SVM

1. 基本概念

支持向量机（Support Vector Machine, SVM）的基本模型是在特征空间上找到最佳的分离超平面使得训练集上正负样本间隔最大。SVM是用来解决二分类问题的有监督学习算法，在引入了核方法之后SVM也可以用来解决非线性问题。
一般SVM有下面三种：

- 硬间隔支持向量机（线性可分支持向量机）：当训练数据线性可分时，可通过硬间隔最大化学得一个线性可分支持向量机。
- 软间隔支持向量机：当训练数据近似线性可分时，可通过软间隔最大化学得一个线性支持向量机。
- 非线性支持向量机：当训练数据线性不可分时，可通过核方法以及软间隔最大化学得一个非线性支持向量机。

2. 硬间隔支持向量机

给定训练样本集$D = \{(\vec{x_1}, y_1), (\vec{x_2}, y_2), \dots, (\vec{x_n}, y_n)\}$ D={(x1→,y1),(x2→,y2),...,(xn→,yn)}，
$y_i \in \{+1, -1\}$yi∈{+1,−1}，ii表示第ii个样本，nn表示样本容量。分类学习最基本的想法就是基于训练集DD在特征空间中找到一个最佳划分超平面将正负样本分开，而SVM算法解决的就是如何找到最佳超平面的问题。超平面可通过如下的线性方程来描述：

$$\vec{w}^T \vec{x} + b = 0 \tag{1}$$

(1)w→Tx→+b=0
其中$\vec{w}$w→表示法向量，决定了超平面的方向；bb表示偏移量，决定了超平面与原点之间的距离。
对于训练数据集DD假设找到了最佳超平面$\vec{w^*} \vec{x} + b^* = 0$w∗→x→+b∗=0，定义决策分类函数

$$f(\vec{x}) = sign(\vec{w^*} \vec{x} + b^*) \tag{2}$$

(2)f(x→)=sign(w∗→x→+b∗)
该分类决策函数也称为线性可分支持向量机。
在测试时对于线性可分支持向量机可以用一个样本离划分超平面的距离来表示分类预测的可靠程度，如果样本离划分超平面越远则对该样本的分类越可靠，反之就不那么可靠。
那么，什么样的划分超平面是最佳超平面呢？
对于图1有A、B、C三个超平面，很明显应该选择超平面B，也就是说超平面首先应该能满足将两类样本点分开。

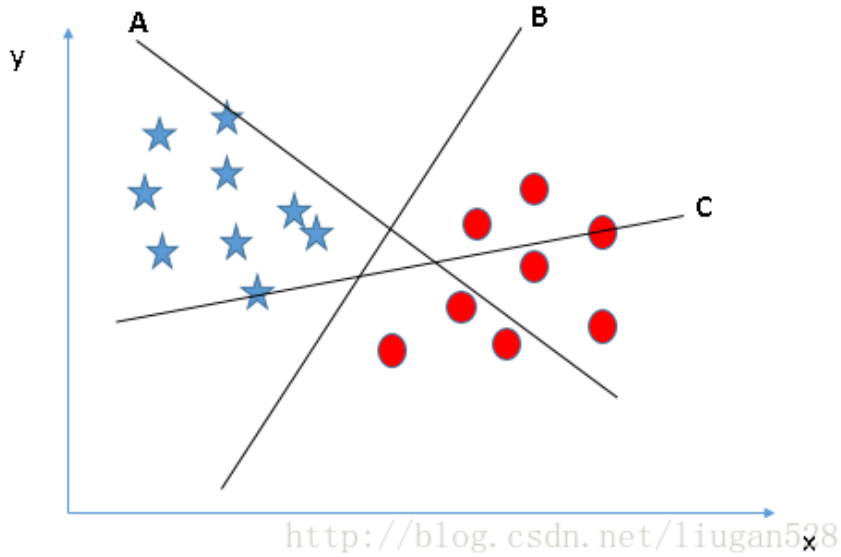

图1

对于图2的A、B、C三个超平面，应该选择超平面C，因为使用超平面C进行划分对训练样本局部扰动的“容忍”度最好，分类的鲁棒性最强。例如，由于训练集的局限性或噪声的干扰，训练集外的样本可能比图2中的训练样本更接近两个类目前的分隔界，在分类决策的时候就会出现错误，而超平面C受影响最小，也就是说超平面C所产生的分类结果是最鲁棒性的、是最可信的，对未见样本的泛化能力最强。

