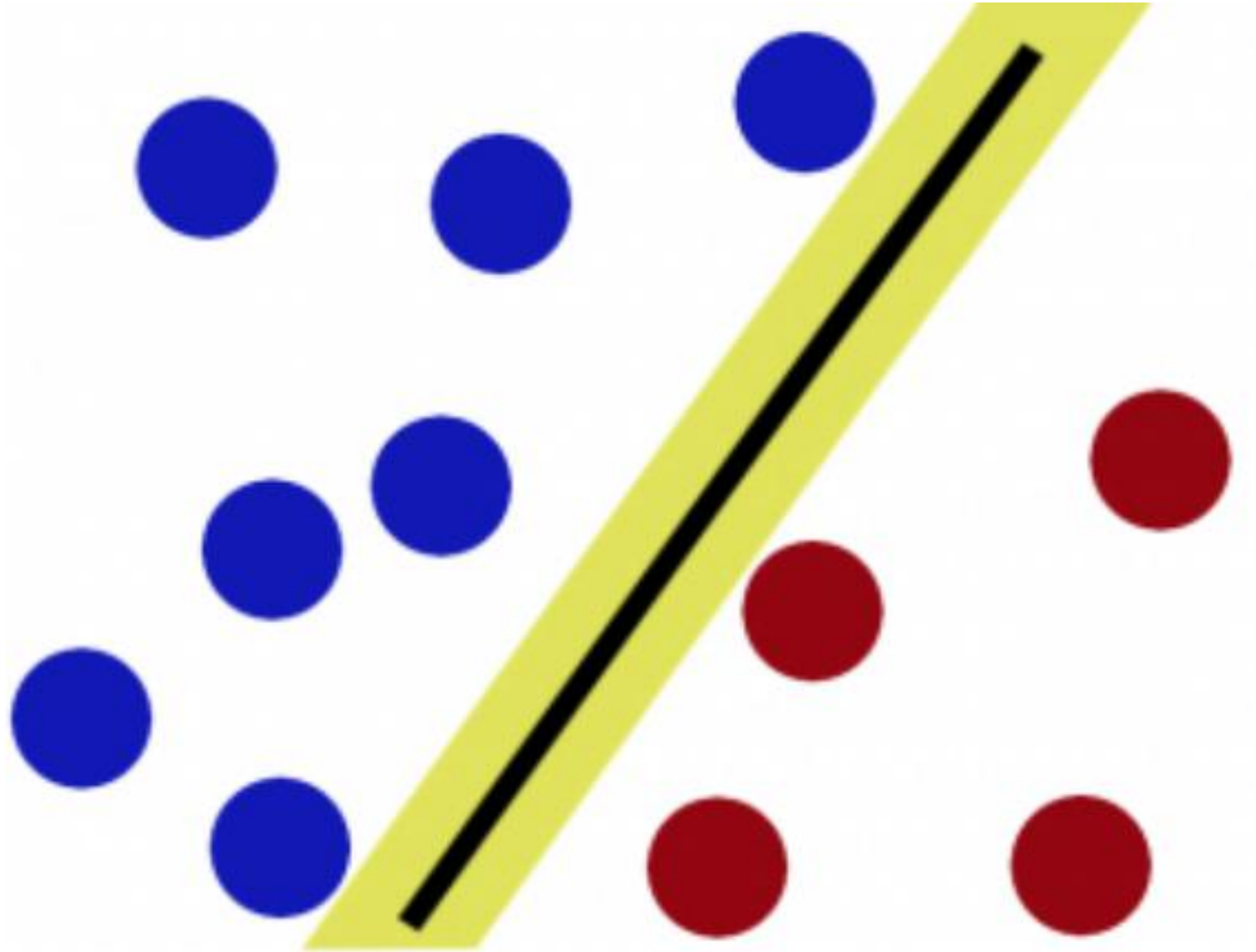
从一个故事出发



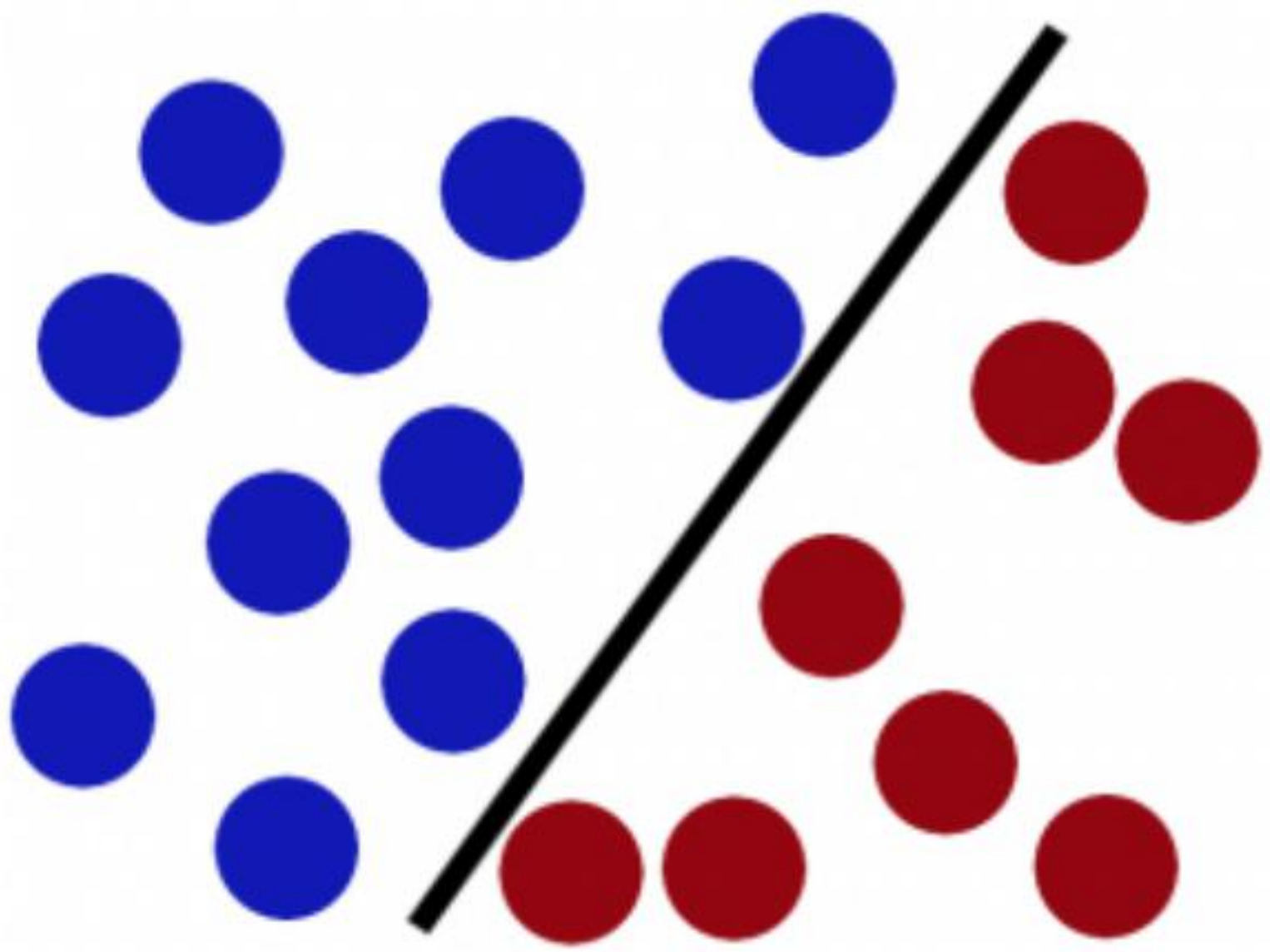
从一个故事出发



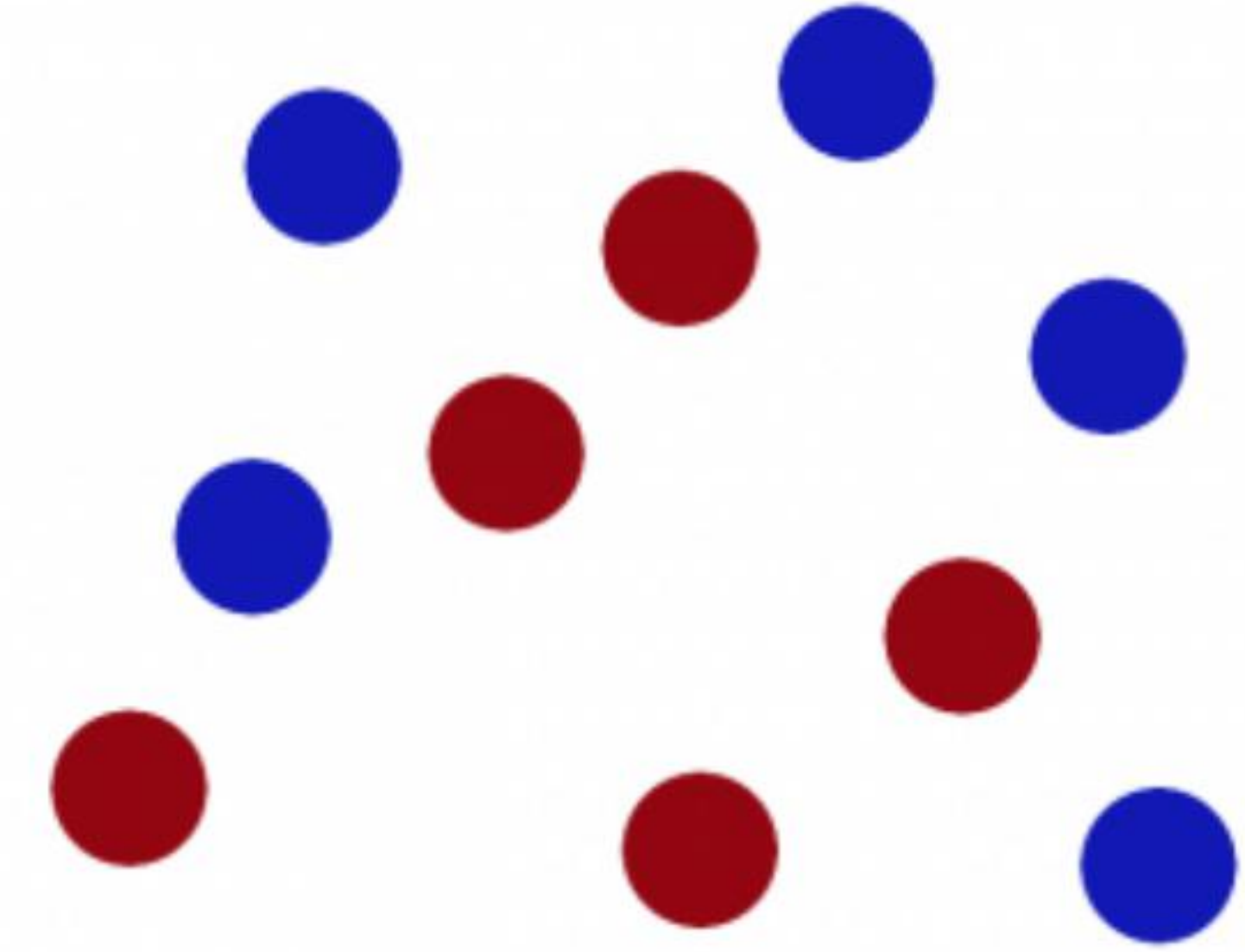
从一个故事出发



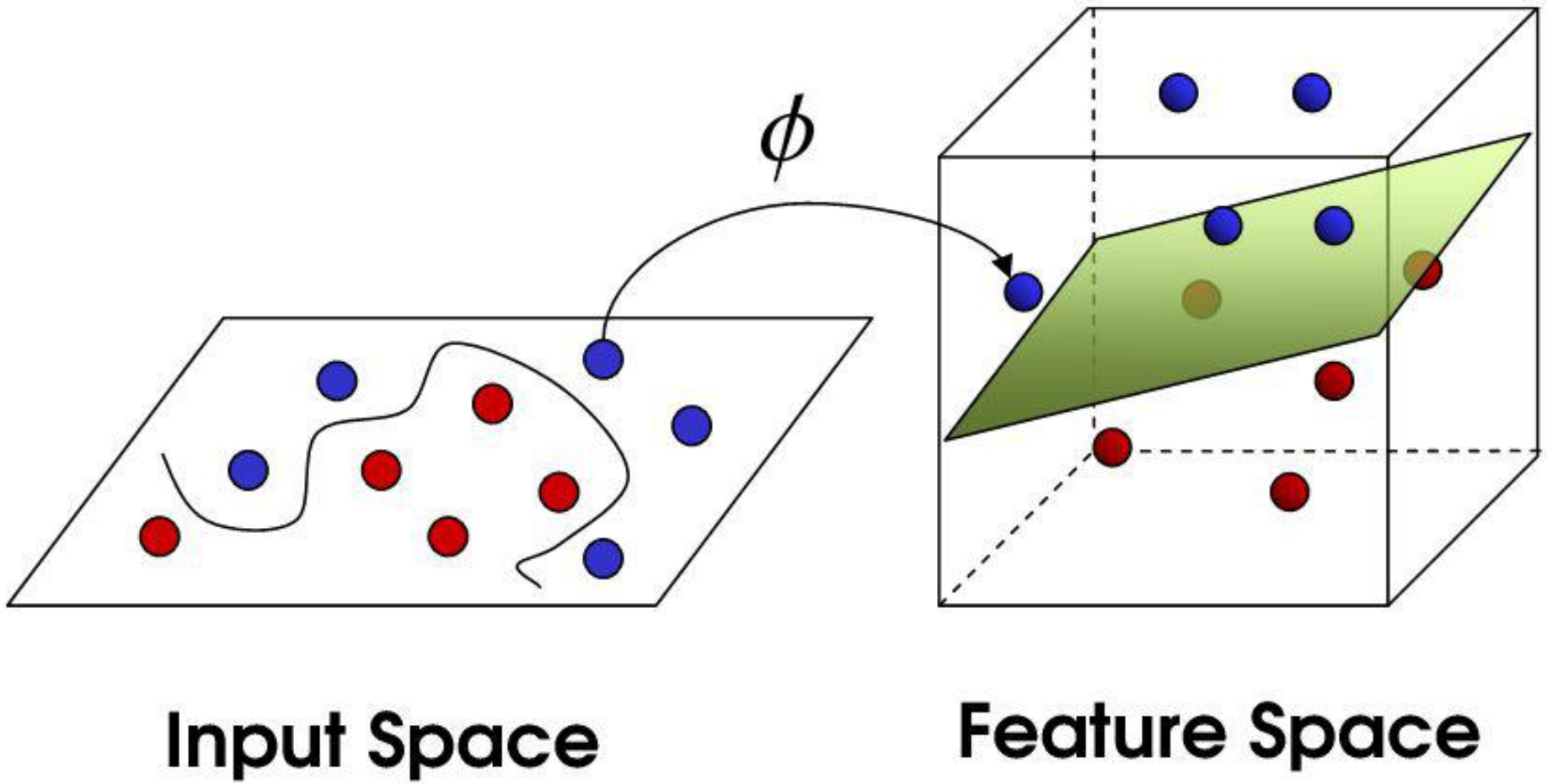
从一个故事出发



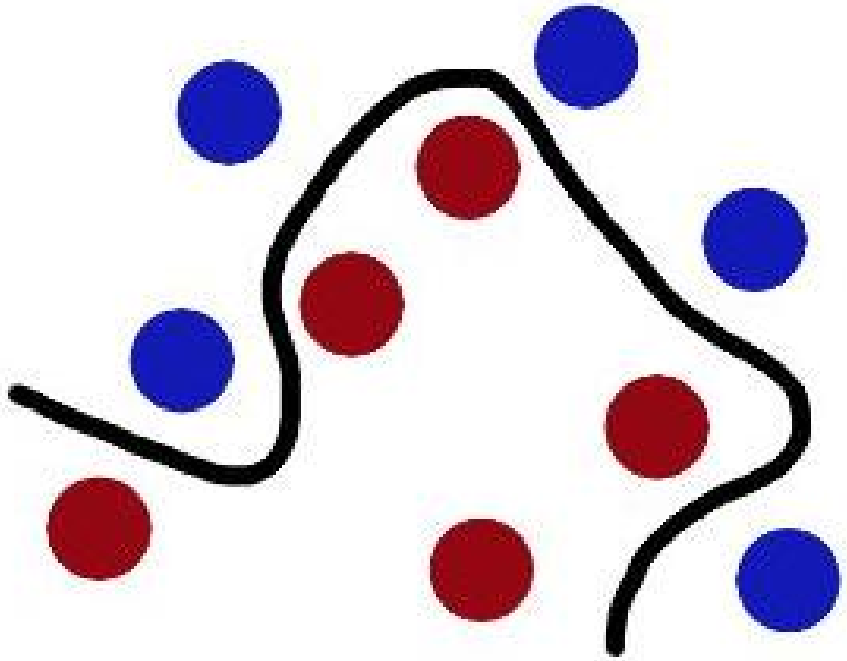
从一个故事出发



从一个故事出发



从一个故事出发



再之后，无聊的大人们把这些球叫做data，把棍子叫做classifier, 最大间隙trick叫做optimization，拍桌子叫做kernelling, 那张纸叫做hyperplane

http://bytesizebio.net/2014/02/05/support-vector-machines-explained-well/

支持向量机（Support Vector Machine，1995，SVM）

- 介绍
  - 支持向量机是一种二类分类模型。
- 支持向量机学习方法包含构建由简至繁的模型
  - 线性可分支持向量机(linear support vector machine in linearly separable case)
  - 线性支持向量机(linear support vector machine)
  - 非线性支持向量机(non- linear support vector machine)

拉格朗日乘子法

- 拉格朗日乘子法就是当我们的优化函数存在等值约束的情况下的一种最优化求解方式；其中参数α被称为**拉格朗日乘子**，要求α不等于0

$$\min_x f(x)$$

$$s.t: h_i(x) = 0, i = 1,2,..., p$$

$$\min_x f(x) + \sum_{i=1}^{p} \alpha_i h_i(x); \alpha_i \neq 0$$

拉格朗日乘子法理解

- 假设现在有一个二维的优化问题

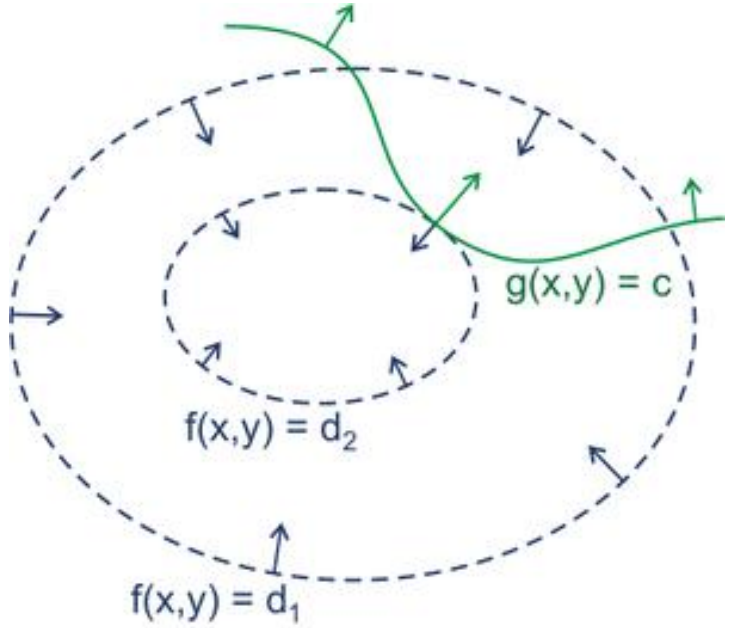
$$\min_{x,y} f(x,y)$$

$$s.t: g(x,y)=c$$

$$L(x,y,\alpha)=f(x,y)+\alpha(g(x,y)-c);\alpha\neq 0$$

$$\begin{cases}\min\limits_{x,y} f(x,y)\\ g(x,y)-c=0\end{cases}\Rightarrow \min_{x,y} L(x,y,\alpha)$$

$$\nabla_{x,y,\alpha} L(x,y,\alpha)=0,\alpha\neq 0$$



数学证明:https://wenku.baidu.com/view/ac56710e2e3f5727a5e962a7.html

KKT条件

- KKT条件是泛拉格朗日乘子法的一种形式；主要应用在当我们的优化函数存在不等值约束的情况下的一种最优化求解方式；KKT条件即满足不等式约束情况下的条件。

$$\min_x f(x)$$

