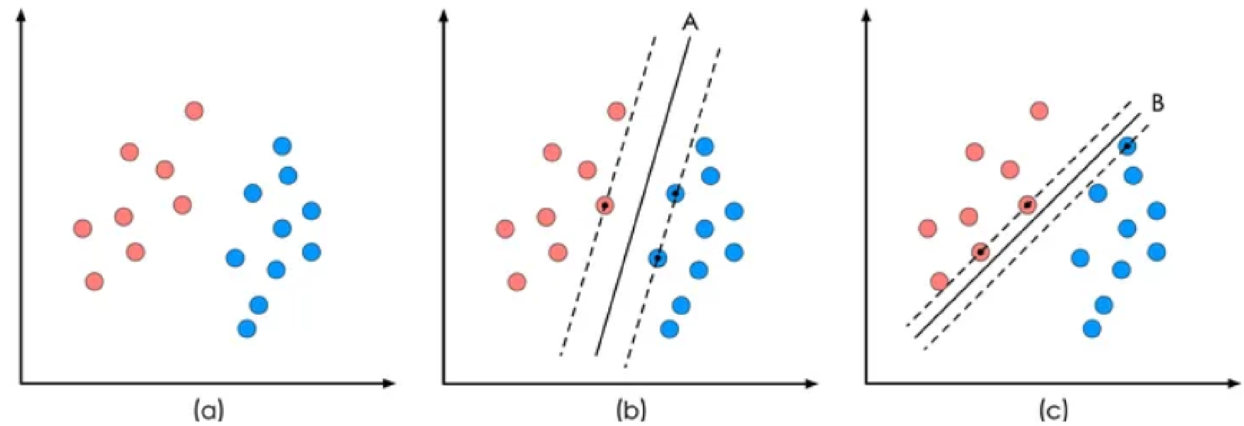
SVM的前世今生

简介

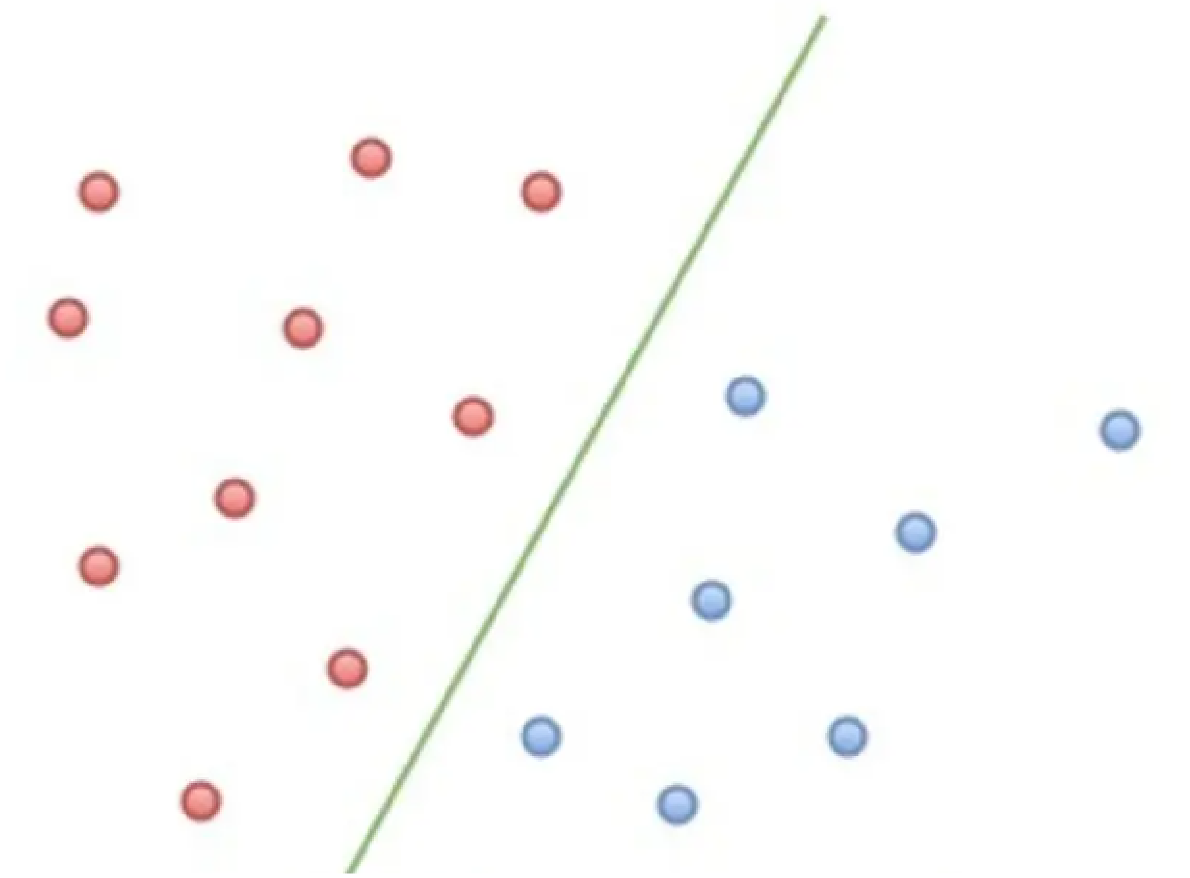
支持向量机（support vector machines, SVM）是一种二分类模型，它的基本模型是定义在特征空间上的**间隔最大的线性分类器**，间隔最大使它有别于感知机；SVM还包括**核技巧**，这使它成为实质上的非线性分类器。SVM的的学习策略就是间隔最大化，可形式化为一个求解凸二次规划的问题，也等价于正则化的合页损失函数的最小化问题。SVM的的学习算法就是求解凸二次规划的最优化算法。