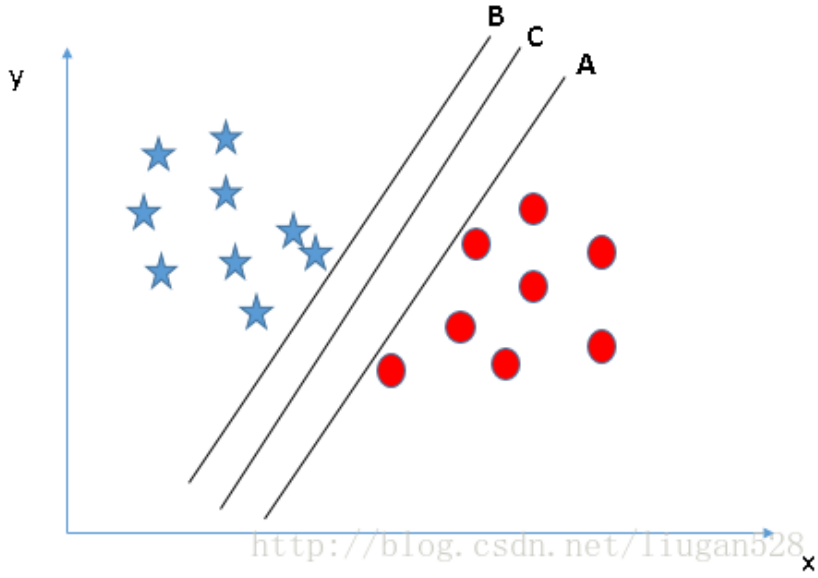

图2

下面以图3中示例进行推导得出最佳超平面。

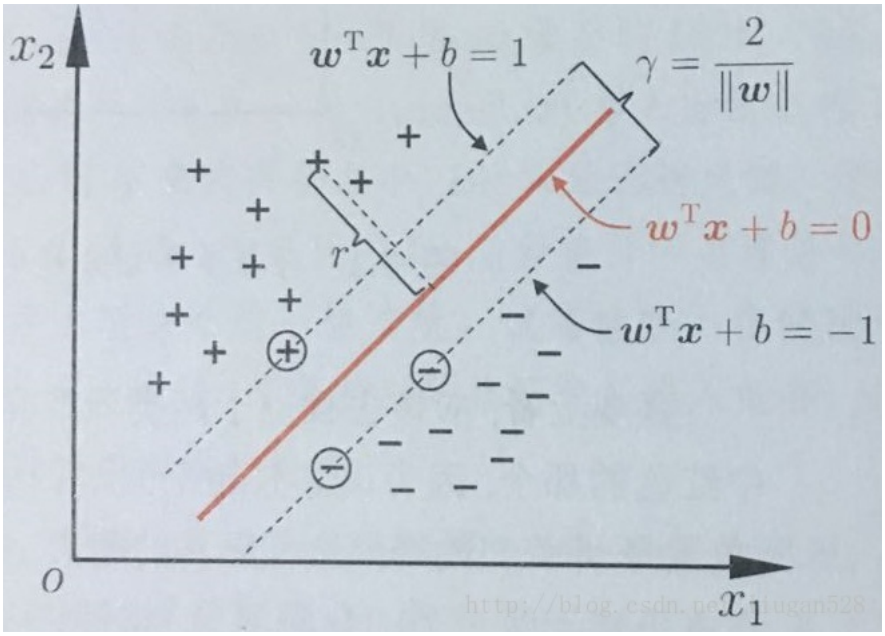

图3

空间中超平面可记为$(\vec{w}, b)$(w→,b)，根据点到平面的距离公式，空间中任意点$\vec{x}$x→到超平面$(\vec{w}, b)$(w→,b)的距离可写为：

$$r = \frac{|\vec{w}\vec{x} + b|}{||\vec{w}||} \tag{3}$$

(3)r=|w→x→+b||w→||

假设超平面$(\vec{w}, b)$(w→,b)能将训练样本正确分类，那么对于正样本一侧的任意一个样本$(\vec{x_i}, y_i) \in D$(xi→,yi)∈D，应该需要满足该样本点往超平面的法向量$\vec{w}$w→的投影到原点的距离大于一定值cc的时候使得该样本点被预测为正样本一类，即存在数值cc使得当$\vec{w}^{\mathrm{T}}\vec{x_i} > c$w→Txi→>c时$y_i = +1$yi=+1。$\vec{w}^{\mathrm{T}}\vec{x_i} > c$w→Txi→>c又可写为$\vec{w}^{\mathrm{T}}\vec{x_i} + b > 0$w→Txi→+b>0。在训练的时候我们要求限制条件更严格点以使最终得到的分类器鲁棒性更强，所以我们要求$\vec{w}^{\mathrm{T}}\vec{x_i} + b > 1$w→Txi→+b>1。也可以写为大于其它距离，但都可以通过同比例缩放$\vec{w}$w→和bb来使得使其变为1，因此为计算方便这里直接选择1。同样对于负样本应该有$\vec{w}^{\mathrm{T}}\vec{x_i} + b < -1$w→Txi→+b<−1时$y_i = -1$yi=−1。即：

$$\begin{cases} \vec{w}^{\mathrm{T}}\vec{x_i} + b \geq +1, & y_i = +1 \\ \vec{w}^{\mathrm{T}}\vec{x_i} + b \leq -1, & y_i = -1 \end{cases} \tag{4}$$

(4){w→Txi→+b≥+1,yi=+1w→Txi→+b≤−1,yi=−1

亦即：

$$y_i(\vec{w}^{\mathrm{T}}\vec{x_i} + b) \geq +1 \tag{5}$$

(5)yi(w→Txi→+b)≥+1

如图3所示，距离最佳超平面$\vec{w}\vec{x} + b = 0$w→x→+b=0最近的几个训练样本点使上式中的等号成立，它们被称为“支持向量”（support vector）。记超平面$\vec{w}\vec{x} + b = +1$w→x→+b=+1和$\vec{w}\vec{x} + b = -1$w→x→+b=−1之间的距离为γγ，该距离又被称为“间隔”（margin），SVM的核心之一就是想办法将“间隔”γγ最大化。下面我们推导一下γγ与哪些因素有关：

记超平面$\vec{w}\vec{x} + b = +1$w→x→+b=+1上的正样本为$\vec{x_+}$x+→，超平面$\vec{w}\vec{x} + b = -1$w→x→+b=−1上的负样本为$\vec{x_-}$x−→，则根据向量的加减法规则$\vec{x_+}$x+→减去$\vec{x_-}$x−→得到的向量在最佳超平面的法向量$\vec{w}$w→方向的投影即为“间隔”γγ：

$$\gamma = (\vec{x_+} - \vec{x_-})\frac{\vec{w}}{||\vec{w}||} = \frac{\vec{x_+}\vec{w}}{||\vec{w}||} - \frac{\vec{x_-}\vec{w}}{||\vec{w}||} \tag{6}$$

