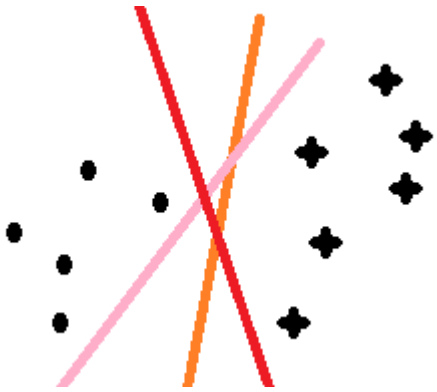
一、 SVM 算法简介

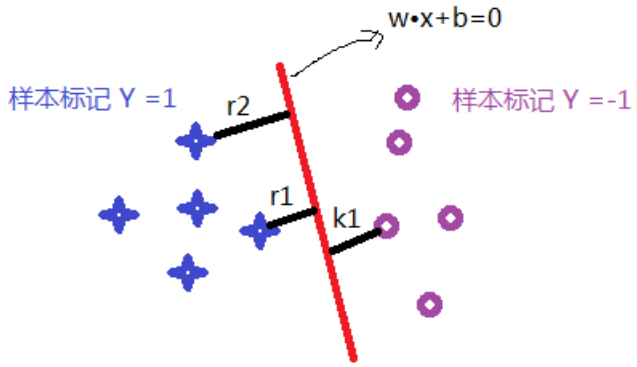
支持向量机（support vector machines, SVM）是一种二分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器，间隔最大使它有别于感知机；SVM 还包括核技巧，这使它成为实质上的非线性分类器。SVM 的的学习策略就是间隔最大化，可形式化为一个求解凸二次规划的问题，也等价于正则化的合页损失函数的最小化问题。SVM 的的学习算法就是求解凸二次规划的最优化算法。

二、 SVM 算法原理



上图中点集合和星集合可以很容易地由一条直线分类。当实例从二维空间扩展到多维空间时，将两个集合能够完全正确地划分开的直线就变成了一个超平面。但是，就像上图一样，能够正确区分开二维空间中两个集合的直线有很多，能够正确区分开多维空间的集合的超平面也有很多。那么究竟哪个超平面才是最佳的超平面呢？这就是我们需要研究的问题。

在 SVM 算法中，我们的目标是寻找到一个超平面，使得距离超平面比较近的这些点，到超平面的距离能够尽量大。也就是说，我们不需要考虑所有的点都远离超平面，我们关心的是求得的超平面能够让所有点中距离它最近的点有最大的间距，如下图所示。



假设蓝色的星星类有 5 个样本，并设定此类样本标记为 Y=1，紫色圈类有 5 个样本 ,并设定此类标记为 Y =-1 ,共 $T=\{(X_1,Y_1),(X_2,Y_2)(X_3,Y_3).....\}$ 10 个样本，超平面（分割线）为 $wx+b=0$; 样本点到超平面的几何距离为：

$$\frac{y_i \cdot y(x_i)}{\|w\|} = \frac{y_i \cdot (w^T \cdot \varphi(x_i) + b)}{\|w\|}$$

接下来就是如何去获取超平面。超平面需要满足支持向量到其最小的距离最大。也就是取支持向量到超平面最小距离的最大值。那么就先需要获得支持向量到超平面最小距离的表达公式。这个公式如下：

$$max_{w,b}[min_{x_i}\frac{y_i(w \cdot x_i + b)}{\|w\|}]$$

根据这个公式我们可以看出 ,通过优化 w 和 b 就可以最大化支持向量到分割超平面之间的距离。但是直接优化这个式子是比较困难的，需要尝试采用其他的方法来进行优化。

我们的目标是最大化数据点到超平面的间隔，那么把最小化的部分，也就是寻找支持向量放到约束条件里，就可以得到

$$argmax_{w,b}\frac{\hat{\gamma}}{\|w\|}$$

其中$\hat{\gamma}$是支持向量到超平面的函数间隔。把参数 w 和 b 都除以$\hat{\gamma}$，就可以得到

$$argmax_{w,b}\frac{1}{\|w\|}$$

这时最大化问题就转化为了求最小值

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

这是一个线性不等式约束下的优化问题，可以依据拉格朗日乘数法来获取优化目标的对偶形式。通过求解对偶问题得到最优解，这是线性可分条件下支持向量机的对偶算法。这样做的好处是对偶问题一般更容易求解，另外可以引入核函数，进而推广到非线性分类问题。

一般化的拉格朗日公式：

$$L(w,\alpha,\beta)=f(w)+\sum_{i=1}^{k}\alpha_i g_i(w)+\sum_{i=1}^{l}\beta_i h_i(w)$$