基础概念

1.线性可分

在二维空间上，两类点被一条直线完全分开叫做线性可分。



严格的数学定义是：

D_0和D_1是n维欧氏空间中的两个点集。

如果存在n维向量w和实数b，使得所有属于D_0的点x_i都有$wx_i+b>0$，

而对于所有属于D_1的点x_j则有$wx_j+b<0$，则我们称D_0和D_1线性可分。

2.最大间隔超平面

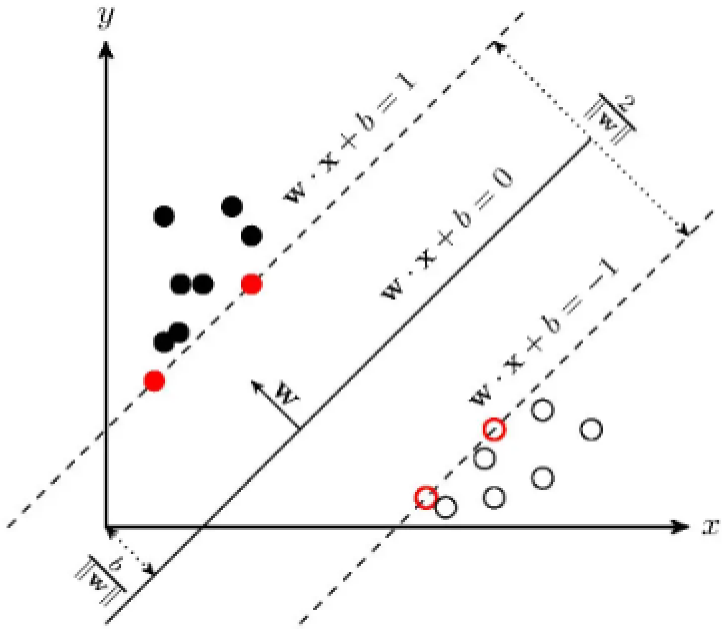
从二维扩展到多维空间中时，将 D_0和D_1完全正确地划分开的$wx+b=0$ 就成了一个超平面。