(6)γ=(x+→−x−→)w→||w→||=x+→w→||w→||−x−→w→||w→||

而$\vec{w}\vec{x_+} + b = +1$w→x+→+b=+1，$\vec{w}\vec{x_-} + b = -1$w→x−→+b=−1，即：

$$\begin{cases} \vec{w}\vec{x_+} = 1 - b \\ \vec{w}\vec{x_+} = -1 - b \end{cases} \tag{7}$$

(7){w→x+→=1−bw→x+→=−1−b

将(7)带入(6)可得：

$$\gamma = \frac{2}{||\vec{w}||} \tag{8}$$

(8)γ=2||w→||

也就是说使两类样本距离最大的因素仅仅和最佳超平面的法向量有关！

要找到具有“最大间隔”（maximum margin）的最佳超平面，就是找到能满足式(4)中约束的参数$\vec{w}$w→、bb使得γγ最大，即：

$$\begin{cases} \max_{\vec{w},b} \frac{2}{||\vec{w}||} \\ s.t. \quad y_i(\vec{w}^T\vec{x_i} + b) \geq +1, i = 1, 2, \dots, n \end{cases} \tag{9}$$

(9){maxw→,b2||w→||s.t.yi(w→Txi→+b)≥+1,i=1,2,...,n

显然(9)等价于

$$\begin{cases} \min_{\vec{w},b} \frac{1}{2}||\vec{w}||^2 \\ s.t. \quad y_i(\vec{w}^T\vec{x_i} + b) \geq +1, i = 1, 2, \dots, n \end{cases} \tag{10}$$

(10){minw→,b12||w→||2s.t.yi(w→Txi→+b)≥+1,i=1,2,...,n

这就是SVM的基本型。

2.1 拉格朗日对偶问题

根据SVM的基本型求解出$\vec{w}$w→和bb即可得到最佳超平面对应的模型：

$$f(\vec{x}) = sign(\vec{w}\vec{x} + b) \tag{11}$$

(11)f(x→)=sign(w→x→+b)

该求解问题本身是一个凸二次规划（convex quadratic propgramming）问题，可以通过开源的优化计算包进行求解，但是这样就无法体现SVM的精髓，我们可以将该凸二次规划问题通过拉格朗日对偶性来解决。

对于式(10)的每条约束添加拉格朗日乘子$\alpha_i \geq 0$αi≥0，则该问题的拉格朗日函数可写为：

$$L(\vec{w}, b, \vec{\alpha}) = \frac{1}{2}||\vec{w}||^2 - \sum_{i=1}^{n} \alpha_i(y_i(\vec{w}^T x_i + b) - 1) \tag{12}$$

(12)L(w→,b,α→)=12||w→||2−∑i=1nαi(yi(w→Txi+b)−1)

其中$\vec{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$ α→=(α1,α2,...,αn)分别是对应着各个样本的拉格朗日乘子。

将$L(\vec{w}, b, \vec{\alpha})$ L(w→,b,α→)对$\vec{w}$w→和bb求偏导并将偏导数等于零可得：

$$\begin{cases} \vec{w} = \sum_{i=1}^{n} \alpha_i y_i \vec{x_i} \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{13}$$

(13){w→=∑i=1nαiyixi→∑i=1nαiyi=0

将(13)带入(12)消去$\vec{w}$w→和bb就可得到(10)的对偶问题：

$$\begin{cases} \max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \vec{x_i}^T\vec{x_j} \\ s.t. \quad \alpha_i \geq 0, i = 1, 2, \dots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{14}$$

(14){maxα∑i=1nαi−12∑i=1n∑j=1nαiαjyiyjxi→Txj→s.t.αi≥0,i=1,2,...,n∑i=1nαiyi=0

由(14)可知我们并不关心单个样本是如何的，我们只关心样本间两两的乘积，这也为后面核方法提供了很大的便利。

求解出$\vec{\alpha}$α→之后，再求解出$\vec{w}$w→和bb即可得到SVM决策模型：

$$f(\vec{x}) = \vec{w}^T\vec{x} + b = \sum_{i=1}^{n} \alpha_i y_i \vec{x_i}^T\vec{x} + b \tag{15}$$

$$(15) f(x\rightarrow)=w\rightarrow Tx\rightarrow+b=\sum i=1n\alpha iyixi\rightarrow Tx\rightarrow+b$$

2.2 SVM问题的KKT条件

在(10)中有不等式约束，因此上述过程满足Karush-Kuhn-Tucker(KKT)条件：

$$\begin{cases} \alpha_i \geq 0 \\ y_i(\vec{w}^T\vec{x}+b)-1 \geq 0 \\ \alpha_i(y_i(\vec{w}^T\vec{x}+b)-1)=0 \end{cases}, \quad i=1,2,\ldots,n \tag{16}$$

(16){αi≥0yi(w→Tx→+b)−1≥0,i=1,2,...,nαi(yi(w→Tx→+b)−1)=0

对于任意样本$(\vec{x_i},y_i)$(xi→,yi)总有$\alpha_i=0$αi=0或$y_i(\vec{w}^T\vec{x}+b)-1=0$yi(w→Tx→+b)−1=0。如果$\alpha_i=0$αi=0则由式(15)可知该样本点对求解最佳超平面没有任何影响。当$\alpha_i>0$αi>0时必有$y_i(\vec{w}^T\vec{x}+b)-1=0$yi(w→Tx→+b)−1=0，表明对应的样本点在最大间隔边界上，即对应着支持向量。也由此得出了SVM的一个重要性质：训练完成之后，大部分的训练样本都不需要保留，最终的模型仅与支持向量有关。