其中的α_i β_i都是拉格朗日算子。

设$\theta_p(w)=max\ _{\alpha,\beta:\alpha_{i\geq 0}}L(w,\alpha,\beta)$；

设约束条件$g_i(w)\leq 0, i=1,2,\dots,k; h_i(w)=0, i=1,2,\dots,l$

当条件不全部满足的时候，可以通过调整α_i β_i使得最大值出现正无穷，因此我们可以得到

$$\theta_p(w)=\begin{cases} f(w) \\ \infty \end{cases}$$

这样我们之前需要求的 min f（w）就可以转化为求

$$min_w\theta_P(w)=min_w L(w,\alpha,\beta)$$

再设$\theta_D(\alpha,\beta)=min_w L(w,\alpha,\beta)$,通过$\theta_D(\alpha,\beta)$可以把问题转化为求拉格朗日关于 w 的最小值。那么先把(α,β)都看作常量，然后再求最大值，这就是原问题的对偶问题。

SVM 算法中另外一个重要的概念就是 KKT 条件 ,也就是 Karush-Kuhn-Tucker(KKT)条件。这个条件是对于有不等式和等式约束的最优化问题有局部最优解的必要条件。该条件描述如下：

设拉格朗日函数为：$\sqrt{x}L(x,\mu_1,\mu_2)=f(x)+\mu_1 g_1(x)+\mu_2 g_2(x)$

则 KKT 条件（具有局部最优解的必要条件）为：

1. $\mu_1 \geq 0$，$\mu_2 \geq 0$
2. $\nabla f(x)+\mu_1 \nabla g_1(x)+\mu_2 \nabla g_2(x)=0$
3. $\mu_1 g_1(x)+\mu_2 g_2(x)=0$

拉格朗日乘数法和 KKT 条件都是要应用于优化 SVM 目标函数上，也就是$argmin_{w,b}\frac{1}{2}\|w\|^2$的优化。它受制于

$$1+y_i\cdot(wx_i^T+b)\leq 0, i=1,2,\dots,k$$

对应的拉格朗日函数：

$$L(w,\alpha,\beta)=\frac{1}{2}\|w\|^2-\sum_{i-1}^{N}\alpha_i[y_i(w^T x_i+b)-1]$$

其中$\alpha_i \geq 0$

令$g_i(w,b)=-y_i(w^T x_i+b)+1$,则有$g_i(w,b)$ ≤0

- 如果$\alpha_i \geq 0$,根据 KKT 条件 ,$\alpha_i g_i(w,b)=0$,可以推出$g_i(w,b)=0$，则约束g_i ≤0 是一个有效约束，对应的x_i就是支持向量；
- 如果$\alpha_i=0$,$g_i(w,b)<0$, $g_i(w,b)\leq 0$是一个不起作用的约束 ,对应的x_i就不是支持向量

三、 SVM 中的核函数问题

SVM 算法显然是一个线性的分类器 ,但是如果数据是线性不可分的应该怎么办呢？这时我们一般采取将数据映射到高维空间的策略，这样就很可能变成线性可分的。但是映射到高维空间以后也会带来新的问题，就是在高维空间上求解出一个带约束的优化问题会比在低维空间上计算量大得多，这就是所谓的维度灾难。为了解决这一问题就需要用到核函数。

假设有一个黑匣子，输入原始数据维度下的两个坐标向量，然后经过黑匣子这么一圈，出来一个值，这个值我们就认为是高维度下的值。而黑匣子的潜在意义就相当于一个高维映射器一样。更重要的是我们并不需要知道黑匣子究竟是怎么映射的，只需要知道它的低纬度下的形式就可以了。常用的黑匣子就是径向基函数，而这个黑匣子在数学上就叫做核函数。

核是一个函数 K，对所有$x, z \in X$，满足$K(x,z) =< \Phi(x) \cdot \Phi(z) >$，这里的$\Phi$是从 X 到内积特征空间 F 的映射。

核函数的优势在于，一个是映射到高维空间中，然后再根据内积的公式进行计算；而另一个则直接在原来的低维空间中进行计算，而不需要显式地写出映射后的结果。这样我们就可以看出，它能够简化映射空间中的内积运算，避免了在高维空间中进行的计算，但运算后的结果却是等价的。常用的核函数有多项式核、高斯核和线性核等。

四、 SVM 算法优缺点总结

优点

- 有严格的数学理论支持，可解释性强，不依靠统计方法
- 能找出对任务至关重要的关键样本（支持向量）
- 采用核技巧以后可以处理非线性分类任务
- 计算的复杂度取决于支持向量的数目，可以避免维度灾难

缺点

- 训练时间长，时间开销大
- 采用核技巧并且存储核矩阵时，空间复杂度大

- 支持向量的数量较大时，预测计算复杂度高
- 适合小批量样本的任务，无法适应百万级以上样本的任务