$$s.t: h_k(x) = 0, k = 1,2,..., p$$

$$g_j(x) \le 0, j = 1,2,..., q$$

$$\downarrow$$

$$L(x,\alpha,\beta) = f(x) + \sum_{i=1}^{p} \alpha_i h_i(x) + \sum_{i=1}^{q} \beta_i g_i(x); \alpha_i \ne 0, \beta_i \ge 0$$

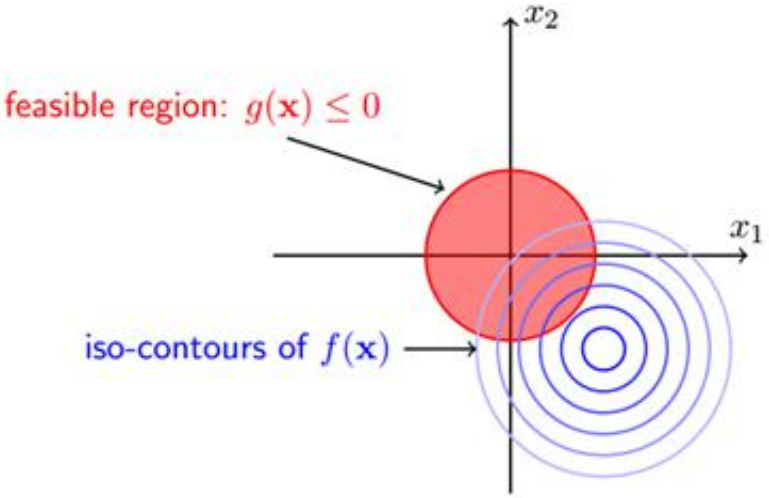
$$\min_x L(x,\alpha,\beta)$$

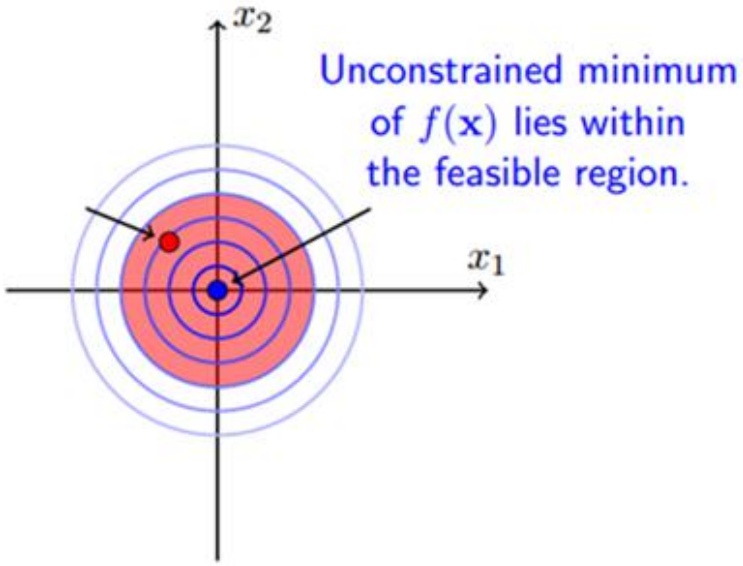
KKT条件理解

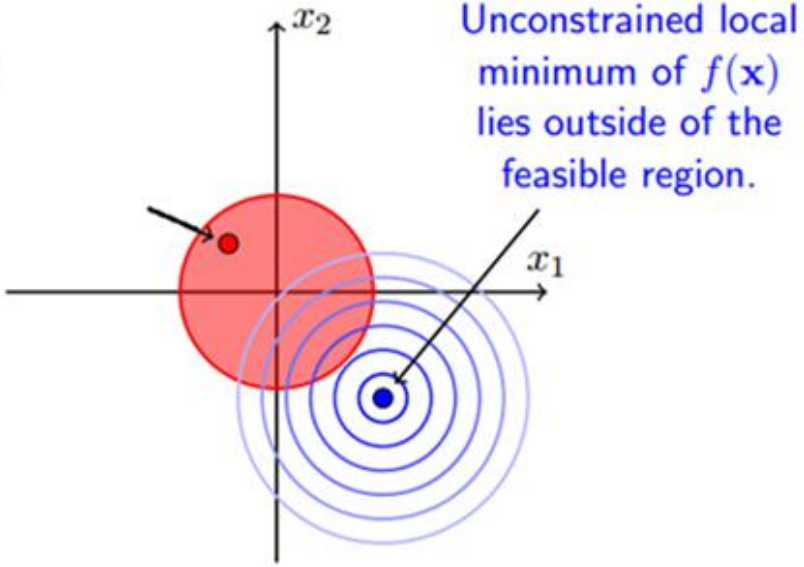
$$\min_x f(x)$$

$$s.t: g(x) \le 0$$

- 可行解必须在约束区域g(x)之内，由图可知可行解x只能在g(x)<0和g(x)=0的区域取得；
  - 当可行解x在g(x)<0的区域中的时候，此时直接极小化f(x)即可得到；
  - 当可行解x在g(x)=0的区域中的时候，此时直接等价于等式约束问题的求解。







KKT理解

$$\min_x f(x)$$

$$s.t:\ g_j(x) \le 0,\ j = 1,2,\ldots,q$$

$$L(x,\beta) = f(x) + \sum_{i=1}^{q} \beta_i g_i(x);\ \beta_i \ge 0$$

$$\because \left.\begin{array}{l} \beta_i \ge 0 \\ g_i(x) \le 0 \end{array}\right\} \Rightarrow \beta_i g_i(x) \le 0$$

$$\therefore f(x) = \max_{\beta} L(x,\beta)$$

$$\therefore \min_x f(x) = \min_x \max_{\beta} L(x,\beta)$$

KKT理解

$$\min_x f(x) = \min_x \max_\beta L(x,\beta) \qquad \min_x f(x) = \max_\beta \min_x L(x,\beta)$$

$$\min_x \max_\beta L(x,\beta) = \max_\beta \min_x L(x,\beta)$$

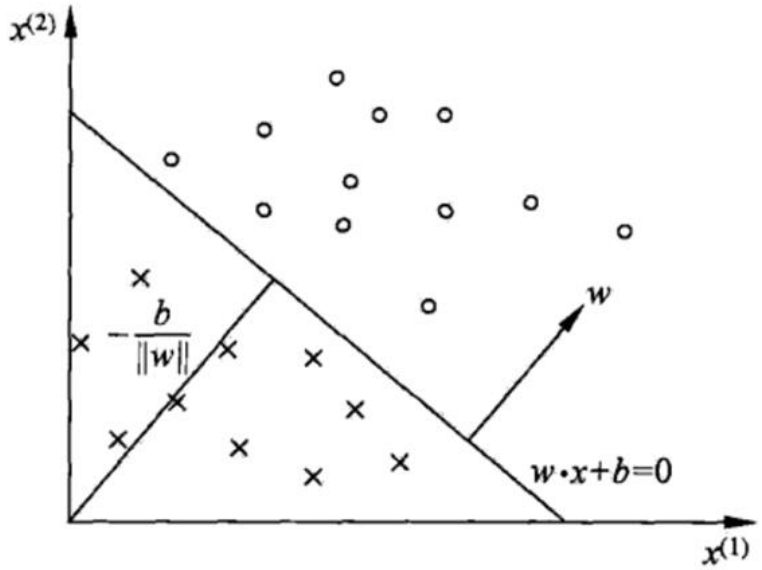
原问题

对偶问题

- 对偶问题的直观理解：最小的里面的那个最大的要比最大的那个里面的最小的大；从而就可以为原问题引入一个下界

感知器模型

- 感知器算法是最古老的分类算法之一，原理比较简单，不过模型的分类泛化能力比较弱，不过感知器模型是SVM、神经网络、深度学习等算法的基础。
- 感知器的思想很简单：比如某一个班上有很多的学员，分为男学员和女学员，感知器模型就是试图找到一条直线，能够把所有的男学员和女学员分隔开，如果是高维空间中，感知器模型寻找的就是一个超平面，能够把所有的二元类别分割开。感知器模型的前提是：**数据是线性可分的。**