那么对于式(14)该如何求解$\vec{\alpha}$α→呢？很明显这是一个二次规划问题，可使用通用的二次规划算法来求解，但是SVM的算法复杂度是$O(n^2)$O(n2)，在实际问题中这种开销太大了。为了有效求解该二次规划问题，人们通过利用问题本身的特性，提出了很多高效算法，Sequential Minimal Optimization(SMO)就是一个常用的高效算法。在利用SMO算法进行求解的时候就需要用到上面的KKT条件。利用SMO算法求出$\vec{\alpha}$α→之后根据：

$$\begin{cases} \vec{w}=\sum_{i=1}^{n}\alpha_i y_i \vec{x_i} \\ y_i(\vec{w}^T\vec{x}+b)-1=0 \end{cases} \tag{17}$$

(17){w→=∑i=1nαiyixi→yi(w→Tx→+b)−1=0

即可求出$\vec{w}$w→和bb。求解出$\vec{w}$w→和bb之后就可利用

$$f(\vec{x})=\text{sign}(\vec{w}^T\vec{x}+b) \tag{18}$$

(18)f(x→)=sign(w→Tx→+b)

进行预测分类了，注意在测试的时候不需要−1−1，测试时没有训练的时候要求严格。

3. 软间隔支持向量机

在现实任务中很难找到一个超平面将不同类别的样本完全划分开，即很难找到合适的核函数使得训练样本在特征空间中线性可分。退一步说，即使找到了一个可以使训练集在特征空间中完全分开的核函数，也很难确定这个线性可分的结果是不是由于过拟合导致的。解决该问题的办法是在一定程度上运行SVM在一些样本上出错，为此引入了“软间隔”（soft margin）的概念，如图4所示：

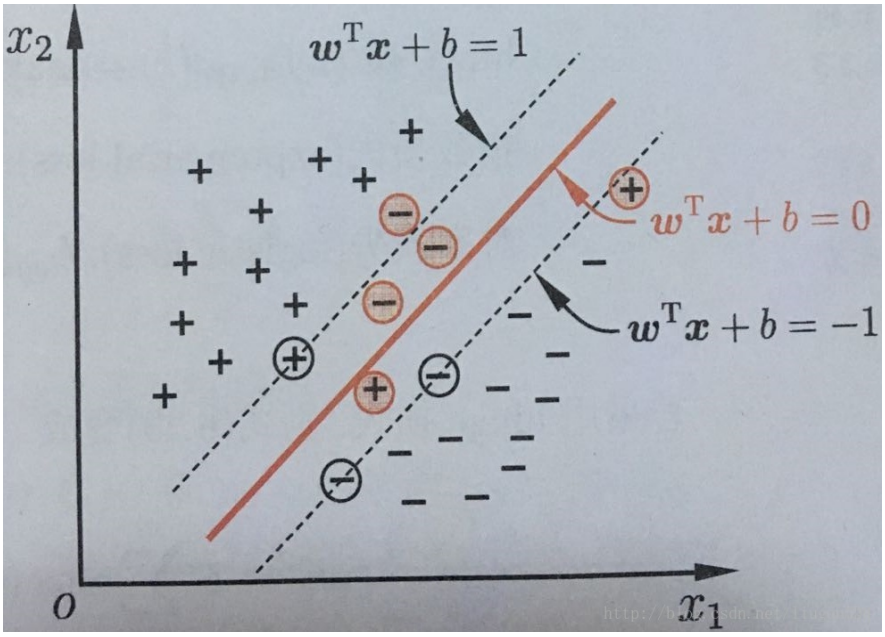

图4

具体来说，硬间隔支持向量机要求所有的样本均被最佳超平面正确划分，而软间隔支持向量机允许某些样本点不满足间隔大于等于1的条件$y_i(\vec{w}\vec{x_i}+b)\geq 1$yi(w→xi→+b)≥1，当然在最大化间隔的时候也要限制不满足间隔大于等于1的样本的个数使之尽可能的少。于是我们引入一个惩罚系数$C>0$C>0，并对每个样本点$(\vec{x_i},y_i)$(xi→,yi)引入一个松弛变量（slack variables）$\xi\geq 0$ξ≥0，此时可将式(10)改写为

$$\begin{cases}\min\limits_{\vec{w},b}\left(\frac{1}{2}\|\vec{w}\|^2+C\sum\limits_{i=1}^{n}\xi_i\right)\\ s.t.\quad y_i(\vec{w}^T\vec{x_i}+b)\geq 1-\xi_i\quad,i=1,2,\ldots,n\\ \qquad\xi_i\geq 0\end{cases}\tag{19}$$

(19){minw→,b(12||w→||2+C∑i=1nξi)s.t.yi(w→Txi→+b)≥1−ξi,i=1,2,...,nξi≥0

上式中约束条件改为$y_i(\vec{w}\vec{x_i}+b)\geq 1-\xi_i$yi(w→xi→+b)≥1−ξi，表示间隔加上松弛变量大于等于1；优化目标改为$\min\limits_{\vec{w},b}\left(\frac{1}{2}\|\vec{w}\|^2+C\sum\limits_{i=1}^{n}\xi_i\right)$minw→,b(12||w→||2+C∑i=1nξi)表示对每个松弛变量都要有一个代价损失$C\xi_i$Cξi，CC越大对误分类的惩罚越大、CC越小对误分类的惩罚越小。

