<http://www.cnblogs.com/subconscious/p/4249581.html>

ω就是超平面ωTx+b=0的法向量。推导很简单：ωTx+b=0与ωTy+b=0相减即可，ωT(x-y)=0。这里将x，y视为超平面上的任意点，x-y即是超平面上的任意向量。

反之对于超平面的任意法向量，它必有一个对应的常量cn使得：ωnTx+cn=0。ωT(x-y)=0推出ωTx=ωTy，固定y让x任意变即得ωTx=c

<https://blog.csdn.net/u014433413/article/details/78427574>

# 【机器学习】支持向量机SVM原理及推导

2017年11月02日 17:54:03

阅读数：3894

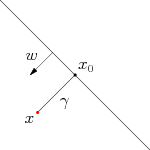
参考：http://blog.csdn.net/ajianyingxiaoqinghan/article/details/72897399 部分图片来自于上面博客。

# 0 由来

在二分类问题中，我们可以计算数据代入模型后得到的结果，如果这个结果有明显的区别，这就说明模型可以把数据分开。那么，怎么表示“区别”这个词呢，拿最简单的二维问题来讲，“区别”可以是数据点分布在一条直线的两侧，而数据点代入方程后得到的结果符号是不同的，这就达到了分类的目的。 而SVM的思想也是这样，目的就是找到一个超平面，将数据点都正确地分在超平面的两侧。那么，又怎么表示这个“都正确”呢？可以这样考虑：就是让那些“很有可能不正确”的数据点彼此分开得明显一点就可以了。对于其它“不那么可能不正确”或者说“一看就很正确”的数据点，就可以不用管了。这也是SVM名称的由来，模型是由那些支持向量（Support Vector）决定的。这也是为什么SVM对outlier不敏感。

# 1 间隔

遵循上面的逻辑，我们去假设空间里找模型了。但是一下子出来好多个模型都符合我们的要求，怎么办？自然我们想要找到“最优”的那一个模型。那么，怎么衡量这个“最优”呢？根据【超平面】【数据点】【分开】这几个词，我们可以想到最优的模型必然是最大程度地将数据点划分开的模型，不能靠近负样本也不能靠近正样本，要不偏不倚，并且与所有Support Vector的距离尽量大才可以。这就引出了间隔的讨论。



上图中 x0是 x 在超平面上的投影，ω

是超平面的法向量，二者平行可以得到：

x−x0=γω / ||ω|| (1.1)

两边同乘 ωT 并利用 ωTx0+b=0,ωTω=||ω||2 得到：

γ=(ωT+b) / ||ω||=f(x) / ||ω|| (1.2)

当然，上式是带正负号的，如果要得到正值，即点到超平面的距离，乘上数据点的类别就好：

γ~=yγ (1.3)

# 2 最大间隔分类器

上面我们推导出了间隔的表达式，自然的，我们想让数据点离超平面越远越好。



回顾一下，在这样的模型中，我们只考虑那些支持向量就可以了，对于那些显然可以分类成功的数据点，我们顺带着讨论它们就可以。   
 不妨令那些“有可能分类不成功的点”，即靠近超平面的点，分布在超平面 ωTx+b=±1上，这里的取值 1 只是为了方便推导，后面我们可以看到，这个值不影响最后的优化过程。 （可以证明一定存在这样的ω，b满足这一假设。上面的支持向量所在超平面记为ωTx=c1，下面的记为ωTx=c2，则中间的分隔超平面为：ωTx=（c1+c2）/ 2，三个超平面两边同时减去（c1+c2）/ 2并记b=-(c1+c2) / 2整理得：ωTx+b=(c1-c2) / 2，ωTx+b=(c2-c1) / 2，ωTx+b=0。ω向量的长度变为原来的2/(c1+c2)倍即可。）  
 这样，支持向量到达我们要优化的超平面 ωTx+b=0 的距离就是 1 / ||ω||，两侧的距离加起来就是2 / ||ω||，同时，我们要求模型对正负样本要做到“不偏不倚”，对于这一条，我们加上限制条件 y(ωT+b)⩾1就好。（这个也是可以证明的：上面选择的支持超平面保证了y和ωT+b符号一致。）于是我们得到了不等式约束优化问题：

max(2 / ||ω||) s.t. yi(ωTxi+b)⩾1, i=1,2,...,m (2.1)