感知器模型

- 对于m个样本，每个样本n维特征以及一个二元类别输出y，如下：

$$\left\{\left(x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}\right), \left(x_1^{(2)}, x_2^{(2)}, \ldots, x_n^{(2)}\right), \ldots, \left(x_1^{(m)}, x_2^{(m)}, \ldots, x_n^{(m)}\right)\right\}$$

- 目标是找到一个超平面，即：$\theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n = 0 \Rightarrow \theta \bullet x = 0$

让一个类别的样本满足：$\theta \bullet x > 0$；另外一个类别的满足：$\theta \bullet x < 0$

- 感知器模型为:

$$\hat{y} = sign(\theta \bullet x) = \begin{cases} +1, \theta \bullet x > 0 \\ -1, \theta \bullet x < 0 \end{cases}$$

感知器模型

$$d=\frac{|Ax_1+By_1+Cz_1+D|}{\sqrt{A^2+B^2+C^2}}$$

- 正确分类（预测和实际类别一致）：yθx>0，错误分类（预测和实际类别不一致）：yθx<0；所以我们可以定义我们的损失函数为：期望使分类错误的所有样本(k条样本)到超平面的距离之和最小。

$$L=\sum_{i=1}^{k}\frac{-y^{(i)}\theta\bullet x^{(i)}}{\|\theta\|_2}$$

- 因为此时分子和分母中都包含了θ值，当分子扩大N倍的时候，分母也会随之扩大，也就是说分子和分母之间存在倍数关系，所以可以固定分子或者分母为1，然后求另一个即分子或者分母的倒数的最小化作为损失函数，简化后的损失函数为（分母为1）：

$$L=-\sum_{i=1}^{k}y^{(i)}\theta\bullet x^{(i)}$$

感知器模型

$$L = -\sum_{i=1}^{k} y^{(i)} \theta \bullet x^{(i)}$$

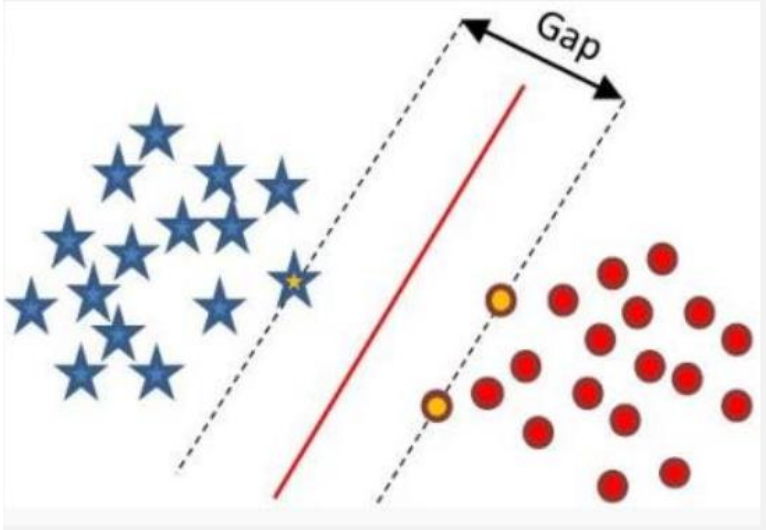
- 直接使用梯度下降法就可以对损失函数求解，不过由于这里的m是分类错误的样本点集合，不是固定的，所以我们不能使用批量梯度下降法(BGD)求解，只能使用随机梯度下降(SGD)或者小批量梯度下降(MBGD)；一般在感知器模型中使用SGD来求解。

$$\frac{\partial L(\theta)}{\partial \theta} = -\sum_{i=1}^{k} y^{(i)} x^{(i)}$$

$$\theta^{k+1} = \theta^{k} + \alpha y^{(i)} x^{(i)}$$

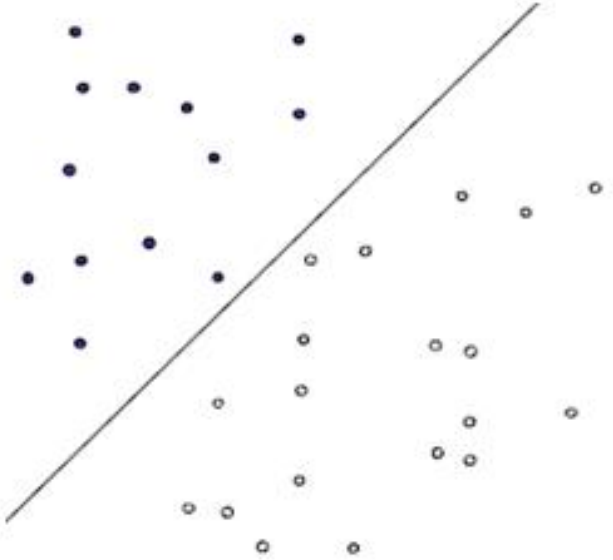
SVM

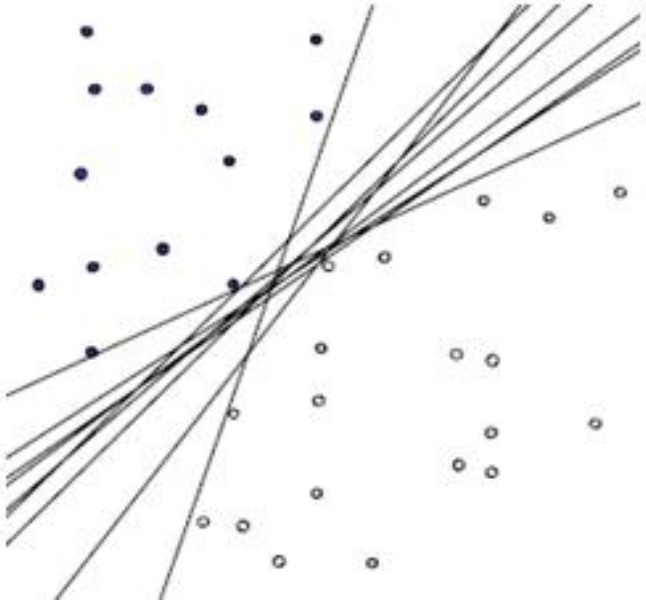
- 支持向量机(Support Vecor Machine, SVM)本身是一个**二元分类算法**，是对感知器算法模型的一种扩展，现在的SVM算法支持**线性分类**和**非线性分类**的分类应用，并且也能够直接将SVM应用于**回归应用**中，同时通过OvR或者OvO的方式我们也可以将SVM应用在**多元分类**领域中。在不考虑集成学习算法，不考虑特定的数据集的时候，在分类算法中SVM可以说是特别优秀的。



线性可分SVM

- 在感知器模型中，算法是在数据中找出一个划分超平面，让尽可能多的数据分布在这个平面的两侧，从而达到分类的效果，但是在实际数据中这个符合我们要求的超平面是可能存在多个的。





线性可分SVM

- 在感知器模型中，我们可以找到多个可以分类的超平面将数据分开，并且优化时希望所有的点(预测正确的点)都离超平面尽可能的远，但是实际上离超平面足够远的点基本上都是被正确分类的，所以这个是没有意义的；反而比较关心那些离超平面很近的点，这些点比较容易分错。所以说我们只要**让离超平面比较近的点尽可能的远离这个超平面**，那么我们的模型分类效果应该就会比较不错喽。SVM其实就是这个思想。

线性可分SVM

- **线性可分(Linearly Separable)**：在数据集中，如果可以找出一个超平面，将两组数据分开，那么这个数据集叫做线性可分数据。
- **线性不可分(Linear Inseparable)**：在数据集中，没法找出一个超平面，能够将两组数据分开，那么这个数据集就叫做线性不可分数据。
- **分割超平面(Separating Hyperplane)**：将数据集分割开来的直线/平面叫做分割超平面。
- **支持向量(Support Vector)**：离分割超平面最近的那些点叫做支持向量。
- **间隔(Margin)**：支持向量数据点到分割超平面的距离称为间隔。

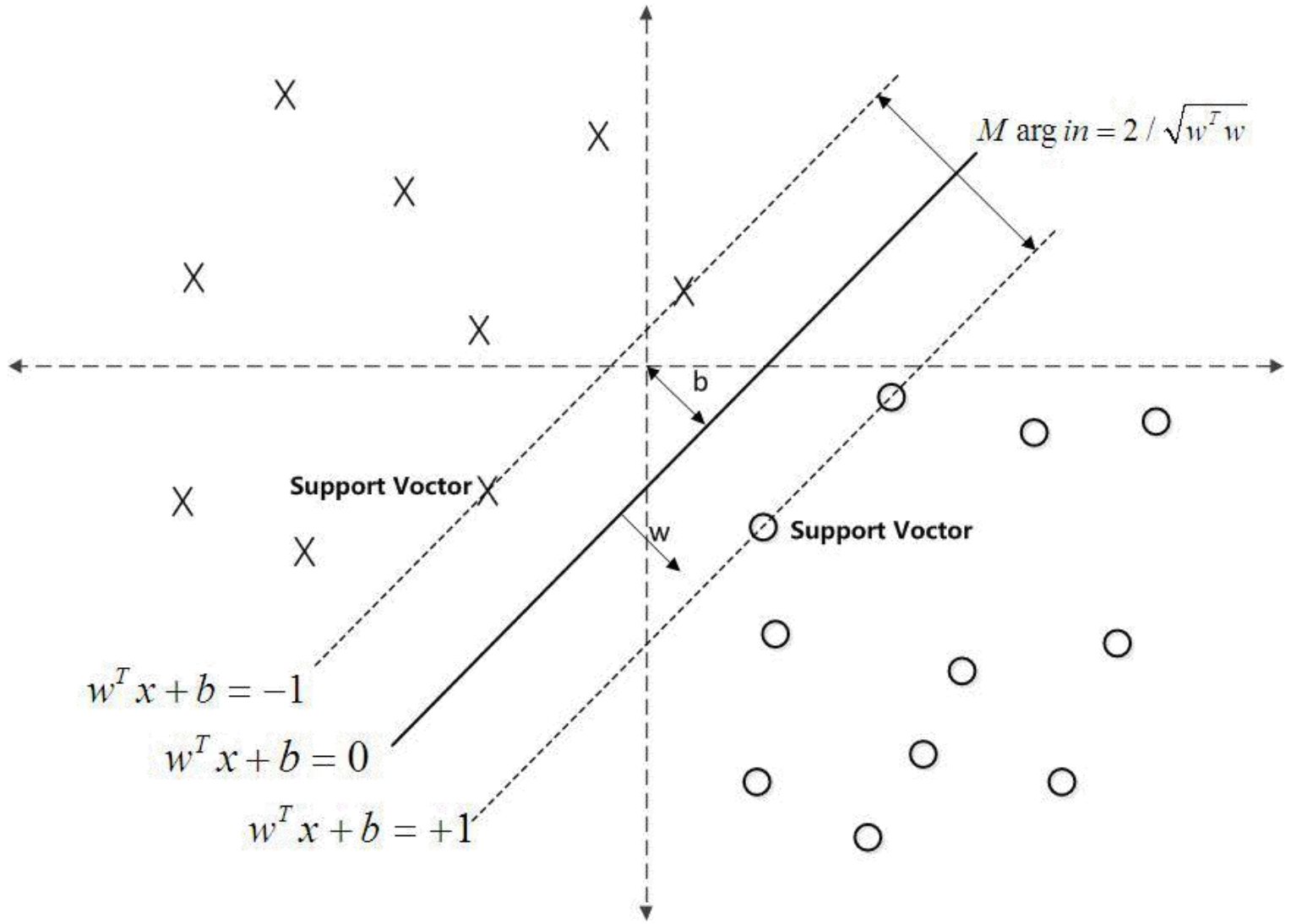
线性可分SVM

- 支持向量到超平面的距离为：

$$\because w^T x + b = \pm 1$$

$$\because y \in \{+1, -1\}$$

$$\therefore \frac{\left|\left(w^T x + b\right)\right|}{\|w\|_2} = \frac{1}{\|w\|_2}$$



- **备注：在SVM中支持向量到超平面的函数距离一般设置为1**

点到平面的距离：$d = \frac{|Ax_1 + By_1 + Cz_1 + D|}{\sqrt{A^2 + B^2 + C^2}}$

线性可分SVM

- SVM模型是让所有的分类点在各自类别的支持向量远离超平面的一侧，同时要求支持向量尽可能的远离这个超平面，用数学公式表示如下：

$$\max_{w,b} \frac{1}{\|w\|_2}$$

$$s.t: y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1, i = 1,2,\ldots,m$$