式(19)是软间隔支持向量机的原始问题。可以证明$\vec{w}$w→的解是唯一的，bb的解不是唯一的，bb的解是在一个区间内。假设求解软间隔支持向量机间隔最大化问题得到的最佳超平面是

$\vec{w^*}\vec{x}+b^*=0$w∗→x→+b∗=0，对应的分类决策函数为

$$f(\vec{x})=\mathrm{sign}(\vec{w^*}\vec{x}+b^*)\tag{20}$$

(20)f(x→)=sign(w∗→x→+b∗)

$f(\vec{x})$ f(x→)称为软间隔支持向量机。

类似式(12)利用拉格朗日乘子法可得到上式的拉格朗日函数

$$L(\vec{w},b,\vec{\alpha},\vec{\xi},\vec{\mu})=\frac{1}{2}\|\vec{w}\|^2+C\sum_{i=1}^{n}\xi_i-\sum_{i=1}^{n}\alpha_i(y_i(\vec{w}^T\vec{x_i}+b)-1+\xi_i)-\sum_{i=1}^{n}\mu_i\xi_i\tag{21}$$

(21)L(w→,b,α→,ξ→,μ→)=12||w→||2+C∑i=1nξi−∑i=1nαi(yi(w→Txi→+b)−1+ξi)−∑i=1nμiξi

其中$\alpha_i\geq 0$αi≥0、$\mu_i\geq 0$μi≥0是拉格朗日乘子。

令$L(\vec{w},b,\vec{\alpha},\vec{\xi},\vec{\mu})$ L(w→,b,α→,ξ→,μ→)分别对$\vec{w}$w→，bb，$\vec{\xi}$ξ→求偏导并将偏导数为零可得：

$$\begin{cases}\vec{w}=\sum\limits_{i=1}^{n}\alpha_i y_i\vec{x_i}\\ \sum\limits_{i=1}^{n}\alpha_i y_i=0\\ C=\alpha_i+\mu_i\end{cases}\tag{22}$$

(22){w→=∑i=1nαiyixi→∑i=1nαiyi=0C=αi+μi

将式(22)带入式(21)便可得到式(19)的对偶问题：

$$\begin{cases}\max\limits_{\vec{\alpha}}\sum\limits_{i=1}^{n}\alpha_i-\frac{1}{2}\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}\alpha_i\alpha_j y_i y_j\vec{x_i}^T\vec{x_j}\\ s.t.\quad\sum\limits_{i=1}^{n}\alpha_i y_i=0\quad,i=1,2,\ldots,n\\ \qquad 0\leq\alpha_i\leq C\end{cases}\tag{23}$$

(23){maxα→∑i=1nαi−12∑i=1n∑j=1nαiαjyiyjxi→Txj→s.t.∑i=1nαiyi=0,i=1,2,...,n0≤αi≤C

对比软间隔支持向量机的对偶问题和硬间隔支持向量机的对偶问题可发现二者的唯一差别就在于对偶变量的约束不同，软间隔支持向量机对对偶变量的约束是$0\leq\alpha_i\leq C$0≤αi≤C，硬间隔支持向量机对对偶变量的约束是$0\leq\alpha_i$0≤αi，于是可采用和硬间隔支持向量机相同的解法求解式(23)。同理在引入核方法之后同样能得到与式(23)同样的支持向量展开式。

类似式(16)对于软间隔支持向量机，KKT条件要求：

$$\begin{cases}\alpha_i\geq 0,\mu_i\geq 0\\ y_i(\vec{w}\vec{x}+b)-1+\xi_i\geq 0\\ \alpha_i(y_i(\vec{w}\vec{x}+b)-1+\xi_i)=0\\ \xi_i\geq 0,\mu_i\xi_i=0\end{cases}\tag{24}$$

(24){αi≥0,μi≥0yi(w→x→+b)−1+ξi≥0αi(yi(w→x→+b)−1+ξi)=0ξi≥0,μiξi=0

同硬间隔支持向量机类似，对任意训练样本$(\vec{x_i},y_i)$(xi→,yi)，总有$\alpha_i=0$αi=0或$y_i(\vec{w}\vec{x}+b-1+\xi_i)$yi(w→x→+b−1+ξi)，若$\alpha_i=0$αi=0，则该样本不会对最佳决策面有任何影响；若

$\alpha_i > 0$αi>0则必有$y_i(\vec{w}\vec{x} + b) = 1 - \xi_i$yi(w→x→+b)=1−ξi，也就是说该样本是支持向量。由式(22)可知若$\alpha_i < C$αi<C则$\mu_i > 0$μi>0进而有$\xi_i = 0$ξi=0，即该样本处在最大间隔边界上；若$\alpha_i = C$αi=C则$\mu_i = 0$μi=0此时如果$xi_i \leq 1$xii≤1则该样本处于最大间隔内部，如果$\xi_i > 1$ξi>1则该样本处于最大间隔外部即被分错了。由此也可看出，软间隔支持向量机的最终模型仅与支持向量有关。

4. 非线性支持向量机

现实任务中原始的样本空间D D中很可能并不存在一个能正确划分两类样本的超平面。例如图4中所示的问题就无法找到一个超平面将两类样本进行很好的划分。
对于这样的问题可以通过将样本从原始空间映射到特征空间使得样本在映射后的特征空间里线性可分。例如对图5做特征映射$z = x^2 + y^2$z=x2+y2可得到如图6所示的样本分布，这样就很好进行线性划分了。