为了使这个超平面更具鲁棒性，我们会去找最佳超平面，以最大间隔把两类样本分开的超平面，也称之为最大间隔超平面。

1. 两类样本分别分割在该超平面的两侧；
2. 两侧距离超平面最近的样本点到超平面的距离被最大化了。

3.支持向量

样本中距离超平面最近的一些点，这些点叫做支持向量。

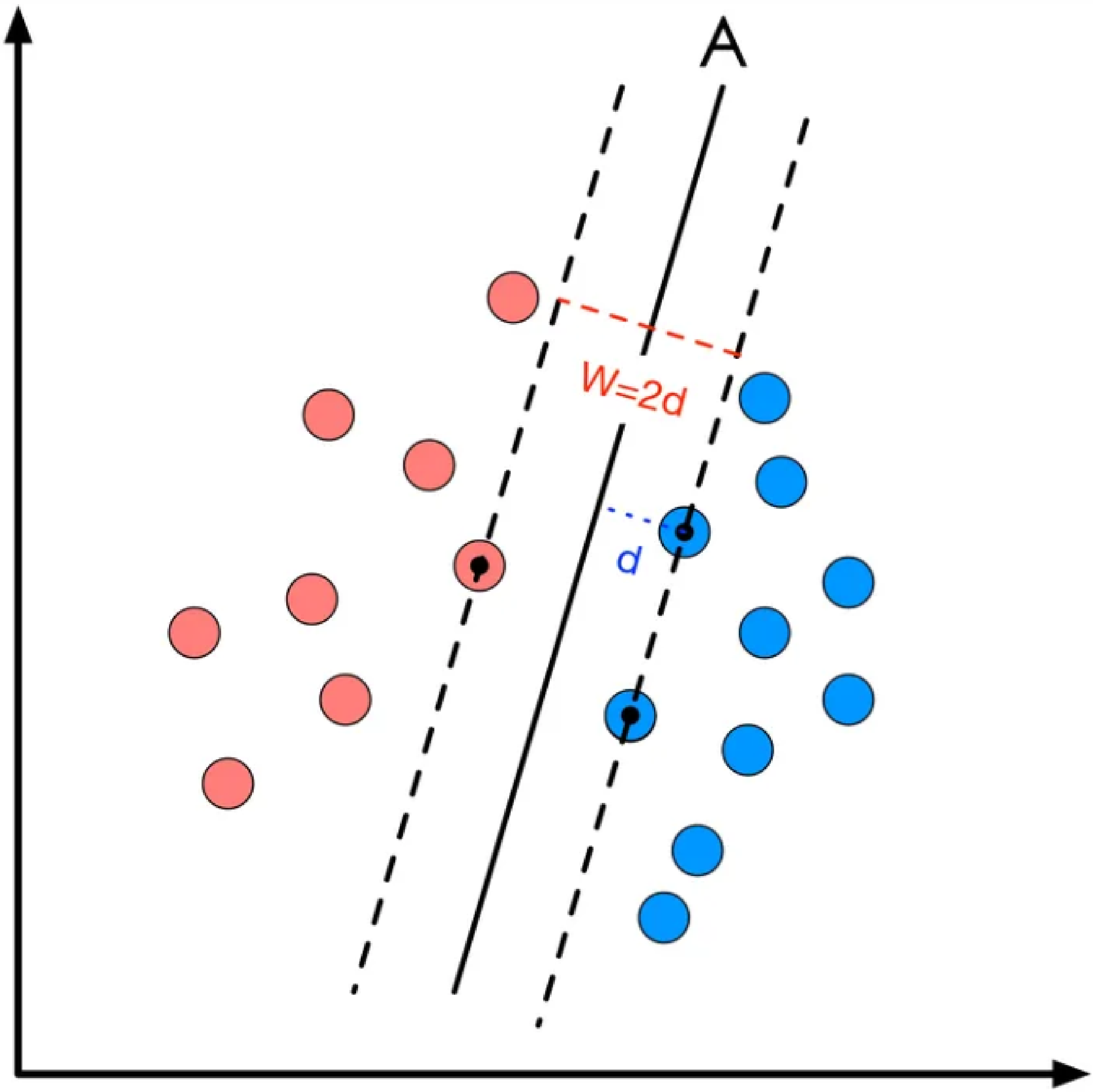


4 SVM 最优化问题

SVM 想要的就是找到各类样本点到超平面的距离最远，也就是找到最大间隔超平面。

$$\text{任意超平面可以用下面这个线性方程来描述：}$$
$$w^T x + b = 0$$
$$\text{二维空间点}(x, y)\text{到直线}Ax + By + C = 0\text{的距离公式是：}\frac{|Ax + By + C|}{\sqrt{A^2 + B^2}}$$
$$\text{扩展到}n\text{维空间后，点}x = (x_1, x_2 \ldots x_n)\text{到直线}w^T x + b = 0\text{的距离为：}\frac{|w^T x + b|}{||w||}$$
$$\text{其中}||w|| = \sqrt{w_1^2 + \ldots w_n^2}\text{。}$$

如图所示，根据支持向量的定义我们知道，支持向量到超平面的距离为 d，其他点到超平面的距离大于 d。



于是我们有这样的一个公式：

$$\begin{cases} \frac{w^Tx+b}{||w||} \geq d \quad y=1 \\ \frac{w^Tx+b}{||w||} \leq -d \quad y=-1 \end{cases}$$

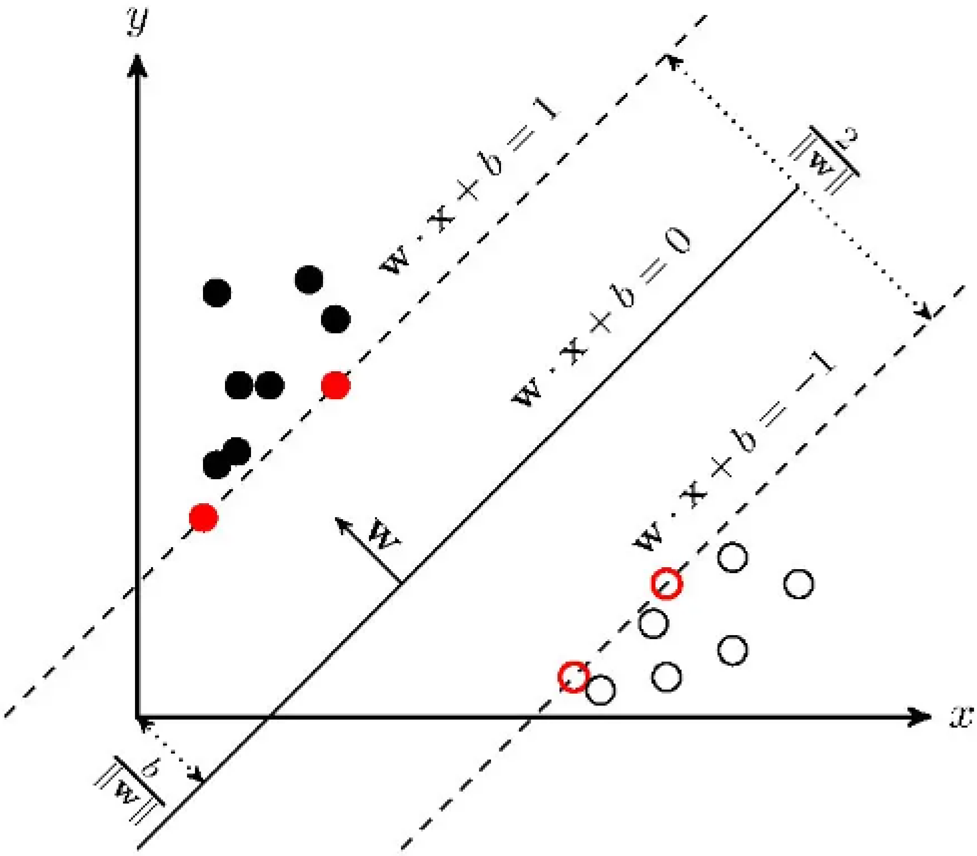
稍作转化可以得到：$\begin{cases} \frac{w^Tx+b}{||w||d} \geq 1 \quad y=1 \\ \frac{w^Tx+b}{||w||d} \leq -1 \quad y=-1 \end{cases}$