线性可分SVM

$$w^T = (w_1, w_2, \ldots, w_n)$$

$$\|w\|_2 = \sqrt{w_1^2 + w_2^2 + \ldots + w_n^2}$$

$$\max_{w,b} \frac{1}{\|w\|_2}$$

$$s.t: y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1, i = 1,2,\ldots,m$$

优化问题等价于 →

$$\min_{w,b} \frac{1}{2}\|w\|_2^2$$

$$s.t: y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1, i = 1,2,\ldots,m$$

线性可分SVM

- SVM目标函数/损失函数为：

$$J(w)=\frac{1}{2}\|w\|_2^2$$

$$s.t: y^{(i)}\left(w^T x^{(i)}+b\right)\geq 1, i=1,2,\ldots,m$$

$$\xrightarrow[\text{目标}]{\text{优化}} w^*, b^* = \min_{w,b} J(w)$$

线性可分SVM

- 将此时的目标函数和约束条件使用KKT条件转换为拉格朗日函数，从而转换为无约束的优化函数。

$$J(w)=\frac{1}{2}\|w\|_2^2 \quad s.t: y^{(i)}\left(w^T x^{(i)}+b\right)\geq 1, i=1,2,...,m \Rightarrow J(w)=\frac{1}{2}\|w\|_2^2 \quad s.t: 1-y^{(i)}\left(w^T x^{(i)}+b\right)\leq 0, i=1,2,...,m$$

$$\downarrow$$

$$L(w,b,\beta)=\frac{1}{2}\|w\|_2^2+\sum_{i=1}^{m}\beta_i\left[1-y^{(i)}\left(w^T x^{(i)}+b\right)\right], \beta_i\geq 0$$

线性可分SVM

- 引入拉格朗日乘子后，优化目标变成：$L(w,b,\beta)=\frac{1}{2}\|w\|_2^2+\sum_{i=1}^{m}\beta_i\left[1-y^{(i)}\left(w^T x^{(i)}+b\right)\right],\ \beta_i\geq 0$

- 根据拉格朗日对偶化特性，将该优化目标转换为等价的对偶问题来求解，从而优化目标变成：

- $$\min_{w,b}\max_{\beta\geq 0}L(w,b,\beta)$$

- 所以对于该优化函数而言，可以先求优化函数对于w和b的极小值，然后再求解对于拉格朗日乘子β的极大值。

$$\max_{\beta\geq 0}\min_{w,b}L(w,b,\beta)$$

线性可分SVM

$$\max_{\beta \geq 0} \min_{w,b} L(w,b,\beta)$$

$$L(w,b,\beta) = \frac{1}{2}\|w\|_2^2 + \sum_{i=1}^{m} \beta_i \left[1 - y^{(i)}\left(w^T x^{(i)} + b\right)\right], \beta_i \geq 0$$

- 首先求让函数L极小化的时候w和b的取值，这个极值可以直接通过对函数L分别求w和b的偏导数得到：

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w - \sum_{i=1}^{m} \beta_i y^{(i)} x^{(i)} = 0 \Rightarrow w = \sum_{i=1}^{m} \beta_i y^{(i)} x^{(i)}$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow -\sum_{i=1}^{m} \beta_i y^{(i)} = 0 \Rightarrow \sum_{i=1}^{m} \beta_i y^{(i)} = 0$$

线性可分SVM

$$L(w,b,\beta)=\frac{1}{2}\|w\|_2^2+\sum_{i=1}^{m}\beta_i\left[1-y^{(i)}\left(w^Tx^{(i)}+b\right)\right],\beta_i\geq 0$$

$$w=\sum_{i=1}^{m}\beta_i y^{(i)}x^{(i)} \qquad \sum_{i=1}^{m}\beta_i y^{(i)}=0$$

- 将求解出来的w和b带入优化函数L中，定义优化之后的函数如下：

$$\begin{aligned}
\ell(\beta)&=\frac{1}{2}\|w\|_2^2+\sum_{i=1}^{m}\beta_i\left[1-y^{(i)}\left(w^Tx^{(i)}+b\right)\right]\\
&=\frac{1}{2}\|w\|_2^2-\sum_{i=1}^{m}\beta_i\left[y^{(i)}\left(w^Tx^{(i)}+b\right)-1\right]\\
&=\frac{1}{2}w^Tw-\sum_{i=1}^{m}\beta_i y^{(i)}w^Tx^{(i)}-\sum_{i=1}^{m}\beta_i y^{(i)}b+\sum_{i=1}^{m}\beta_i\\
&=\frac{1}{2}w^T\sum_{i=1}^{m}\beta_i y^{(i)}x^{(i)}-w^T\sum_{i=1}^{m}\beta_i y^{(i)}x^{(i)}-b\sum_{i=1}^{m}\beta_i y^{(i)}+\sum_{i=1}^{m}\beta_i\\
&=-\frac{1}{2}w^T\sum_{i=1}^{m}\beta_i y^{(i)}x^{(i)}-b\sum_{i=1}^{m}\beta_i y^{(i)}+\sum_{i=1}^{m}\beta_i\\
&=-\frac{1}{2}\left(\sum_{j=1}^{m}\beta_j y^{(j)}x^{(j)}\right)^T\left(\sum_{i=1}^{m}\beta_i y^{(i)}x^{(i)}\right)+\sum_{i=1}^{m}\beta_i\\
&=-\frac{1}{2}\sum_{j=1}^{m}\beta_j y^{(j)}x^{(j)^T}\sum_{i=1}^{m}\beta_i y^{(i)}x^{(i)}+\sum_{i=1}^{m}\beta_i\\
&=\sum_{i=1}^{m}\beta_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\beta_i\beta_j y^{(i)}y^{(j)}x^{(j)^T}x^{(i)}
\end{aligned}$$

$$\Rightarrow \ell(\beta)=\sum_{i=1}^{m}\beta_i-\frac{1}{2}\sum_{i=1,j=1}^{m}\beta_i\beta_j y^{(i)}y^{(j)}x^{(i)^T}x^{(j)}$$

$$s.t:\sum_{i=1}^{m}\beta_i y^{(i)}=0$$

$$\beta_i\geq 0, i=1,2,\ldots,m$$

线性可分SVM

$$L(w,b,\beta)=\frac{1}{2}\|w\|_2^2+\sum_{i=1}^{m}\beta_i\left[1-y^{(i)}\left(w^T x^{(i)}+b\right)\right],\ \beta_i\geq 0$$

$$\ell(\beta)=\sum_{i=1}^{m}\beta_i-\frac{1}{2}\sum_{i=1,j=1}^{m}\beta_i\beta_j y^{(i)}y^{(j)}x^{(i)^T}x^{(j)} \qquad \max_{\beta\geq 0}\min_{w,b}L(w,b,\beta)$$

- 通过对w、b极小化后，我们最终得到的优化函数只和β有关，所以此时我们可以直接极大化我们的优化函数，得到β的值，从而可以最终得到w和b的值。

$$\max_{\beta\geq 0}\ell(\beta) \qquad \xrightarrow{\text{等价于}} \qquad \min_{\beta\geq 0}-\ell(\beta)$$

$$s.t:\sum_{i=1}^{m}\beta_i y^{(i)}=0 \qquad \xrightarrow{\text{等价于}} \qquad s.t:\sum_{i=1}^{m}\beta_i y^{(i)}=0$$

$$\downarrow$$

$$\min_{\beta\geq 0}\frac{1}{2}\sum_{i=1,j=1}^{m}\beta_i\beta_j y^{(i)}y^{(j)}x^{(i)^T}x^{(j)}-\sum_{i=1}^{m}\beta_i$$

$$s.t:\sum_{i=1}^{m}\beta_i y^{(i)}=0$$

- 备注：β值的求解使用SMO算法，后续课程中介绍。

线性可分SVM

- 假设存在最优解β^*； 根据w、b和β的关系，可以分别计算出对应的w值和b值(一般使用所有支持向量的计算均值来作为实际的b值)；

$$w^* = \sum_{i=1}^{m} \beta_i^* y^{(i)} x^{(i)}$$

- 这里的(x^s, y^s)即支持向量，根据KKT条件中的对偶互补条件(松弛条件约束)，支持向量必须满足一下公式：

$$y^s\left(w^T x^s + b\right) = y^s\left(\sum_{i=1}^{m} \beta_i^* y^{(i)} x^{(i)T} x^s + b\right) = 1 \Rightarrow b^* = y^s - \sum_{i=1}^{m} \beta_i^* y^{(i)} x^{(i)T} x^s$$

$$\beta_i\left(y^{(i)}\left(w^T x^{(i)} + b\right) - 1\right) = 0 \qquad \left\{\left(x^{(i)}, y^{(i)}\right) \middle| \beta_i > 0, i = 1,2,\ldots,m\right\}$$

怎么得到支持向量呢？

根据KKT条件中的对偶互补条件 $\beta_i\left(y^{(i)}\left(w^T x^{(i)} + b\right) - 1\right) = 0$，如果βi>0则有yi(wTxi+b)=1 即点在支持向量上，否则如果βi=0则有yi(wTxi+b)≥1，即样本在支持向量上或者已经被正确分类。

线性可分SVM算法流程

- 输入线性可分的m个样本数据{$(x^1,y^1),(x^2,y^2),...,(x^m,y^m)$}，其中x为n维的特征向量，y为二元输出，取值为+1或者-1；SVM模型输出为参数w、b以及分类决策函数。
  - 构造约束优化问题；

$$\min_{\beta \geq 0} \frac{1}{2} \sum_{i=1,j=1}^{m} \beta_i \beta_j y^{(i)} y^{(j)} x^{(i)^T} x^{(j)} - \sum_{i=1}^{m} \beta_i$$

$$s.t: \sum_{i=1}^{m} \beta_i y^{(i)} = 0$$

  - 使用SMO算法求出上式优化中对应的最优解β^*；
  - 找出所有的支持向量集合S；$S = \left\{ \left(x^{(i)}, y^{(i)} \right) \middle| \beta_i > 0, i = 1,2,...,m \right\}$
  - 更新参数w^*、b^*的值；
  - $$w^* = \sum_{i=1}^{m} \beta_i^* y^{(i)} x^{(i)} \qquad b^* = \frac{1}{S} \sum_{s=1}^{S} \left(y^s - \sum_{i=1}^{m} \beta_i^* y^{(i)} x^{(i)^T} x^s \right)$$
  - 构建最终的分类器

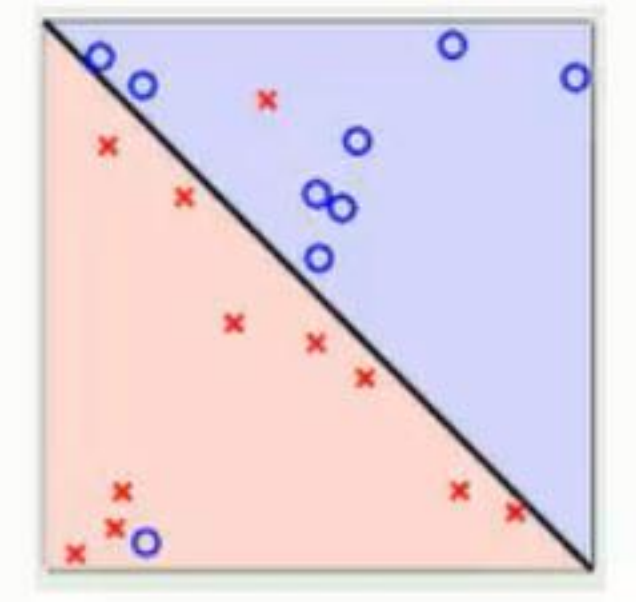
$$f(x) = sign\left(w^* \bullet x + b^* \right)$$

线性可分SVM总结

- 1. 要求数据必须是线性可分的；
- 2. 纯线性可分的SVM模型对于异常数据的预测可能会不太准；
- 3. 对于线性可分的数据，线性SVM分类器的效果非常不错。

SVM的软间隔模型

- 线性可分SVM中要求数据必须是线性可分的，才可以找到分类的超平面，但是有的时候线性数据集中存在少量的异常点，由于这些异常点导致了数据集不能够线性划分；直白来讲就是：正常数据本身是线性可分的，但是由于存在异常点数据，导致数据集不能够线性可分；



SVM的软间隔模型

- 如果线性数据中存在异常点导致没法直接使用SVM线性分割模型的时候，我们可以通过引入软间隔的概念来解决这个问题；
- **硬间隔**：可以认为线性划分SVM中的距离度量就是硬间隔，在线性划分SVM中，要求函数距离一定是大于1的，最大化硬间隔条件为：

$$\min_{w,b}\frac{1}{2}\|w\|_2^2;\ \ s.t: y^{(i)}\left(w^T x^{(i)}+b\right)\geq 1, i=1,2,\ldots,m$$

- **软间隔**：SVM对于训练集中的每个样本都引入一个松弛因子(ξ)，使得函数距离加上松弛因子后的值是大于等于1；这表示相对于硬间隔，对样本到超平面距离的要求放松了。

$$y^{(i)}\left(w^T x^{(i)}+b\right)\geq 1-\xi_i, i=1,2,\ldots,m, \xi_i\geq 0$$

SVM的软间隔模型

$$y^{(i)}\left(w^T x^{(i)}+b\right)\geq 1-\xi_i, i=1,2,...,m, \xi_i \geq 0$$

- 松弛因子(ξ)越大，表示样本点离超平面越近，如果松弛因子大于1，那么表示允许该样本点分错，所以说加入松弛因子是有成本的，过大的松弛因子可能会导致模型分类错误，所以最终的目标函数就转换成为：