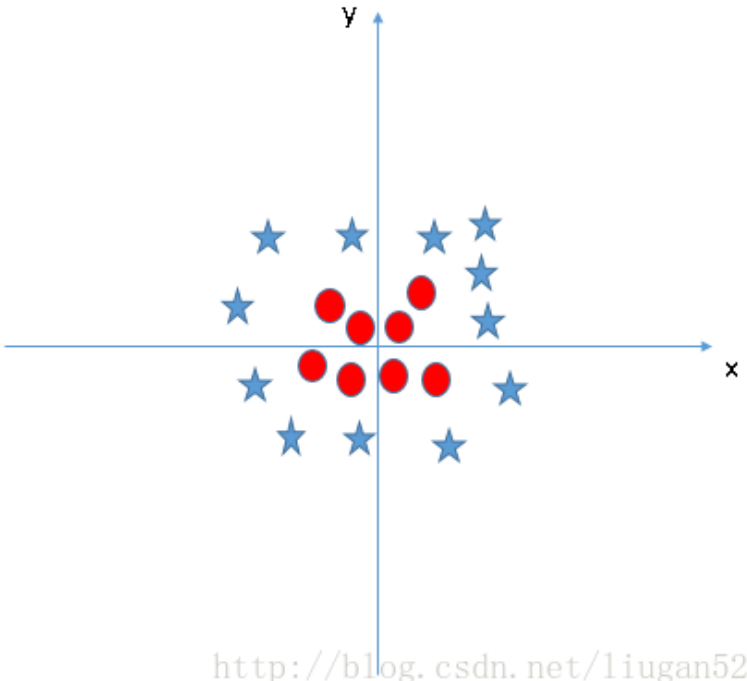

图5

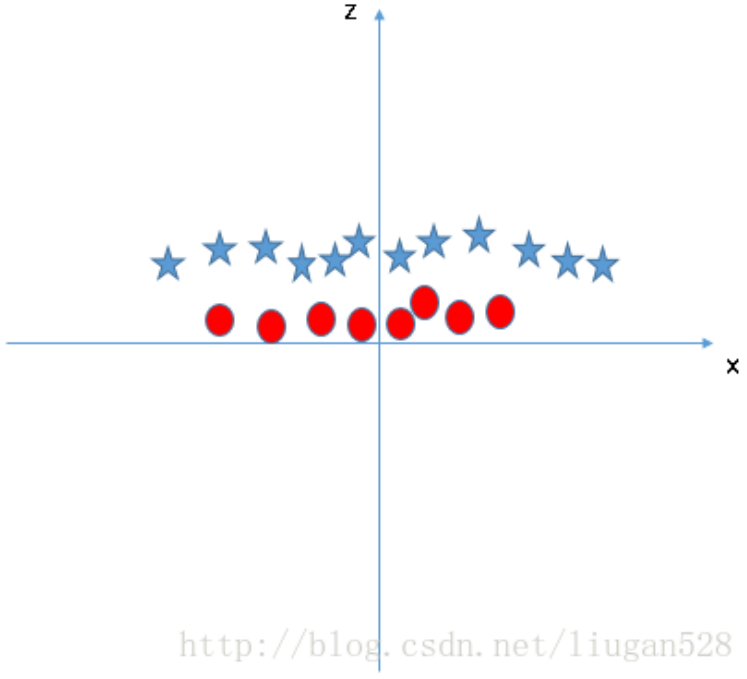

图6

令$\phi(\vec{x})$ϕ(x→)表示将样本点$\vec{x}$x→映射后的特征向量，类似于线性可分支持向量机中的表示方法，在特征空间

中划分超平面所对应的模型可表示为

$$f(\vec{x}) = \vec{w}^T x + b \tag{25}$$

(25)f(x→)=w→Tx+b

其中$\vec{w}$w→和bb是待求解的模型参数。类似式(10)，有

$$\begin{cases} \min_{\vec{w},b} \frac{1}{2}||\vec{w}||^2 \\ s.t. \quad y_i(\vec{w}^T \phi(\vec{x}) + b) \geq 1 \quad , i = 1, 2, \ldots, n \end{cases} \tag{26}$$

(26){minw→,b12||w→||2s.t.yi(w→Tϕ(x→)+b)≥1,i=1,2,...,n

其拉格朗日对偶问题是

$$\begin{cases} \max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \phi(\vec{x_i}^T) \phi(\vec{x_j}) \\ s.t. \quad \alpha_i \geq 0 \quad , i = 1, 2, \ldots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{27}$$

(27){maxα∑i=1nαi−12∑i=1n∑j=1nαiαjyiyjϕ(xi→T)ϕ(xj→)s.t.αi≥0,i=1,2,...,n∑i=1nαiyi=0

求解(27)需要计算$\phi(\vec{x_i}^T)\phi(\vec{x_j})$ϕ(xi→T)ϕ(xj→)，即样本映射到特征空间之后的内积，由于特征空间可能维度很高，甚至可能是无穷维，因此直接计算$\phi(\vec{x_i}^T)\phi(\vec{x_j})$ϕ(xi→T)ϕ(xj→)通常是很困难的，在上文中我们提到其实我们根本不关心单个样本的表现，只关心特征空间中样本间两两的乘积，因此我们没有必要把原始空间的样本一个个地映射到特征空间中，只需要想法办求解出样本对应到特征空间中样本间两两的乘积即可。为了解决该问题可设想存在核函数：

$$\kappa(\vec{x_i}, \vec{x_j}) = \phi(\vec{x_i}^T)\phi(\vec{x_j}) \tag{28}$$

(28)κ(xi→,xj→)=ϕ(xi→T)ϕ(xj→)