$||w||d$是正数，我们暂且令它为1（之所以令它等于1，是为了方便推导和优化，且这样做对目标函数的优化没有影响），故：

$$0\begin{cases} w^Tx+b \geq 1 \quad y=1 \\ w^Tx+b \leq -1 \quad y=-1 \end{cases}$$

将两个方程合并，我们可以简写为：$y(w^Tx+b) \geq 1$

至此我们就可以得到最大间隔超平面的上下两个超平面



每个支持向量到超平面的距离可以写为：

$$d=\frac{|w^Tx+b|}{||w||}$$

由上述$y(w^Tx+b)>1>0$可以得到$y(w^Tx+b)=|w^Tx+b|$，所以我们得到：$d=\frac{y(w^Tx+b)}{||w||}$

最大化这个距离：$\max 2*\frac{y(w^Tx+b)}{||w||}$

这里乘上2倍也是为了后面推导，对目标函数没有影响。

刚刚我们得到支持向量$y(w^Tx+b)=1$，所以我们得到：

$$\max\frac{2}{||w||}$$

再做一个转换：$\min\frac{1}{2}||w||$

为了方便计算（去除$||w||$的根号），我们有：$\min\frac{1}{2}||w||^2$

所以得到的最优化问题是：$\min\frac{1}{2}||w||^2\ s.t.\quad y_i（w^Tx_i+b）\geq 1$

转换为对偶问题

（1）等式约束优化问题

拉格朗日乘数法求解等式约束优化问题：

$$\min f(x_1,x_2,\dots,x_n)$$
$$s.t.\quad h_k(x_1,x_2,\dots,x_n)=0\quad k=1,2,\dots,l$$

我们令$L(x,\lambda)=f(x)+\sum_{k=1}^{l}\lambda_k h_k(x)$，

函数$L(x,y)$称为$Lagrange$函数，

参数λ称为$Lagrange$乘子没有非负要求。

利用必要条件找到可能的极值点：$\begin{cases}\dfrac{\partial L}{\partial x_i}=0 & i=1,2,\dots,n\\ \dfrac{\partial L}{\partial \lambda_k}=0 & k=1,2,\dots,l\end{cases}$

具体是否为极值点需根据问题本身的具体情况检验。

这个方程组称为等式约束的极值必要条件。

等式约束下的$Lagrange$乘数法引入了l个$Lagrange$乘子，

我们将x_i与λ_k一视同仁，把λ_k也看作优化变量，共有$(n+l)$个优化变量。

（2） 不等式约束优化问题

而我们现在面对的是不等式优化问题，针对这种情况其主要思想是将不等式约束条件转变为等式约束条件，引入松弛变量，将松弛变量也是为优化变量。

以我们的例子为例：$minf(w)=min\dfrac{1}{2}||w||^2$

$$s.t.\quad g_i(w)=1-y_i(w^Tx_i+b)\le 0$$

我们引入松弛变量a_i^2得到$h_i(w,a_i)=g_i(w)+a_i^2=0$。

这里加平方主要为了不再引入新的约束条件，

如果只引入a_i那我们必须要保证$a_i\ge 0$才能保证$h_i(w,a_i)=0$，

这不符合我们的意愿。

由此我们将不等式约束转化为了等式约束，并得到$Lagrange$函数：

$$\begin{aligned}L(w,\lambda,a)&=f(w)+\sum_{i=1}^{n}\lambda_i h_i(w)\\&=f(w)+\sum_{i=1}^{n}\lambda_i[g_i(w)+a_i^2]\quad \lambda_i\ge 0\end{aligned}$$

由等式约束优化问题极值的必要条件对其求解，联立方程：

$$\begin{cases}\dfrac{\partial L}{\partial w_i}=\dfrac{\partial f}{\partial w_i}+\sum_{i=1}^{n}\lambda_i\dfrac{\partial g_i}{\partial w_i}=0,\\ \dfrac{\partial L}{\partial a_i}=2\lambda_i a_i=0,\\ \dfrac{\partial L}{\partial \lambda_i}=g_i(w)+a_i^2=0,\\ \lambda_i\ge 0\end{cases}$$