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{m}\xi_i$$

$$s.t: y^{(i)}\left(w^T x^{(i)}+b\right)\geq 1-\xi_i, i=1,2,...,\ m$$

$$\xi_i \geq 0, i=1,2,...,\ m$$

- 备注：函数中的C>0是惩罚参数，是一个超参数，类似L1/L2 norm的参数；C越大表示对误分类的惩罚越大，也就是越不允许存在分错的样本；C越小表示对误分类的惩罚越小， 也就是表示允许更多的分错样本存在；C值的给定需要调参。

SVM的软间隔模型

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{m}\xi_i$$
$$s.t: y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 - \xi_i, i = 1,2,...,m$$
$$\xi_i \geq 0, i = 1,2,...,m$$

- 同线性可分SVM，根据KKT条件构造软间隔最大化的约束问题对应的拉格朗日函数如下：

$$L(w,b,\xi,\beta,\mu) = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{m}\xi_i + \sum_{i=1}^{m}\beta_i\left[1-\xi_i - y^{(i)}\left(w^T x^{(i)} + b\right)\right] - \sum_{i=1}^{m}\mu_i\xi_i, \beta_i \geq 0, \mu_i \geq 0$$

- 从而将我们的优化目标函数转换为：

$$\min_{w,b,\xi}\max_{\beta,\mu} L(w,b,\xi,\beta,\mu)$$

- 优化目标同样满足KKT条件，所以使用拉格朗日对偶将优化问题转换为等价的对偶问题：

$$\max_{\beta,\mu}\min_{w,b,\xi} L(w,b,\xi,\beta,\mu)$$

SVM的软间隔模型

- 先求优化函数对于w、b、ξ的极小值，这个可以通过分别对优化函数L求w、b、ξ的偏导数得，从而可以得到w、b、ξ关于β和μ之间的关系。

$$\frac{\partial L}{\partial w}=0\Rightarrow w-\sum_{i=1}^{m}\beta_i y^{(i)}x^{(i)}=0\Rightarrow w=\sum_{i=1}^{m}\beta_i y^{(i)}x^{(i)}$$

$$\frac{\partial L}{\partial b}=0\Rightarrow -\sum_{i=1}^{m}\beta_i y^{(i)}=0\Rightarrow \sum_{i=1}^{m}\beta_i y^{(i)}=0$$

$$\frac{\partial L}{\partial \xi_i}=0\Rightarrow C-\beta_i-\mu_i=0$$

SVM的软间隔模型

- 将w、b、ξ的值带入L函数中，就可以消去优化函数中的w、b、ξ，定义优化之后的函数如下：

$$
\begin{aligned}
\ell(\beta,\mu) &= \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{m}\xi_i + \sum_{i=1}^{m}\beta_i\left[1-\xi_i - y^{(i)}\left(w^T x^{(i)}+b\right)\right] - \sum_{i=1}^{m}\mu_i\xi_i \\
&= \frac{1}{2}\|w\|_2^2 - \sum_{i=1}^{m}\beta_i\left[y^{(i)}\left(w^T x^{(i)}+b\right)-1+\xi_i\right] + \sum_{i=1}^{m}(C-\mu_i)\xi_i \\
&= \frac{1}{2}\|w\|_2^2 - \sum_{i=1}^{m}\beta_i\left[y^{(i)}\left(w^T x^{(i)}+b\right)-1\right] \\
&= \frac{1}{2}w^T w - \sum_{i=1}^{m}\beta_i y^{(i)} w^T x^{(i)} - \sum_{i=1}^{m}\beta_i y^{(i)} b + \sum_{i=1}^{m}\beta_i \\
&= -\frac{1}{2}w^T\sum_{i=1}^{m}\beta_i y^{(i)} x^{(i)} - b\sum_{i=1}^{m}\beta_i y^{(i)} + \sum_{i=1}^{m}\beta_i \\
&= -\frac{1}{2}\left(\sum_{i=1}^{m}\beta_i y^{(i)} x^{(i)}\right)^T\left(\sum_{i=1}^{m}\beta_i y^{(i)} x^{(i)}\right) + \sum_{i=1}^{m}\beta_i \\
&= -\frac{1}{2}\sum_{i=1}^{m}\beta_i y^{(i)} x^{(i)^T}\sum_{i=1}^{m}\beta_i y^{(i)} x^{(i)} + \sum_{i=1}^{m}\beta_i \\
&= \sum_{i=1}^{m}\beta_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\beta_i\beta_j y^{(i)} y^{(j)} x^{(i)^T} x^{(j)}
\end{aligned}
$$

$$
\Rightarrow \ell(\beta) = \sum_{i=1}^{m}\beta_i - \frac{1}{2}\sum_{i=1,j=1}^{m}\beta_i\beta_j y^{(i)} y^{(j)} x^{(i)^T} x^{(j)}
$$

$$
s.t: \sum_{i=1}^{m}\beta_i y^{(i)} = 0
$$

$$
C - \beta_i - \mu_i = 0
$$

$$
\beta_i \geq 0, i = 1,2,\ldots,m
$$

$$
\mu_i \geq 0, i = 1,2,\ldots,m
$$

SVM的软间隔模型

- 最终优化后的目标函数/损失函数和线性可分SVM（硬间隔）模型基本一样，除了约束条件不同而已， 也就是说也可以使用SMO算法来求解。

$$\min \frac{1}{2} \sum_{i=1, j=1}^{m} \beta_i \beta_j y^{(i)} y^{(j)} x^{(i)^T} x^{(j)} - \sum_{i=1}^{m} \beta_i$$

$$s.t: \sum_{i=1}^{m} \beta_i y^{(i)} = 0$$

$$0 \le \beta_i \le C, i = 1,2,...,m$$

SVM的软间隔模型

- 在硬间隔最大化的时候，支持向量比较简单，就是离超平面的函数距离为1的样本点就是支持向量。
- 在软间隔中，根据KKT条件中的对偶互补条件: $\beta(1-\xi-y(wx+b))=0$和$\mu(-\xi)=0$，从而有:
  - 当$0<\beta_i\leq C$的时候，并且$\xi_i=0$的样本点均是支持向量(即所有的$0<\beta_i<C$)。即满足$|wx+b|=1$的所有样本均是支持向量。
  - 当$0<\beta_i<C$对应的样本就是支持向量。
  - 备注：软间隔和硬间隔中的支持向量的规则是一样的；

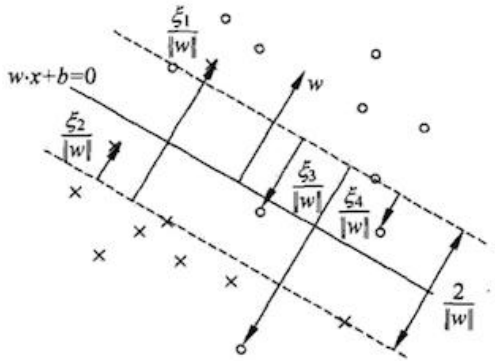


图 7.5　软间隔的支持向量

SVM的软间隔模型算法流程

- 输入线性可分的m个样本数据{$(x^1,y^1),(x^2,y^2),...,(x^m,y^m)$}，其中x为n维的特征向量，y为二元输出，取值为+1或者-1；SVM模型输出为参数w、b以及分类决策函数。
  - 选择一个惩罚系数C>0，构造约束优化问题；

$$\min \ \frac{1}{2}\sum_{i=1,j=1}^{m}\beta_i\beta_j y^{(i)}y^{(j)}x^{(i)^T}x^{(j)}-\sum_{i=1}^{m}\beta_i$$
$$s.t:\sum_{i=1}^{m}\beta_i y^{(i)}=0$$
$$0\le\beta_i\le C,i=1,2,...,m$$

  - 使用SMO算法求出上式优化中对应的最优解β^*；
  - 找出所有的支持向量集合S；

$$S=\left\{\left(x^{(i)},y^{(i)}\right)\middle|0<\beta_i<C,i=1,2,...,m\right\}$$

  - 更新参数w^*、b^*的值；

$$w^*=\sum_{i=1}^{m}\beta_i^* y^{(i)}x^{(i)}\qquad b^*=\frac{1}{S}\sum_{s=1}^{S}\left(y^s-\sum_{i=1}^{m}\beta_i^* y^{(i)}x^{(i)^T}x^s\right)$$

  - 构建最终的分类器

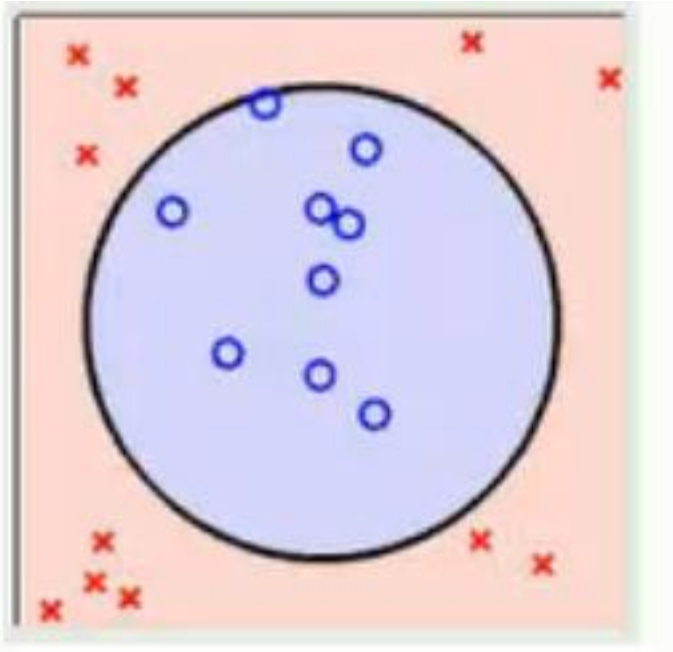
$$f(x)=sign\left(w^*\bullet x+b^*\right)$$

SVM的软间隔模型总结

- 1. 可以解决线性数据中携带异常点的分类模型构建的问题；
- 2. 通过引入惩罚项系数(松弛因子)，可以增加模型的泛化能力，即鲁棒性；
- 3. 如果给定的惩罚项系数C越小，表示在模型构建的时候，就允许存在越多的分类错误的样本， 也就表示此时模型的准确率会比较低；如果惩罚项系数越大，表示在模型构建的时候，就越不允许存在分类错误的样本，也就表示此时模型的准确率会比较高。

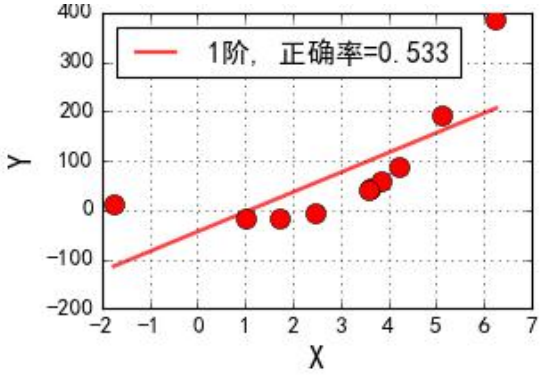
非线性可分SVM

- 不管是线性可分SVM还是加入惩罚系数后的软间隔线性可分SVM其实都要求数据本身是线性可分的，对于完全不可以线性可分的数据，这两种算法模型就没法解决这个问题了



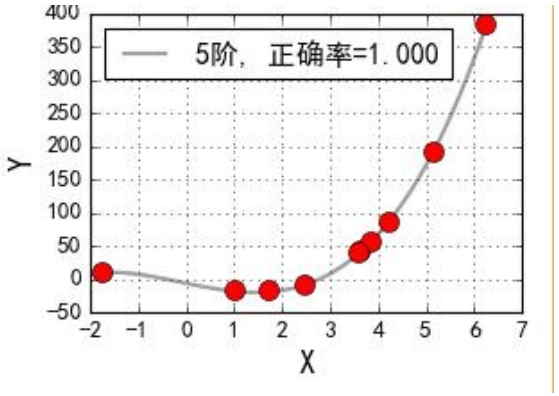
多项式回归回顾

- 在线性回归中，我们可以通过多项式扩展将低维度的数据扩展成为高维度的数据，从而可以使用线性回归模型来解决问题。也就是说对于二维空间中不是线性可分的数据，将其映射到高维空间中后，变成了线性可分的数据。



两维线性模型：$h_\theta(x_1,x_2)=\theta_0+\theta_1x_1+\theta_2x_2$

$$(x_1,x_2)\xrightarrow{\text{多项式扩展}}(x_1,x_2,x_1^2,x_2^2,x_1x_2)$$



五维线性模型：

$$h_\theta(x_1,x_2)=\theta_0+\theta_1x_1+\theta_2x_2+\theta_3x_1^2+\theta_4x_2^2+\theta_5x_1x_2$$

$$\xrightarrow{\text{等价}}h_\theta(x_1,x_2)=\theta_0+\theta_1z_1+\theta_2z_2+\theta_3z_3+\theta_4z_4+\theta_5z_5$$

非线性可分SVM

- 结合多项式回归在处理非线性可分数据时候的作用，在SVM的线性不可分的数据上，如果将数据映射到高维空间中，那么数据就会变成线性可分的，从而就可以使用线性可分SVM模型或者软间隔线性可分SVM模型。
- 也就是说，对于线性不可分SVM模型来讲，重点在于低维特征数据到高维特征数据之间的映射。