也就是说$\vec{x_i}$xi→与$\vec{x_j}$xj→在特征空间的内积等于它们在原始空间中通过函数$\kappa(\cdot,\cdot)$κ(⋅,⋅)计算的结果，这给求解带来很大的方便。于是式(27)可写为：

$$\begin{cases} \max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \kappa(\vec{x_i}, \vec{x_j}) \\ s.t. \quad \alpha_i \geq 0 \quad , i = 1, 2, \ldots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{29}$$

(29){maxα∑i=1nαi−12∑i=1n∑j=1nαiαjyiyjκ(xi→,xj→)s.t.αi≥0,i=1,2,...,n∑i=1nαiyi=0

同样的我们只关心在高维空间中样本之间两两点乘的结果而不关心样本是如何变换到高维空间中去的。求解后即可得到

$$f(\vec{x}) = \vec{w}^T \phi(\vec{x}) + b = \sum_{i=1}^{n} \alpha_i y_i \phi(\vec{x})^T \phi(\vec{x}) + b = \sum_{i=1}^{n} \alpha_i y_i \kappa(\vec{x_i}, \vec{x_j}) + b \tag{30}$$

(30)f(x→)=w→Tϕ(x→)+b=∑i=1nαiyiϕ(x→)Tϕ(x→)+b=∑i=1nαiyiκ(xi→,xj→)+b

剩余的问题同样是求解α_iαi，然后求解$\vec{w}$w→和bb即可得到最佳超平面。

支持向量回归

支持向量机不仅可以用来解决分类问题还可以用来解决回归问题，称为支持向量回归（Support Vector Regression，SVR）。

对于样本$(\vec{x}, y)$(x→,y)通常根据模型的输出$f(\vec{x})$f(x→)与真实值（即groundtruth）y_i yi之间的差别来计算损失，当且仅当$f(\vec{x}) = y_i$ f(x→)=yi时损失才为零。SVR的基本思路是允许预测值$f(\vec{x})$ f(x→)与y_i yi之间最多有εε的偏差，当$|f(\vec{x}) - y_i| \leq \varepsilon$ |f(x→)−yi|≤ε时认为预测正确不计算损失，仅当$|f(\vec{x}) - y_i| > \varepsilon$ |f(x→)−yi|>ε时才计算损失。SVR问题可描述为：

$$\min_{\vec{w},b} \left(\frac{1}{2}||\vec{w}||^2 + C \sum_{i=1}^{n} L_\varepsilon(f(\vec{x}) - y_i) \right) \tag{31}$$

(31)minw→,b(12||w→||2+C∑i=1nLε(f(x→)−yi))

其中，$C \geq 0$C≥0为惩罚项，L_εLε为损失函数，定义为：

$$L_\varepsilon(z) = \begin{cases} 0 & , |z| \leq \varepsilon \\ |z| - \xi & , otherwise \end{cases} \tag{32}$$

$$(32)L\varepsilon(z)=\{0,|z|\le\varepsilon|z|-\xi,otherwise$$

进一步地引入松弛变量ξ_i ξi，$\hat{\xi}_i$ ξ^i，则新的最优化问题为：

$$\begin{cases} \min_{\vec{w},b,\xi,\hat{\xi}_i}(\frac{1}{2}||\vec{w}||^2 + C\sum_{i=1}^{n}(\xi_i + \hat{\xi}_i)) \\ s.t. \quad f(\overrightarrow{x_i}) - y_i \le \varepsilon + \xi_i \\ \qquad y_i - f(\vec{x}) \le \varepsilon + \hat{\xi}_i \\ \qquad \xi_i \ge 0, \hat{\xi}_i \ge 0 \end{cases} \quad , i = 1, 2, \dots, n \tag{33}$$

(33){minw→,b,ξ,ξ^i(12||w→||2+C∑i=1n(ξi+ξ^i))s.t.f(xi→)−yi≤ε+ξi,i=1,2,
…,nyi−f(x→)≤ε+ξ^iξi≥0,ξ^i≥0

这就是SVR的原始问题。类似地引入拉格朗日乘子$\mu_i \ge 0$μi≥0，$\hat{\mu}_i \ge 0$μ^i≥0，$\alpha_i \ge 0$αi≥0，$\hat{\alpha}_i \ge 0$α^i≥0，则对应的拉格朗日函数为：

$$L(\vec{w}, b, \vec{\alpha}, \vec{\hat{\alpha}}, \vec{\xi}, \vec{\hat{\xi}}, \vec{\mu}, \vec{\hat{\mu}}) = \frac{1}{2}||\vec{w}||^2 + C\sum_{i=1}^{n}(\xi + \hat{\xi}) - \sum_{i=1}^{n}\mu_i\xi_i - \sum_{i=1}^{n}\hat{\mu}_i\hat{\xi}_i + \tag{34}$$

$$\sum_{i=1}^{n}\alpha_i(f(\overrightarrow{x_i}) - y_i - \varepsilon - \xi) + \sum_{i=1}^{n}\hat{\alpha}_i(y_i - f(\overrightarrow{x_i}) - \varepsilon - \hat{\xi}_i)$$

(34)L(w→,b,α→,α^→,ξ→,ξ^→,μ→,μ^→)=12||w→||2+C∑i=1n(ξ+ξ^)
−∑i=1nμiξi−∑i=1nμ^iξ^i+∑i=1nαi(f(xi→)−yi−ε−ξ)+∑i=1nα^i(yi−f(xi→)−ε−ξ^i)