针对$\lambda_i a_i = 0$我们有两种情况：

情形一：$\lambda_i = 0, a_i \neq 0$

由于$\lambda_i = 0$，因此约束条件$g_i(w)$不起作用，且$g_i(w) < 0$

情形二：$\lambda_i \neq 0, a_i = 0$

此时$g_i(w) = 0$且$\lambda_i > 0$，可以理解为约束条件$g_i(w)$起作用了，且$g_i(w) = 0$

综合可得：$\lambda_i g_i(w) = 0$，且在约束条件起作用时$\lambda_i > 0, g_i(w) = 0$；约束不起作用时$\lambda_i = 0, g_i(w) < 0$

由此方程组转换为：

$$\begin{cases} \dfrac{\partial L}{\partial w_i} = \dfrac{\partial f}{\partial w_i} + \sum_{j=1}^{n} \lambda_j \dfrac{\partial g_j}{\partial w_i} = 0, \\ \lambda_i g_i(w) = 0, \\ g_i(w) \leq 0 \\ \lambda_i \geq 0 \end{cases}$$

以上便是不等式约束优化优化问题的$KKT(Karush-Kuhn-Tucker)$条件，λ_i称为KKT乘子。

这个式子告诉了我们什么事情呢？

直观来讲就是，支持向量$g_i(w) = 0$，所以$\lambda_i > 0$即可。而其他向量$g_i(w) < 0, \lambda_i = 0$。

我们原本问题时要求：$min\frac{1}{2}||w||^2$，即求$minL(w, \lambda, a)$

$$\begin{aligned} L(w, \lambda, a) &= f(w) + \sum_{i=1}^{n} \lambda_i [g_i(w) + a_i^2] \\ &= f(w) + \sum_{i=1}^{n} \lambda_i g_i(w) + \sum_{i=1}^{n} \lambda_i a_i^2 \end{aligned}$$

由于$\sum_{i=1}^{n} \lambda_i a_i^2 \geq 0$，故我们将问题转换为：$minL(w, \lambda)$：

$$L(w, \lambda) = f(w) + \sum_{i=1}^{n} \lambda_i g_i(w)$$

假设找到了最佳参数是的目标函数取得了最小值p。即$\frac{1}{2}||w||^2 = p$。

而根据$\lambda_i \geq 0$，可知$\sum_{i=1}^{n} \lambda_i g_i(w) \leq 0$，

因此$L(w, \lambda) \leq p$，为了找到最优的参数λ，使得$L(w, \lambda)$接近p，

故问题转换为出$\max_{\lambda} L(w, \lambda)$。

故我们的最优化问题转换为：

$$\min_{w} \max_{\lambda} L(w, \lambda)$$
$$s.t. \quad \lambda_i \geq 0$$

出了上面的理解方式，我们还可以有另一种理解方式：由于$\lambda_i \geq 0$，

$$\max_{\lambda} L(w, \lambda) = \begin{cases} \infty & g_i(w) \geq 0 \\ \frac{1}{2}||w||^2 & g_i(w) \leq 0 \end{cases}$$

所以$\min(\infty, \frac{1}{2}||w||^2) = \frac{1}{2}||w||^2$，所以转化后的式子和原来的式子也是一样的。

证明对偶问题与原问题的强对偶性

对偶问题其实就是将：

$$\min_w \max_\lambda L(w,\lambda)$$

$$s.t. \quad \lambda_i \geq 0$$

变成了：

$$\max_\lambda \min_w L(w,\lambda)$$

$$s.t. \quad \lambda_i \geq 0$$

假设有个函数f我们有：

$$\min \max f \geq \max \min f$$

也就是说，最大的里面挑出来的最小的也要比最小的里面挑出来的最大的要大。

这关系实际上就是弱对偶关系，而强对偶关系是当等号成立时，即：

$$\min \max f = \max \min f$$

如果f是凸优化问题，强对偶性成立。而我们之前求的KKT条件是强对偶性的充要条件。

SVM优化

我们已知SVM优化的主问题是：

$$\min_w \frac{1}{2}||w||^2$$

$$s.t. \quad g_i(w,b) = 1 - y_i（w^T x_i + b） \leq 0, \quad i = 1,2,\ldots,n$$

那么求解线性可分的SVM的步骤为：

步骤1：

构造拉格朗日函数：

$$\min_{w,b} \max_\lambda L(w,b,\lambda) = \frac{1}{2}||w||^2 + \sum_{i=1}^n \lambda_i[1 - y_i(w^T x_i + b)]$$

$$s.t. \quad \lambda_i \geq 0$$

步骤2：

利用强对偶性转化：

$$\max_\lambda \min_{w,b} L(w,b,\lambda)$$

现对参数w和b求偏导数：

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^n \lambda_i x_i y_i = 0$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^n \lambda_i y_i = 0$$

得到：

$$\sum_{i=1}^n \lambda_i x_i y_i = w$$

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

我们将这个结果带回到函数中可得：

$$\begin{aligned} L(w,b,\lambda) &= \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^n \lambda_i - \sum_{i=1}^n \lambda_i y_i (\sum_{j=1}^n \lambda_j y_j (x_i \cdot x_j) + b) \\ &= \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^n \lambda_i - \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^n \lambda_i y_i b \\ &= \sum_{j=1}^n \lambda_i - \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j (x_i \cdot x_j) \end{aligned}$$