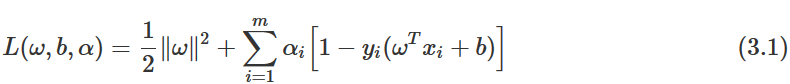
为了方便推导，上式可以等价地写成：

min( ||ω||2 / 2) s.t. yi(ωTxi+b)⩾1, i=1,2,...,m (2.2)

# 3 拉格朗日乘子法对偶问题

(2.2) 的优化目标是二次的，约束是线性的，整体是一个凸二次规划问题。有现成的优化包可以求解。但将其转化为拉格朗日对偶问题后求解更容易，也方便我们后面引入核函数。

对式 (2.2) 的每一个不等式约束条件（m个数据点共有m个不等式）设置对应的拉格朗日乘子 αi>0，得到**原始问题**的拉格朗日函数：



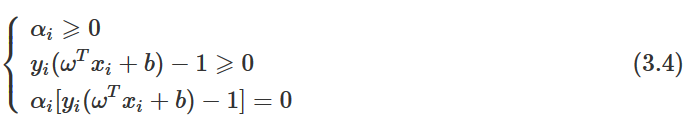
目标是让拉格朗日函数 L(ω,b,α) 针对 α 达到最大值。为什么能够这么写呢，我们可以这样想，哪怕有一个 yi(ωTxi+b)⩾1 不满足，只要让对应的 αi 是正无穷就好了。所以，如果L(ω,b,α) 有有限的最大值，那么那些不等式条件是自然满足的。 之后，我们再让 L(ω,b,α) 针对 ω,b 达到最小值，就可以了。 从而，我们的目标函数变成：



为方便求解，我们将 min 和 max 的位置交换一下：



可以证明， d∗⩽p∗ ，可以通过求解 d∗ 而近似求解 p∗ 。(3.3)即为原始问题的对偶问题。 由于原始优化问题中有不等式条件，这里的对偶问题需要满足下面形式的KKT条件才能有解：