令$L(\vec{w}, b, \vec{\alpha}, \vec{\hat{\alpha}}, \vec{\xi}, \vec{\hat{\xi}}, \vec{\mu}, \vec{\hat{\mu}})$ L(w→,b,α→,α^→,ξ→,ξ^→,μ→,μ^→)对$\vec{w}, b, \vec{\xi}, \vec{\hat{\xi}}$w→,b,ξ→,ξ^→的偏导数为零可得：

$$\begin{cases} \vec{w} = \sum_{i=1}^{n}()\overrightarrow{x_i} \\ \sum_{i=1}^{n}(\hat{\alpha}_i - \alpha_i) = 0 \\ C = \alpha_i + \mu_i \\ C = \hat{\alpha}_i + \hat{\mu}_i \end{cases} \tag{35}$$

(35){w→=∑i=1n()xi→∑i=1n(α^i−αi)=0C=αi+μiC=α^i+μ^i

将式(35)代入式(34)即可得到SVR的对偶问题：

$$\begin{cases} \max_{\alpha,\hat{\alpha}}\sum_{i=1}^{n}(y_i(\hat{\alpha}_i - \alpha_i) - \varepsilon(\hat{\alpha}_i + \alpha_i) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j)\overrightarrow{x_i}^T\overrightarrow{x_j}) \\ s.t.\sum_{i=1}^{n}(\hat{\alpha}_i - \alpha_i) = 0 \qquad 0 \le \alpha_i, \hat{\alpha}_i \le C \end{cases} \tag{36}$$

(36){maxα,α^∑i=1n(yi(α^i−αi)−ε(α^i+αi)−12∑i=1n∑j=1n(α^i−αi)
(α^j−αj)xi→Txj→)s.t.∑i=1n(α^i−αi)=00≤αi,α^i≤C

其KKT条件为：

$$\begin{cases} \alpha_i(f(\overrightarrow{x_i}) - y_i - \varepsilon - \xi_i) = 0 \\ \hat{\alpha}_i(f(\overrightarrow{x_i}) - y_i - \varepsilon - \hat{\xi}_i) = 0 \\ \alpha_i\hat{\alpha}_i = 0, \xi_i\hat{\xi}_i = 0 \\ (C - \alpha_i)\xi_i = 0, (C - \hat{\alpha}_i)\hat{\xi}_i = 0 \end{cases} \tag{37}$$

(37){αi(f(xi→)−yi−ε−ξi)=0α^i(f(xi→)−yi−ε−ξ^i)=0αiα^i=0,ξiξ^i=0(C−αi)ξi=0,(C−α^i)ξ^i=0

SVR的解形如：

$$f(\vec{x}) = \sum_{i=1}^{n}(\hat{\alpha}_i - \alpha_i)\overrightarrow{x_i}^T\vec{x} + b \tag{38}$$

(38)f(x→)=∑i=1n(α^i−αi)xi→Tx→+b

进一步地如果引入核函数则SVR可表示为：

$$f(\vec{x}) = \sum_{i=1}^{n}(\hat{\alpha}_i - \alpha_i)\kappa(\overrightarrow{x_i}, \vec{x}) + b \tag{39}$$

(39)f(x→)=∑i=1n(α^i−αi)κ(xi→,x→)+b

其中$\kappa(\overrightarrow{x_i}, \vec{x}) = \phi(\overrightarrow{x_i})^T\phi(\vec{x})$ κ(xi→,x→)=ϕ(xi→)Tϕ(x→)为核函数。

常用核函数

名称	表达式	参数
线性核	$\kappa(\vec{x_i}, \vec{x_j}) = \vec{x_i}^T \vec{x_j}$ κ(xi→,xj→)=xi→Txj→	
多项式核	$\kappa(\vec{x_i}, \vec{x_j}) = (\vec{x_i}^T \vec{x_j})^n$ κ(xi→,xj→)=(xi→Txj→)n	$n \geq 1$n≥1为多项式的次数
高斯核(RBF)	$\kappa(\vec{x_i}, \vec{x_j}) = \exp(-\frcd{\|\vec{x_i}-\vec{x_j}\|^2}{2\sigma^2})$ κ(xi→,xj→)=exp(−||xi→−xj→||22σ2)	$\sigma > 0$σ>0为高斯核的带宽
拉普拉斯核	$\kappa(\vec{x_i}, \vec{x_j}) = \exp(-\frac{\|x_i-x_j\|}{\sigma})$ κ(xi→,xj→)=exp(−||xi−xj||σ)	σσ>0
Sigmoid核	$\kappa(\vec{x_i}, \vec{x_j}) = \tanh(\beta\vec{x_i}^T \vec{x_j} + \theta)$ κ(xi→,xj→)=tanh(βxi→Txj→+θ)	thah为双曲正切函数

5. SVM的优缺点

优点：
SVM在中小量样本规模的时候容易得到数据和特征之间的非线性关系，可以避免使用神经网络结构选择和局部极小值问题，可解释性强，可以解决高维问题。
缺点：
SVM对缺失数据敏感，对非线性问题没有通用的解决方案，核函数的正确选择不容易，计算复杂度高，主流的算法可以达到$O(n^2)$O(n2)的复杂度，这对大规模的数据是吃不消的。

6. 参考文献

周志华. 机器学习 [D]. 清华大学出版社，2016.
华校专、王正林. Python大战机器学习 [D]. 电子工业出版社，2017.
Peter Flach著、段菲译. 机器学习 [D]. 人民邮电出版社，2016.
Understanding Support Vector Machine algorithm from examples (along with code)
KKT条件介绍

更多资料请移步github：
https://github.com/GarryLau/MachineLearning