非线性可分SVM

- 定义一个从低维特征空间到高维特征空间的映射函数Φ，非线性可分SVM的优化目标函数：

$$\min \frac{1}{2} \sum_{i=1,j=1}^{m} \beta_i \beta_j y^{(i)} y^{(j)} \phi\left(x^{(i)}\right) \bullet \phi\left(x^{(j)}\right) - \sum_{i=1}^{m} \beta_i$$

$$s.t: \quad \sum_{i=1}^{m} \beta_i y^{(i)} = 0$$

$$0 \le \beta_i \le C, i = 1,2,...,m$$

- 可以看到的是，只需要将原来的低维空间中的两个向量的点积转换为高维空间中两个向量的点积即可。

非线性可分SVM

- 这样一来问题就解决了吗？似乎是的：拿到非线性数据，就找一个映射，然后一股脑把原来的数据映射到新空间中，再做线性 SVM 即可。不过事实上没有这么简单！其实刚才的方法稍想一下就会发现有问题：在最初的例子里做了一个二阶多项式的转换，对一个二维空间做映射，选择的新空间是原始空间的所有一阶和二阶的组合，得到了5个维度；如果原始空间是三维，那么我们会得到9维的新空间；如果原始空间是n维，那么我们会得到一个n(n+3)/2维的新空间；这个数目是呈爆炸性增长的，这给计算带来了非常大的困难，而且如果遇到无穷维的情况，就根本无从计算。

核函数

- 假设函数Φ是一个从低维特征空间到高维特征空间的一个映射，那么如果存在函数K(x,z), 对于任意的低维特征向量x和z，都有：

$$\mathrm{K}(x,z)=\phi(x)\bullet\phi(z)$$

- 称函数K(x,z)为**核函数(kernal function)**；
- 核函数：在低维空间上的计算量等价于特征做维度扩展后的点乘的结果。

核函数

- 核函数在解决线性不可分问题的时候，采取的方式是：使用低维特征空间上的计算来避免在高维特征空间中向量内积的恐怖计算量；也就是说此时SVM模型可以应用在高维特征空间中数据可线性分割的优点，同时又避免了引入这个高维特征空间恐怖的内积计算量。
- 即：用低维空间中少的内积的计算量来让模型具有高维空间中的线性可分的优点。

核函数

- 不妨还是从最开始的简单例子出发，设两个向量 $x_1 = (\mu_1, \mu_2)^T$ 和 $x_2 = (\eta_1, \eta_2)^T$，而即是到前面说的五维空间的映射，因此映射过后的内积为：

- 而同时我们可以发现有一下公式：$$\phi(x_1) \bullet \phi(x_2) = \mu_1\eta_1 + \mu_2\eta_2 + \mu_1^2\eta_1^2 + \mu_2^2\eta_2^2 + \mu_1\mu_2\eta_1\eta_2$$

- 可以发现两者之间非常相似，所以我们只要乘上一个相关的系数，就可以让这两个式子的值相等，这样不就将五维空间的一个内积转换为两维空间的内积的运算。

$$(x_1 \bullet x_2 + 1)^2 = 2\mu_1\eta_1 + 2\mu_2\eta_2 + \mu_1^2\eta_1^2 + \mu_2^2\eta_2^2 + 2\mu_1\mu_2\eta_1\eta_2 + 1$$

核函数

- 现有有两个两维的向量，进行二阶多项式扩展，然后进行内积计算，这个时候映射高维后计算的计算量为：11次乘法+4次加法；采用近似计算的计算量为：3次乘法+2次加法；采用加系数后的近似计算的计算量为：4次乘法+2次加法；

x_1	3	5
x_2	4	2

z_1	3	5	3*3=9	5*5=25	3*5=15
z_2	4	2	4*4=16	2*2=4	4*2=8

$$\phi(x_1)\bullet\phi(x_2)=z_1\bullet z_2=3*4+2*5+9*16+4*25+8*15=386$$

$$(x_1\bullet x_2+1)^2=(3*4+2*5+1)^2=23*23=529$$

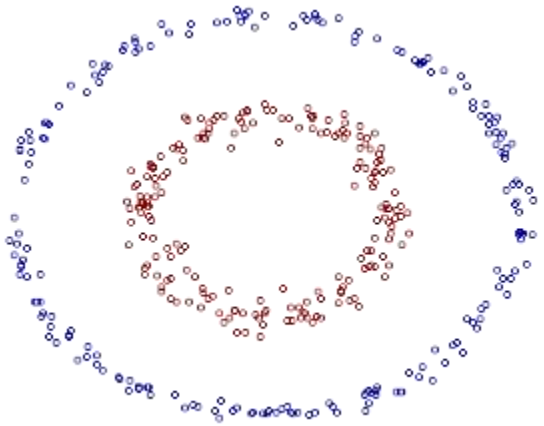
$$(0.8476x_1\bullet x_2+1)^2=(0.8476*(3*4+2*5)+1)^2=18.6472*18.6472\approx 386.01$$

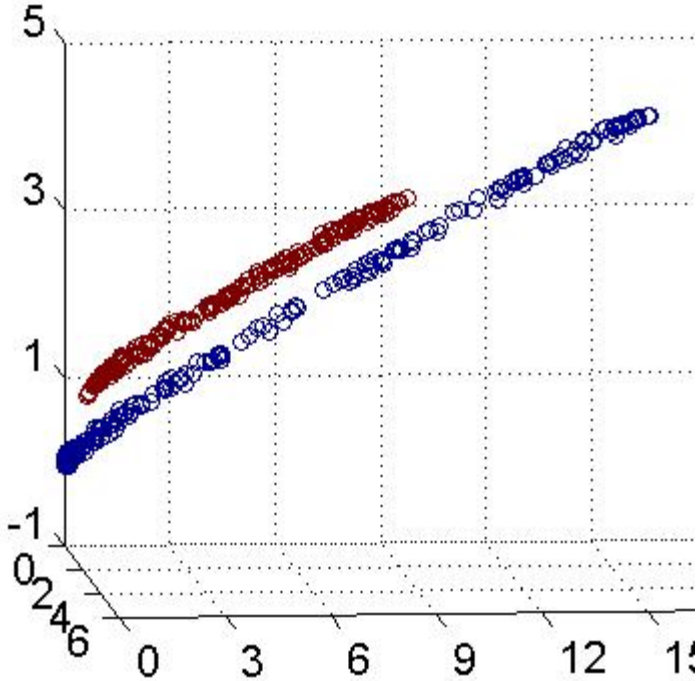
核函数

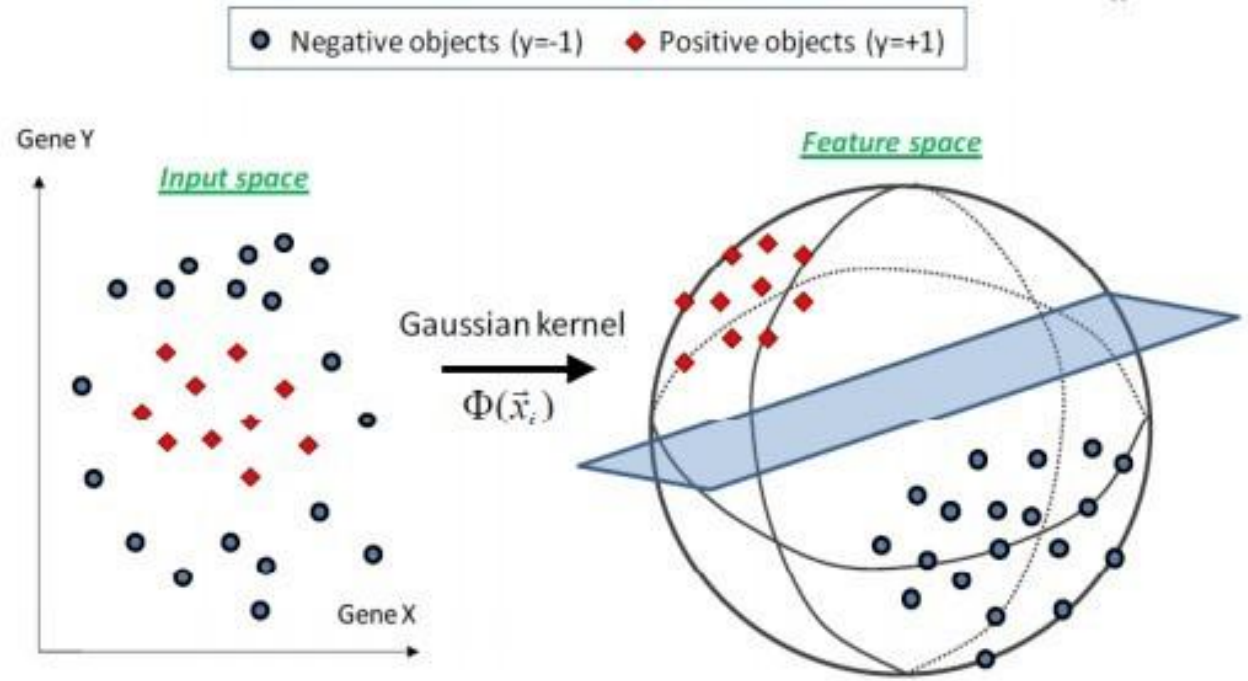
- 线性核函数(Linear Kernel): $$\mathrm{K}(x,z)=x\bullet z$$
- 多项式核函数(Polynomial Kernel)：其中γ、r、d属于超参，需要调参定义；
- $$\mathrm{K}(x,z)=(\gamma x\bullet z+r)^d$$
- 高斯核函数(Gaussian Kernel)：其中γ属于超参，要求大于0，需要调参定义；
- $$\mathrm{K}(x,z)=e^{-\gamma\|x-z\|_2^2}$$
- Sigmoid核函数(Sigmoid Kernel)：其中γ、r属于超参，需要调参定义；

$$\mathrm{K}(x,z)=\tanh(\gamma x\bullet z+r)$$

核函数




5
3
1
-1
0
2
4
6
0
3
6
9
12


Negative objects (y=-1)
Positive objects (y=+1)
Gene Y
Input space
Gene X
Gaussian kernel
$\Phi(\vec{x}_i)$
Feature space

核函数总结

- 1. 核函数可以自定义；核函数必须是正定核函数，即Gram矩阵是半正定矩阵；
- 2. 核函数的价值在于它虽然也是将特征进行从低维到高维的转换，但核函数它事先在低维上进行计算，而将实质上的分类效果表现在了高维上，也就如上文所说的避免了直接在高维空间中的复杂计算；
- 3. 通过核函数，可以将非线性可分的数据转换为线性可分数据；

$$Gram = \begin{bmatrix} \mathrm{K}(x_1, x_1) & \mathrm{K}(x_1, x_2) & \dots & \mathrm{K}(x_1, x_m) \\ \mathrm{K}(x_2, x_1) & \mathrm{K}(x_2, x_2) & \dots & \mathrm{K}(x_2, x_m) \\ \dots & \dots & \dots & \dots \\ \mathrm{K}(x_m, x_1) & \mathrm{K}(x_m, x_2) & \dots & \mathrm{K}(x_m, x_m) \end{bmatrix}$$

scitit-learn SVM算法库概述

- scikit-learn中SVM的算法库主要分为两类，一类是分类算法，包括：SVC、NuSVC和LinearSVC这三个类，另外一类是回归算法，包括：SVR、NuSVR和LinearSVR这三个类；除此之外，用的比较多的SVM模型还有OneClassSVM类(主要功能是：异常点检测)。
- 详见：http://scikit-learn.org/0.18/modules/classes.html#module-sklearn.svm

Estimators

svm.SVC ([C, kernel, degree, gamma, coef0, ...])	C-Support Vector Classification.
svm.LinearSVC ([penalty, loss, dual, tol, C, ...])	Linear Support Vector Classification.
svm.NuSVC ([nu, kernel, degree, gamma, ...])	Nu-Support Vector Classification.
svm.SVR ([kernel, degree, gamma, coef0, tol, ...])	Epsilon-Support Vector Regression.
svm.LinearSVR ([epsilon, tol, C, loss, ...])	Linear Support Vector Regression.
svm.NuSVR ([nu, C, kernel, degree, gamma, ...])	Nu Support Vector Regression.
svm.OneClassSVM ([kernel, degree, gamma, ...])	Unsupervised Outlier Detection.

scitit-learn SVM算法库概述
分类算法

参数	LinearSVC	SVC	NuSVC
C	惩罚项系数，默认为1，一般通过交叉验证给定适合值		不支持
nu	不支持		表示训练集上错误率的上限，默认为0.5，取值范围(0,1]，功能和C类似
kernel	不支持，只能使用线性核函数	给定具体的核函数或者Gram矩阵，参数可选:linear、rbf、poly、sigmoid、precomputed	
penalty	对线性拟合有意义，参数可选l1和l2	不支持	
degree	不支持	多项式核函数中，d的参数值，一般需要通过交叉验证给定	
gamma	不支持	rbf、poly和sigmoid核函数中的参数值γ，一般需要通过交叉验证给定	
coef0	不支持	poly和sigmoid核函数中的参数值r，一般需要通过交叉验证给定	

scitit-learn SVM算法库概述
分类算法

参数	LinearSVC	SVC	NuSVC
class_weight	样本类别权重，主要是为了防止训练集中某些类别的样本过多，导致训练出来的决策偏向这些类别。参数可选：权重列表、balanced，如果是balanced的话，表示算法会自己计算权重，样本量少的类别权重会高；如果样本没有比较明显的偏重的话，一般不用管这个参数，默认为None		
decision_function_shape	不支持该参数，使用multi_class代替	指定进行多元分类的时候，进行如何操作：可选ovo和ovr；推荐选择：ovr	
multi_class	指定进行多元分类的时候，进行如何构建操作，可选ovr和crammer_singer；一般情况为ovr；(crammer_singer为ovr的优化，但是实际效果不好)	不支持该参数，使用decision_function_shape代替	
cache_size	不支持	在大样本的时候，缓存会影响模型的构建速度的；当机器内存比较大的时候，可以考虑将该值设置为500或者1000，默认为200(单位为M)	

scitit-learn SVM算法库概述

- OneClassSVM是一种异常点检测的算法，算法原理：先使用SVM算法，对正常数据进行模型训练，找出正常数据的数据特性(eg: 数据集中在哪儿？？)，也就是找出数据中分割面(也就是正常数据的边界)，以边界作为算法找到的支持向量。预测的时候，如果样本点落在支持向量内部，那就认为属于正常样本，否则就认为属于异常样本。