也就是说：

$$\min_{w,b} L(w,b,\lambda) = \sum_{j=1}^n \lambda_i - \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j (x_i \cdot x_j)$$

步骤3：

步骤3：

由步骤2得：

$$\max_{\lambda}[\sum_{j=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j (x_i \cdot x_j)]$$

$$s.t. \quad \sum_{i=1}^{n}\lambda_i y_i = 0 \quad \lambda_i \geq 0$$

我们可以看出来这是一个二次规划问题，问题规模正比于训练样本数，

我们常用$SMO(SequentialMinimalOptimization)$算法求解。

$SMO(SequentialMinimalOptimization)$，序列最小优化算法，其核心思想非常简单：

每次只优化一个参数，其他参数先固定住，仅求当前这个优化参数的极值。

我们来看一下SMO算法在SVM中的应用。

我们刚说了SMO算法每次只优化一个参数，但我们的优化目标有约束条件：

$\sum_{i=1}^{n}\lambda_i y_i = 0$，没法一次只变动一个参数。所以我们选择了一次选择两个参数。具体步骤为：

选择两个需要更新的参数λ_i和λ_j，固定其他参数。

于是我们有以下约束：

$$\lambda_i y_i + \lambda_j y_j = c \quad \lambda_i \geq 0, \lambda_j \geq 0$$

其中$c = -\sum_{k \neq i,j}\lambda_k y_k$，由此可以得出$\lambda_j = \frac{c - \lambda_i y_i}{y_j}$，

也就是说我们可以用λ_i的表达式代替λ_j。

这样就相当于把目标问题转化成了仅有一个约束条件的最优化问题，仅有的约束是$\lambda_i \geq 0$。

2.对于仅有一个约束条件的最优化问题，我们完全可以在λ_i上对优化目标求偏导，

令导数为零，从而求出变量值$\lambda_{i_{new}}$，

然后根据$\lambda_{i_{new}}$求出$\lambda_{j_{new}}$。

3.多次迭代直至收敛。

通过SMO求得最优解λ^*。

步骤4：

我们求偏导数时得到：

$$w = \sum_{i=1}^{m}\lambda_i y_i x_i$$

由上式可求得w。

我们知道所有$\lambda_i > 0$对应的点都是支持向量，

我们可以随便找个支持向量，然后带入：$y_s(wx_s + b) = 1$，求出b即可，

两边同乘y_s，得$y_s^2(wx_s + b) = y_s$

因为$y_s^2 = 1$，所以：$b = y_s - wx_s$

为了更具鲁棒性，我们可以求得支持向量的均值：

$$b = \frac{1}{|S|}\sum_{s \in S}(y_s - wx_s)$$

步骤5：w和b都求出来了，我们就能构造出最大分割超平面：$w^T x + b = 0$

分类决策函数：$f(x) = sign(w^T x + b)$

其中$sign(\cdot)$为阶跃函数：

$$sign(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}$$

将新样本点导入到决策函数中既可得到样本的分类。

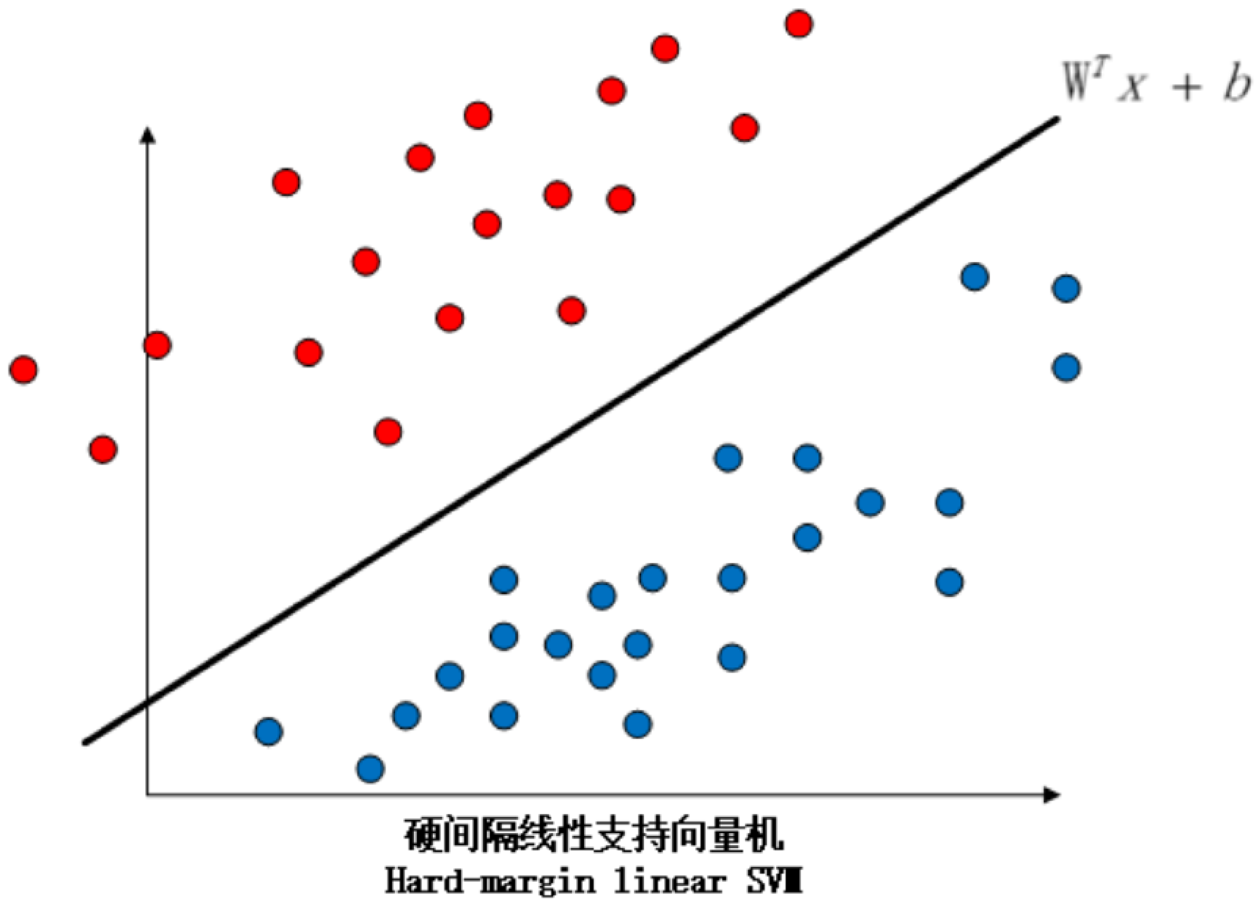
软间隔问题

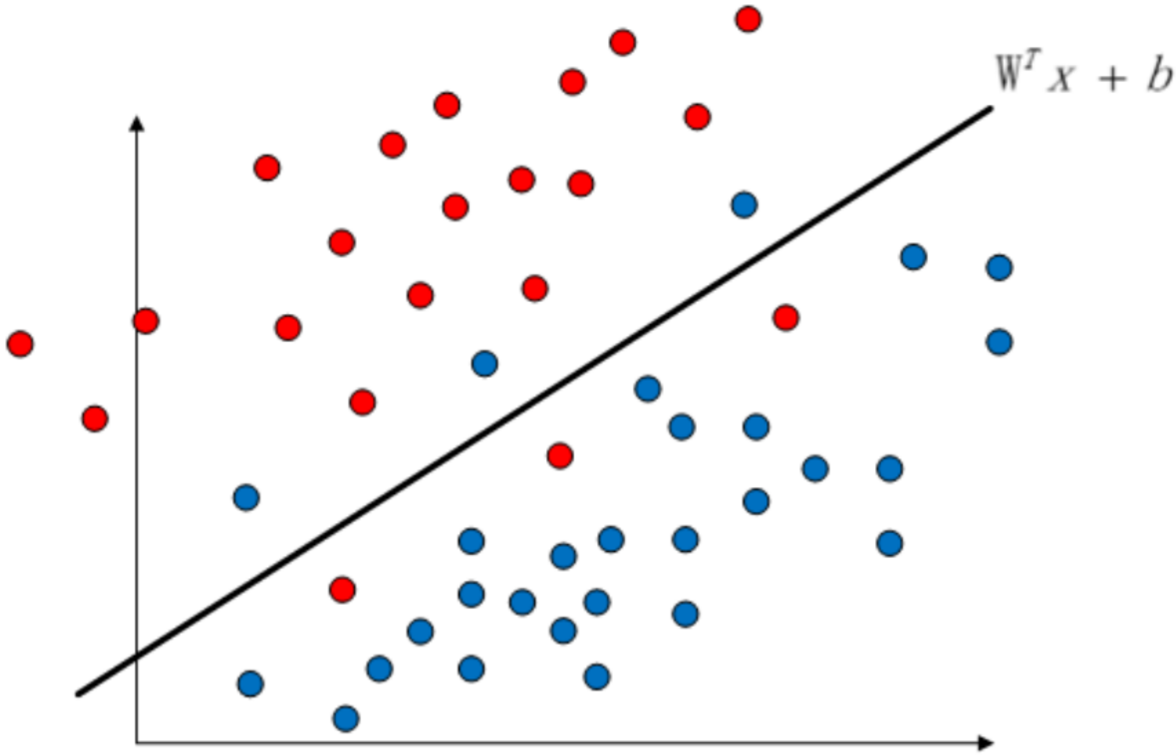
到这里都是基于训练集数据线性可分的假设下进行的，但是实际情况下几乎不存在完全线性可分的数据，

为了解决这个问题，引入了“软间隔”的概念，即允许某些点不满足约束：

$1-y_i(w^Tx_i+b)\le 0$

硬间隔和**软间隔**的区别如下图所示：





软间隔线性支持向量机
Soft-margin linear SVM

为了度量这个间隔软到何种程度，我们为每个样本引入一个松弛变量 ξ_i，

令$\xi_i \geq 0$，且$1 - y_i(w^T x_i + b) - \xi_i \leq 0$。

求解软间隔问题

增加软间隔后我们的优化目标变成了：

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

$$s.t. \quad g_i(w,b) = 1 - y_i(w^Tx_i + b) - \xi_i \le 0, \quad \xi_i \ge 0, \quad i = 1,2,\ldots,n$$