参数	OneClassSVM
kernel	核函数，可选:‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’
nu	控制分类错误的样本百分比上限，或者说控制支持向量个数的下限，默认为0.5，可取值范围为:(0,1]
degree	多项式核函数超参，一般通过交叉验证给定
gamma	rbf、poly、sigmoid核函数超参，一般通过交叉验证给定
coef0	poly、sigmoid核函数超参，一般通过交叉验证给定
cache_size	在大样本的时候，缓存会影响模型的构建速度的；当机器内存比较大的时候，可以考虑将该值设置为500或者1000，默认为200(单位为M)

SMO

- **序列最小优化算法**(Sequential minimal optimization, **SMO**)是一种用于解决SVM训练过程中所产生的优化问题的算法。 于1998年由John Platt发明，论文详见：Sequencial Minimal Optimization-a Fast Alg for Training SVM.pdf

$$\min_{\beta} \frac{1}{2}\sum_{i=1,j=1}^{m}\beta_i\beta_j y^{(i)}y^{(j)}\mathrm{K}\left(x^{(i)},x^{(j)}\right)-\sum_{i=1}^{m}\beta_i$$

$$s.t: \quad \sum_{i=1}^{m}\beta_i y^{(i)}=0$$

$$0\le\beta_i\le C, i=1,2,\ldots m$$

SMO

- 假定存在一个$\beta^*=(\beta_1,\beta_2,\ldots,\beta_m)$是我们最终的最优解，那么根据KKT条件我们可以计算出w和b的最优解，如下：

$$w=\sum_{i=1}^{m}\beta_i y^{(i)}x^{(i)} \qquad b=y-\sum_{i=1}^{m}\beta_i y^{(i)}\mathrm{K}\left(x^{(i)},x\right)$$

- 进而我们可以得到最终的分离超平面为：

$$g(x)=wx+b=\sum_{i=1}^{m}\beta_i y^{(i)}\mathrm{K}\left(x^{(i)},x\right)+b$$

SMO

- 拉格朗日乘子法和KKT的对偶互补条件为：

$$\beta_i\left(y^{(i)}\left(w^T x^{(i)}+b\right)-1+\xi_i\right)=0 \qquad \mu_i\xi_i=0$$

- β、μ和C之间的关系为：

$$C-\beta_i-\mu_i=0$$

- 根据这个对偶互补条件，我们有如下关系式：

$$\beta_i=0\Rightarrow\mu_i>0\Rightarrow\xi_i=0\Rightarrow y^{(i)}\left(w^T x^{(i)}+b\right)>1$$

$$0<\beta_i<C\Rightarrow\mu_i>0\Rightarrow\xi_i=0\Rightarrow y^{(i)}\left(w^T x^{(i)}+b\right)=1$$

$$\beta_i=C\Rightarrow\mu_i=0\Rightarrow\xi_i>0\Rightarrow y^{(i)}\left(w^T x^{(i)}+b\right)<1$$

$$\Longrightarrow\quad y^{(i)}g\left(x^{(i)}\right)=\begin{cases}>1, \left\{\left(x^{(i)},y^{(i)}\right)\middle|\beta_i=0\right\}\\ =1, \left\{\left(x^{(i)},y^{(i)}\right)\middle|0<\beta_i<C\right\}\\ <1, \left\{\left(x^{(i)},y^{(i)}\right)\middle|\beta_i=C\right\}\end{cases}$$

SMO

- 也就是说我们找出的最优的分割超平面必须满足下列的**目标条件**(g(x)):

$$y^{(i)}g\left(x^{(i)}\right)=\begin{cases}>1, \left\{\left(x^{(i)},y^{(i)}\right)\middle|\beta_i=0\right\}\\ =1, \left\{\left(x^{(i)},y^{(i)}\right)\middle|0<\beta_i<C\right\}\\ <1, \left\{\left(x^{(i)},y^{(i)}\right)\middle|\beta_i=C\right\}\end{cases}$$

- 拉格朗日对偶化要求的两个**限制的初始条件**为:

$$\sum_{i=1}^{m}\beta_i y^{(i)}=0$$

$$0\le\beta_i\le C, i=1,2,\ldots,m$$

SMO

- 从而可以得到解决问题的思路如下：
  - 首先，初始化后一个β值，让它满足对偶问题的两个**初始限制条件**；
  - 然后不断优化这个β值，使得由它确定的分割超平面满足**g(x)目标条件**；而且在优化过程中，始终保证β值满足**初始限制条件**。
  - 备注：这个求解过程中，和传统的思路不太一样，不是对目标函数求最小值，而是让**g(x)目标条件**尽可能的满足。

SMO

- 在这样一个过程中，到底如何优化这个β值呢？？？整理可以发现β值的优化必须遵循以下两个基本原则：
  - 每次优化的时候，必须同时优化β的两个分量；因为如果只优化一个分量的话，新的β值就没法满足**初始限制条件**中的**等式约束条件**了。
  - 每次优化的两个分量应该是违反**g(x)目标条件**比较多的。也就是说，本来应当是大于等于1的，越是小于1违反**g(x)目标条件**就越多。
- 或者换一种思路来理解，因为目标函数中存在m个变量，直接优化比较难，利用启发式的方法/EM算法的思想，每次优化的时候，只优化两个变量，将其它的变量看成常数项，这样SMO算法就将一个复杂的优化算法转换为一个比较简单的两变量优化问题了。

SMO

$$\min \frac{1}{2}\sum_{i=1,j=1}^{m}\beta_i\beta_j y^{(i)}y^{(j)}\mathrm{K}\left(x^{(i)},x^{(j)}\right)-\sum_{i=1}^{m}\beta_i$$

$$s.t:\ \sum_{i=1}^{m}\beta_i y^{(i)}=0$$

$$0\le\beta_i\le C, i=1,2,...,m$$

$$\mathrm{K}_{ij}=\mathrm{K}\left(x^{(i)},x^{(j)}\right)$$

- 认为β_1、β_2是变量，其它β值是常量，从而将目标函数转换如下(C是常数项)：

$$\min_{\beta_1,\beta_2}\frac{1}{2}\mathrm{K}_{11}\beta_1^2+\frac{1}{2}\mathrm{K}_{22}\beta_2^2+\mathrm{K}_{12}y^{(1)}y^{(2)}\beta_1\beta_2-\beta_1-\beta_2+y^{(1)}\beta_1\sum_{i=3}^{m}y^{(i)}\beta_i\mathrm{K}_{i1}+y^{(2)}\beta_2\sum_{i=3}^{m}y^{(i)}\beta_i\mathrm{K}_{i2}+C$$

$$s.t:\beta_1y^{(1)}+\beta_2y^{(2)}=-\sum_{i=3}^{m}\beta_iy^{(i)}=k$$

$$0\le\beta_i\le C,i=1,2$$

SMO

- 由于 $\beta_1 y^{(1)} + \beta_2 y^{(2)} = k$，并且$y^2=1$，也就是我们使用$\beta_2$来表示$\beta_1$的值：

$$\beta_1 = y^{(1)}\left(k - \beta_2 y^{(2)}\right)$$

- 将上式带入目标优化函数，就可以消去β_1，从而只留下仅仅包含β_2的式子。

$$令：\nu_j = \sum_{i=3}^{m} y^{(i)} \beta_i \mathrm{K}_{ij}$$

$$W\left(\beta_2\right) = \frac{1}{2}\mathrm{K}_{11}\left(k - \beta_2 y^{(2)}\right)^2 + \frac{1}{2}\mathrm{K}_{22}\beta_2^2 + \mathrm{K}_{12} y^{(2)}\left(k - \beta_2 y^{(2)}\right)\beta_2$$
$$- y^{(1)}\left(k - \beta_2 y^{(2)}\right) - \beta_2 + \left(k - \beta_2 y^{(2)}\right)\nu_1 + y^{(2)}\beta_2\nu_2$$

SMO
求解β_2的值

$$g(x)=\sum_{i=1}^{m}\beta_i y^{(i)}\mathrm{K}\left(x^{(i)},x\right)+b$$

$$v_j=\sum_{i=3}^{m}y^{(i)}\beta_i\mathrm{K}_{ij}$$

$$\frac{\partial W}{\partial\beta_2}=\mathrm{K}_{11}\beta_2-\mathrm{K}_{11}ky^{(2)}+\mathrm{K}_{22}\beta_2+\mathrm{K}_{12}ky^{(2)}-2\mathrm{K}_{12}\beta_2+y^{(1)}y^{(2)}-1-y^{(2)}v_1+y^{(2)}v_2$$

$$\xrightarrow{令\frac{\partial W}{\partial\beta_2}=0}$$

$$\left(\mathrm{K}_{11}+\mathrm{K}_{22}-2\mathrm{K}_{12}\right)\beta_2=y^{(2)}\left(y^{(2)}-y^{(1)}+k\mathrm{K}_{11}-k\mathrm{K}_{12}+v_1-v_2\right)$$

$$=y^{(2)}\left(y^{(2)}-y^{(1)}+k\mathrm{K}_{11}-k\mathrm{K}_{12}+\left(g(x_1)-\sum_{i=1}^{2}\beta_i y^{(i)}K_{i1}-b\right)-\left(g(x_2)-\sum_{i=1}^{2}\beta_i y^{(i)}K_{i2}-b\right)\right)$$

SMO
求解β_2的值

将$k=\beta_1 y^{(1)}+\beta_2 y^{(2)}$
带入上页ppt中式子

$$\left(\mathrm{K}_{11}+\mathrm{K}_{22}-2\mathrm{K}_{12}\right)\beta_2=$$

$$=y^{(2)}\left(y^{(2)}-y^{(1)}+\beta_1^{old}y^{(1)}\mathrm{K}_{11}+\beta_2^{old}y^{(2)}\mathrm{K}_{11}-\beta_1^{old}y^{(1)}\mathrm{K}_{12}-\beta_2^{old}y^{(2)}\mathrm{K}_{12}\right)$$

$$+y^{(2)}\left(g\left(x_1\right)-\sum_{i=1}^{2}\beta_i^{old}y^{(i)}K_{i1}-b\right)-y^{(2)}\left(g\left(x_2\right)-\sum_{i=1}^{2}\beta_i^{old}y^{(i)}K_{i2}-b\right)$$

$$=y^{(2)}\left(\left(\mathrm{K}_{11}-2\mathrm{K}_{12}+\mathrm{K}_{22}\right)\beta_2^{old}y^{(2)}+y^{(2)}-y^{(1)}+g\left(x^{(1)}\right)-g\left(x^{(2)}\right)\right)$$

$$=\left(\mathrm{K}_{11}-2\mathrm{K}_{12}+\mathrm{K}_{22}\right)\beta_2^{old}+y^{(2)}\left[\left(g\left(x^{(1)}\right)-y^{(1)}\right)-\left(g\left(x^{(2)}\right)-y^{(2)}\right)\right]$$

SMO算法

$$\left(\mathrm{K}_{11}+\mathrm{K}_{22}-2\mathrm{K}_{12}\right)\beta_2=$$

$$=\left(\mathrm{K}_{11}-2\mathrm{K}_{12}+\mathrm{K}_{22}\right)\beta_2^{old}+y^{(2)}\left[\left(g\left(x^{(1)}\right)-y^{(1)}\right)-\left(g\left(x^{(2)}\right)-y^{(2)}\right)\right]$$

$$\xrightarrow{令E_i=g\left(x^{(i)}\right)-y^{(i)}}$$

$$\beta_2^{new,unt}=\beta_2=\beta_2^{old}+\frac{y^{(2)}\left(E_1-E_2\right)}{\mathrm{K}_{11}+\mathrm{K}_{22}-2\mathrm{K}_{12}}$$

SMO

$$\beta_1 y^{(1)} + \beta_2 y^{(2)} = -\sum_{i=3}^{m} \beta_i y^{(i)} = k$$

$$0 \le \beta_i \le C, i = 1,2$$

- 考虑β_1和β_2的取值限定范围，假定新求出来的β值是满足我们的边界限制的，即如下所示：

$$L \le \beta_2^{new} \le H$$

- 当$y_1=y_2$的时候，$\beta_1+\beta_2=k$； 由于β的限制条件，我们可以得到：

$$\begin{cases} 0 \le k - \beta_2^{new} \le C \\ 0 \le \beta_2^{new} \le C \end{cases} \Rightarrow \begin{cases} L = \max\left(0, \beta_2^{old} + \beta_1^{old} - C\right) \\ H = \min\left(C, \beta_2^{old} + \beta_1^{old}\right) \end{cases}$$

- 当$y_1 \neq y_2$的时候，$\beta_1-\beta_2=k$；由于β的限制条件，我们可以得到：

$$\begin{cases} 0 \le k + \beta_2^{new} \le C \\ 0 \le \beta_2^{new} \le C \end{cases} \Rightarrow \begin{cases} L = \max\left(0, \beta_2^{old} - \beta_1^{old}\right) \\ H = \min\left(C, C + \beta_2^{old} - \beta_1^{old}\right) \end{cases}$$

SMO

- 结合β的取值限制范围以及函数W的β最优解，我们可以得带迭代过程中的最优解为：

$$\beta_2^{new} = \begin{cases} H; \beta_2^{new,unt} > H \\ \beta_2^{new,unt}; L \le \beta_2^{new,unt} \le H \\ L; \beta_2^{new,unt} < L \end{cases}$$