其中C是一个大于0的常数，可以理解为错误样本的惩罚程度，若C为无穷大，ξ_i必然无穷小，

如此一来线性SVM就又变成了线性可分SVM；当C为有限值的时候，才会允许部分样本不遵循约束条件。

接下来我们将针对新的优化目标求解最优化问题：

步骤1：

构造拉格朗日函数：

$$\min_{w,b,\xi}\max_{\lambda,\mu} L(w,b,\xi,\lambda,\mu) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i + \sum_{i=1}^{n}\lambda_i[1 - \xi_i - y_i(w^Tx_i + b)] - \sum_{i=1}^{n}\mu_i\xi_i$$

$$s.t. \quad \lambda_i \ge 0 \quad \mu_i \ge 0$$

其中λ_i和μ_i是拉格朗日乘子，w、b和ξ_i是主问题参数。

根据强对偶性，将对偶问题转换为：

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

步骤2：

分别对主问题参数w、b和ξ_i求偏导数，并令偏导数为0，得出如下关系：

$$w = \sum_{i=1}^{m}\lambda_i y_i x_i$$

$$0 = \sum_{i=1}^{m}\lambda_i y_i$$

$$C = \lambda_i + \mu_i$$

将这些关系带入拉格朗日函数中，得到：

$$\min_{w,b,\xi} L(w,b,\xi,\lambda,\mu) = \sum_{j=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j(x_i \cdot x_j)$$

最小化结果只有λ而没有μ，所以现在只需要最大化λ就好：$\max\limits_{\lambda}[\sum_{j=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j(x_i \cdot x_j)]$

$$s.t. \quad \sum_{i=1}^{n}\lambda_i y_i = 0, \quad \lambda_i \ge 0, \quad C - \lambda_i - \mu_i = 0$$

我们可以看到这个和硬间隔的一样，只是多了个约束条件。

然后我们利用SMO算法求解得到拉格朗日乘子λ^*。

步骤3：

$$w = \sum_{i=1}^{m}\lambda_i y_i x_i$$

$$b = \frac{1}{|S|}\sum_{s \in S}(y_s - wx_s)$$

然后我们通过上面两个式子求出w和b，最终求得超平面$w^Tx + b = 0$，

这边要注意一个问题，在间隔内的那部分样本点是不是支持向量？

我们可以由求参数w的那个式子可看出，只要$\lambda_i > 0$的点都能够影响我们的超平面，因此都是支持向量。

非线性SVM算法

对于输入空间中的非线性分类问题，可以通过非线性变换将它转化为某个维特征空间中的线性分类问题，在高维特征空间中学习线性支持向量机。

由于在线性支持向量机学习的对偶问题里，目标函数和分类决策函数都只涉及实例和实例之间的内积，所以不需要显式地指定非线性变换，

而是用核函数替换当中的内积。核函数表示，通过一个非线性转换后的两个实例间的内积。

具体地，$K(x,z)$是一个函数，或正定核，意味着存在一个从输入空间到特征空间的映射$\phi(x)$，对任意输入空间中的x,z，有

$$K(x,z)=\phi(x)\cdot\phi(z)$$

在线性支持向量机学习的对偶问题中，用核函数$K(x,z)$替代内积，求解得到的就是非线性支持向量机

$$f(x)=sign\left(\sum_{i=1}^{N}\alpha_i^*y_iK(x,x_i)+b^*\right)$$

综合以上讨论，我们可以得到非线性支持向量机学习算法如下：

输入：训练数据集$T=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_N,y_N)\}$

其中，$\boldsymbol{x}_i\in\mathbb{R}^n$，$y_i\in\{+1,-1\},i=1,2,\ldots N$；

输出：分离超平面和分类决策函数

（1）选取适当的核函数$K(x,z)$和惩罚参数$C>0$，构造并求解凸二次规划问题

$$\min_{\boldsymbol{\alpha}}\ \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_jy_iy_jK(\boldsymbol{x_i},\boldsymbol{x_j})-\sum_{i=1}^{N}\alpha_i$$

$$s.t.\quad\sum_{i=1}^{N}\alpha_iy_i=0$$

$$0\le\alpha_i\le C,\ i=1,2,\ldots,N$$

得到最优解$\boldsymbol{\alpha}^*=(\alpha_1^*,\alpha_2^*,\ldots,\alpha_N^*)^T$

（2）计算

选择$\boldsymbol{\alpha}^*$的一个分量α_j^*满足条件$0<\alpha_j^*<C$，计算

$$b^*=y_j-\sum_{i=1}^{N}\alpha_i^*y_iK(\boldsymbol{x_i},\boldsymbol{x_j})$$

（3）分类决策函数：

$$f(x)=sign\left(\sum_{i=1}^{N}\alpha_i^*y_iK(x,x_i)+b^*\right)$$

介绍一个常用的核函数——高斯核函数

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

对应的SVM是高斯径向基函数分类器，在此情况下，分类决策函数为

$$f(x)=sign\left(\sum_{i=1}^{N}\alpha_i^*y_i\exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right)+b^*\right)$$