- 然后根据β_1和β_2的关系，从而可以得到迭代后的β_1的值：

$$\beta_1^{old} y^{(1)} + \beta_2^{old} y^{(2)} = \beta_1^{new} y^{(1)} + \beta_2^{new} y^{(2)}$$

$$\Rightarrow \beta_1^{new} = \beta_1^{old} + y^{(1)} y^{(2)} \left(\beta_2^{old} - \beta_2^{new} \right)$$

SMO

- 求解β的过程中，相关公式如下：

$$\mathrm{K}_{ij} = \mathrm{K}\left(x^{(i)}, x^{(j)}\right) \qquad E_i = g\left(x^{(i)}\right) - y^{(i)} = \sum_{j=1}^{m} \beta_j y^{(j)} \mathrm{K}_{ij} + b - y^{(i)}$$

$$\beta_2^{new,unt} = \beta_2^{old} + \frac{y^{(2)}\left(E_1 - E_2\right)}{\mathrm{K}_{11} + \mathrm{K}_{22} - 2\mathrm{K}_{12}}$$

$$\beta_2^{new} = \begin{cases} H; \beta_2^{new,unt} > H \\ \beta_2^{new,unt}; L \le \beta_2^{new,unt} \le H \\ L; \beta_2^{new,unt} < L \end{cases}$$

$$\beta_1^{new} = \beta_1^{old} + y^{(1)} y^{(2)} \left(\beta_2^{old} - \beta_2^{new}\right)$$

SMO

- 可以发现SMO算法中，是选择两个合适的β变量做迭代，其它变量作为常量来进行优化的一个过程，那么这两个变量到底怎么选择呢???
  - 每次优化的时候，必须同时优化β的两个分量；因为如果只优化一个分量的话，新的β值就没法满足**初始限制条件**中的**等式约束条件**了。
  - 每次优化的两个分量应该是违反**g(x)目标条件**比较多的。也就是说，本来应当是大于等于1的，越是小于1违反**g(x)目标条件**就越多。

SMO
第一个β变量的选择

- SMO算法在选择第一个β变量的时候，需要选择在训练集上违反KKT条件最严重的样本点。一般情况下，先选择0<β<C的样本点(即支持向量)，只有当所有的支持向量都满足KKT条件的时候，才会选择其它样本点。因为此时违反KKT条件越严重，在经过一次优化后，会让变量β尽可能的发生变化，从而可以以更少的迭代次数让模型达到**g(x)目标条件**。

$$y^{(i)}g\left(x^{(i)}\right)=\begin{cases}>1, \left\{\left(x^{(i)},y^{(i)}\right)\middle|\beta_i=0\right\}\\ =1, \left\{\left(x^{(i)},y^{(i)}\right)\middle|0<\beta_i<C\right\}\\ <1, \left\{\left(x^{(i)},y^{(i)}\right)\middle|\beta_i=C\right\}\end{cases}$$

SMO
第二个β变量的选择

- 在选择第一个变量β_1后，在选择第二个变量β_2的时候，希望能够按照优化后的β_1和β_2有尽可能多的改变来选择，也就是说让$|E_1-E_2|$足够的大，当E_1为正的时候，选择最小的E_i作为E_2；当E_1为负的时候，选择最大的Ei作为E_2。
- 备注：如果选择的第二个变量不能够让目标函数有足够的下降，那么可以通过遍历所有样本点来作为β_2，直到目标函数有足够的下降，如果都没有足够的下降的话，那么直接跳出循环，重新选择β_1；

SMO
计算阈值b和差值E_i

- 在每次完成两个β变量的优化更新之后，需要重新计算阈值b和差值E_i。当$0<\beta_1^{new}<C$时，有：

$$y^{(1)}-\sum_{i=1}^{m}\beta_i y^{(i)}\mathrm{K}_{i1}-b_1=0$$

- 化简可得：

$$b_1^{new}=y^{(1)}-\sum_{i=3}^{m}\beta_i y^{(i)}\mathrm{K}_{i1}-\beta_1^{new}y^{(1)}\mathrm{K}_{11}-\beta_2^{new}y^{(2)}\mathrm{K}_{12}$$

$$\because E_1=g\left(x^{(1)}\right)-y^{(1)}=\sum_{j=3}^{m}\beta_j y^{(j)}\mathrm{K}_{j1}+\beta_1^{old}y^{(1)}\mathrm{K}_{11}+\beta_2^{old}y^{(2)}\mathrm{K}_{12}+b^{old}-y^{(1)}$$

$$\therefore b_1^{new}=-E_1-y^{(1)}\mathrm{K}_{11}\left(\beta_1^{new}-\beta_1^{old}\right)-y^{(2)}\mathrm{K}_{12}\left(\beta_2^{new}-\beta_2^{old}\right)+b^{old}$$

SMO
计算阈值b和差值E_i

$$b_1^{new} = -E_1 - y^{(1)}\mathrm{K}_{11}\left(\beta_1^{new} - \beta_1^{old}\right) - y^{(2)}\mathrm{K}_{12}\left(\beta_2^{new} - \beta_2^{old}\right) + b^{old}$$

- 同样的当β_2的取值为: $0<\beta_2<C$的时候，我们也可以得到

$$b_2^{new} = -E_2 - y^{(1)}\mathrm{K}_{12}\left(\beta_1^{new} - \beta_1^{old}\right) - y^{(2)}\mathrm{K}_{22}\left(\beta_2^{new} - \beta_2^{old}\right) + b^{old}$$

- 最终计算出来的b为:

$$b^{new} = \frac{b_1^{new} + b_2^{new}}{2}$$

SMO
计算阈值b和差值E_i

- 当更新计算阈值b后，就可以得到差值E_i为：

$$E_i = g\left(x^{(i)}\right) - y^{(i)} = \sum_{j=3}^{m} \beta_j y^{(j)} \mathrm{K}_{ij} + \beta_1^{new} y^{(1)} \mathrm{K}_{11} + \beta_2^{new} y^{(2)} \mathrm{K}_{12} + b^{new} - y^{(1)}$$

SMO算法流程总结

- 输入线性可分的m个样本数据{$(x^1,y^1),(x^2,y^2),\ldots,(x^m,y^m)$}，其中x为n维的特征向量，y为二元输出，取值为+1或者-1；精度为e。
  - 取初值$\beta^0=0$，k=0;
  - 选择需要进行更新的两个变量: β_1^k和β_2^k，计算出来新的$\beta_2^{new,unt}$;

$$\beta_2^{new,unt}=\beta_2^k+\frac{y^{(2)}\left(E_1-E_2\right)}{\mathrm{K}_{11}+\mathrm{K}_{22}-2\mathrm{K}_{12}}$$

  - 按照下列式子求出具体的β_2^{k+1};

$$\beta_2^{k+1}=\begin{cases}H;\beta_2^{new,unt}>H\\ \beta_2^{new,unt};L\le\beta_2^{new,unt}\le H\\ L;\beta_2^{new,unt}<L\end{cases}$$

SMO算法流程总结

- 按照β_1^k和β_2^k的关系，求出β_1^{k+1}的值：$$\beta_1^{k+1} = \beta_1^k + y^{(1)}y^{(2)}\left(\beta_2^k - \beta_2^{k+1}\right)$$

- 按照公式计算b^{k+1}和E_i的值；

$$b^{new} = \frac{b_1^{new} + b_2^{new}}{2}$$

$$E_i = g\left(x^{(i)}\right) - y^{(i)} = \sum_{j=3}^{m} \beta_j y^{(j)} \mathrm{K}_{ij} + \beta_1^{new} y^{(1)} \mathrm{K}_{11} + \beta_2^{new} y^{(2)} \mathrm{K}_{12} + b^{new} - y^{(1)}$$

- 检查函数$y^{(i)}*E_i$的绝对值是否在精度范围内，并且求解出来的β解满足KKT相关约束条件，那么此时结束循环，返回此时的β解即可，否则继续迭代计算$\beta_2^{new,unt}$的值。

SVR

- SVM和决策树一样，可以将模型直接应用到回归问题中；在SVM的分类模型(SVC)中，目标函数和限制条件如下：

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{m}\xi_i$$

$$s.t: y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 - \xi_i, i = 1,2,..., m$$

$$\xi_i \geq 0, i = 1,2,..., m$$

- 在SVR中，目的是为了尽量拟合一个线性模型y=wx+b；从而我们可以定义常量eps>0，对于任意一点(x,y)，如果|y-wx-b|≤eps，那么认为没有损失，从而我们可以得到目标函数和限制条件如下：

$$\min_{w,b} \frac{1}{2}\|w\|_2^2$$

$$s.t: \left|y^{(i)} - w^T x^{(i)} - b\right| \leq \varepsilon, i = 1,2,..., m$$

SVR

- 加入松弛因子ξ>0，从而我们的目标函数和限制条件变成：

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{m}\left(\xi_i^{\vee} + \xi_i^{\wedge}\right)$$

$$s.t: -\varepsilon + \xi_i^{\vee} \le y^{(i)} - w^T x^{(i)} - b \le \varepsilon + \xi_i^{\wedge}, i = 1,2,..., m$$

$$\xi_i^{\wedge} \ge 0, \xi_i^{\vee} \ge 0, i = 1,2,..., m$$

SVR

- 构造拉格朗日函数：

$$L\left(w,b,\xi^{\vee},\xi^{\wedge},\beta^{\vee},\beta^{\wedge},\mu^{\vee},\mu^{\wedge}\right)=\frac{1}{2}\|w\|_2^2+C\sum_{i=1}^{m}\left(\xi_i^{\vee}+\xi_i^{\wedge}\right)+\sum_{i=1}^{m}\beta_i^{\vee}\left[-\varepsilon-\xi_i^{\vee}-y^{(i)}+w^T x^{(i)}+b\right]$$

$$+\sum_{i=1}^{m}\beta_i^{\wedge}\left[y^{(i)}-w^T x^{(i)}-b-\varepsilon-\xi_i^{\wedge}\right]-\sum_{i=1}^{m}\mu_i^{\vee}\xi_i^{\vee}-\sum_{i=1}^{m}\mu_i^{\wedge}\xi_i^{\wedge}$$

$$s.t:\beta_i^{\vee}\geq 0,\beta_i^{\wedge}\geq 0,\mu_i^{\vee}\geq 0,\mu_i^{\wedge}\geq 0,i=1,2,\ldots,m$$

SVR

- 拉格朗日函数对偶化

$$\min_{w,b,\xi^{\wedge},\xi^{\vee}} \max_{\beta^{\wedge},\beta^{\vee},\mu^{\wedge},\mu^{\vee}} L\left(w,b,\xi^{\vee},\xi^{\wedge},\beta^{\vee},\beta^{\wedge},\mu^{\vee},\mu^{\wedge}\right)$$

$$\max_{\beta^{\wedge},\beta^{\vee},\mu^{\wedge},\mu^{\vee}} \min_{w,b,\xi^{\wedge},\xi^{\vee}} L\left(w,b,\xi^{\vee},\xi^{\wedge},\beta^{\vee},\beta^{\wedge},\mu^{\vee},\mu^{\wedge}\right)$$

SVR

- 首先来求优化函数对于w、b、ξ的极小值，通过求导可得：

$$\frac{\partial L}{\partial w}=0\Rightarrow w=\sum_{i=1}^{m}\left(\beta_i^{\wedge}-\beta_i^{\vee}\right)x^{(i)}$$

$$\frac{\partial L}{\partial b}=0\Rightarrow \sum_{i=1}^{m}\left(\beta_i^{\wedge}-\beta_i^{\vee}\right)=0$$

$$\frac{\partial L}{\partial \xi_i^{\vee}}=0\Rightarrow C-\beta_i^{\vee}-\mu_i^{\vee}=0$$

$$\frac{\partial L}{\partial \xi_i^{\wedge}}=0\Rightarrow C-\beta_i^{\wedge}-\mu_i^{\wedge}=0$$

SVR

- 将w、b、ξ的值带入函数L中，就可以将L转换为只包含β的函数，从而我们可以得到最终的优化目标函数为(**备注：对于β的求解照样可以使用SMO算法来求解**)：

$$\min_{\beta^{\wedge},\beta^{\vee}} \frac{1}{2}\sum_{i=1,j=1}^{m}\left(\beta_i^{\wedge}-\beta_i^{\vee}\right)\left(\beta_j^{\wedge}-\beta_j^{\vee}\right)K_{ij}+\sum_{i=1}^{m}\left(\varepsilon-y^{(i)}\right)\beta_i^{\wedge}+\left(\varepsilon+y^{(i)}\right)\beta_i^{\vee}$$

$$s.t:\sum_{i=1}^{m}\left(\beta_i^{\wedge}-\beta_i^{\vee}\right)=0$$

$$0<\beta_i^{\wedge}<C,i=1,2,\ldots,m$$

$$0<\beta_i^{\vee}<C,i=1,2,\ldots,m$$

scitit-learn SVM算法库概述
回归算法

参数	LinearSVR	SVR	NuSVR
C	惩罚项系数，默认为1，一般通过交叉验证给定适合值		
nu	不支持，使用epsilon代替		表示训练集上错误率的上限，或者说是支持向量的百分比下限；默认为0.5，取值范围(0,1]，功能和epsilon类似
epsilon	回归模型中引入的误差epsilon值		不支持该，使用nu代替
kernel	不支持，只能使用线性核函数		给定具体的核函数或者Gram矩阵，参数可选:linear、rbf、poly、sigmoid、precomputed
penalty	对线性拟合有意义，参数可选l1和l2		不支持
degree、gamma、coef0	不支持		核函数中的参数值，一般需要通过交叉验证给定；和分类模型一样
loss	可选”epsilon_insensitive”和“squared_epsilon_insensitive”; 第一个表示我们的损坏度量使用绝对值，第二个表示我们的损坏度量使用平方来表示，一般选择epsilon_insensitive即可		不支持