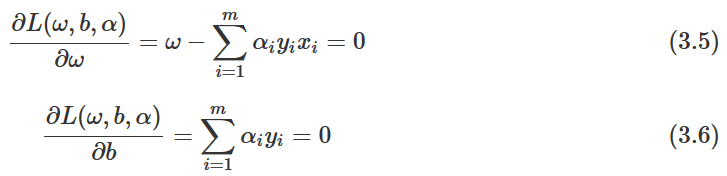


这里知道到我们到目前为止的推导是需要满足KKT条件的就好了。

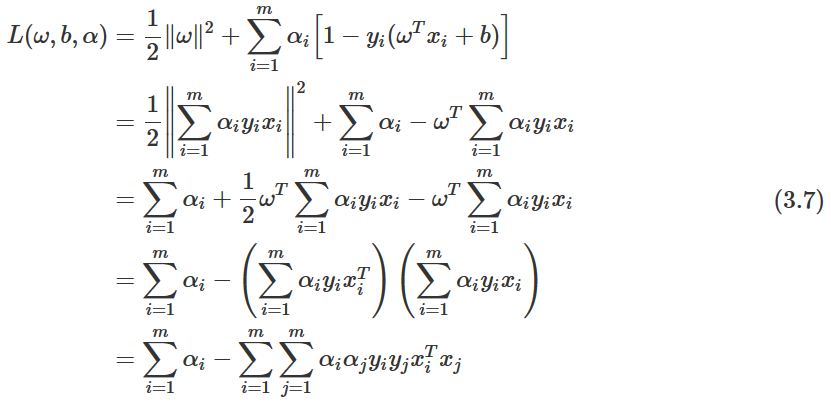
下面我们看一下具体怎样求解对偶问题 (3.3) ：

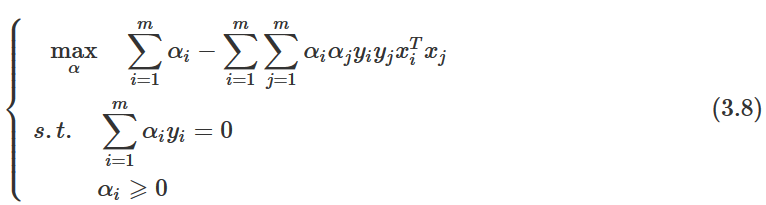
## 3.1 ω,b的部分

我们先看一下 minω,b的部分。分别令 L(ω,b,α) 对 ω,b的导数等于0：



将（3.5）（3.6）代入（3.1）：

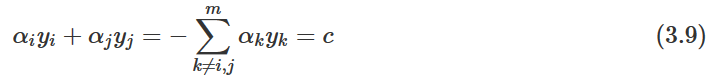
问题就变为了：



## 3.2 α的部分

得到（3.8）之后怎么求解呢？不难发现这是一个二次规划问题。,但如果样本数过多，则计算量过大。SMO是高效求解这个问题的算法代表。

我们选取两个变量 αi,αj，（3.8）中的其它变量保持固定。得到：



可以将 αj 消掉，只保留 αi ，（3.8）就变成了关于 αi 的单变量二次规划问题，约束是 αi⩾0 ，有\*\*闭式解\*\*。这样算起来肯定快。 那么，怎样找这两个变量 αi,αj 比较好呢？第一个变量 αi 我们肯定选取那个最不满足KKT条件的 α ，第二个我们需要选让目标函数增长得最多的 α ，但每次计算太过麻烦。所以有一个启发式的方法：我们选取那个与 αi 所对应的样本间隔最大的样本所对应的 α 作为 αj 。这样与更新两个相似的样本相比，目标函数值变化更大。这里的具体推导可以参考 http://blog.csdn.net/ajianyingxiaoqinghan/article/details/73087304 。具体来说，我们可以用\*\*预测值与真实值之差\*\*来衡量这个“样本间的差异”，如果 αi 对应的样本预测值与真实值之差为负的，那么我们就尽量找一个差值为正的且绝对值较大的，反之我们就找一个差值为负的且绝对值较大的。在几何上可以理解为找那些暂时被分类错误的样本。 当然，如果得到的 αj 不能使函数值下降很多，那么我们还可以干脆就暴力找一个让函数值下降最多的 αj ，或者再找一个不符合KKT条件的 αj 当做第二个 α 。 求解出 αnewi αnewj 之后，由KKT条件(3.4)可得，若 αnewi>0 ，则：

yi(∑i=1mαiyixi+b)=1(3.10)

就可以对 b 进行更新，更新之后将预测值与真实值之差的列表更新一遍，以供下次循环使用。 当然，如果 *αnewj*>0 ，也可以计算出相应的 b 值，如果两个 b 值相等就取该值。 如果不相等就取平均值。 还有一种做法就是，对目前模型中所有的 *α*>0 ，都计算一个 b ，取平均值。 等到算法停止，b 也计算好了， *ω* 由（3.5）式得：

*ω*=∑*i*=1*mαiyixi*(3.11)

这样，原始优化问题就解完了。

# 4 核函数

在前面的讨论中，我们假设数据集是线性可分的。但是现实任务中，可能并不存在一个超平面将数据集完美得分开。   
这种情况下，我们可以通过将原始空间映射到一个高维空间，如果高维空间中数据集是线性可分的，那么问题就可以解决了。   
这样，超平面变为：

*ωTϕ*(*x*)+*b*=0(4.1)

经过像前面的一顿推导之后我们得到：

⎧⎩⎨⎪⎪⎪⎪⎪⎪⎪⎪⎪⎪⎪⎪⎪⎪max*αs*.*t*.∑*i*=1*mαi*−∑*i*=1*m*∑*j*=1*mαiαjyiyjϕ*(*xi*)*Tϕ*(*xj*)∑*i*=1*mαiyi*=0*αi*⩾0(4.2)

可见，需要计算 *ϕ*(*xi*)*Tϕ*(*xj*) ，但很多时候，我们并不知道高维空间是什么样子，也就是我们根本连 *ϕ*(*x*) 是什么样子都不知道，更不要说如果高维空间维数很大， *ϕ*(*xi*)*Tϕ*(*xj*) 计算十分困难。 其实 *ϕ*(*xi*)*Tϕ*(*xj*) 只是一个实数，如果将它们看成一个整体，它也是关于 *xi*,*xj* 的一个函数，所以，如果存在那么一个神奇的函数 *κ*(*xi*,*xj*)=*ϕ*(*xi*)*Tϕ*(*xj*) ，我们就可以在低维空间计算出高维空间的点积结果。这个函数*κ*(*xi*,*xj*) 就叫做\*\*核函数\*\*。 常用的核函数有：

|  |  |  |
| --- | --- | --- |
| **名称** | **表达式** | **参数** |
| 线型核 | *κ*(*xi*,*xj*)=*xTixj* |  |

|  |  |
| --- | --- |
|  |  |
| 多项式核 | *κ*(*xi*,*xj*)=(*xTixj*)*d* |

|  |  |
| --- | --- |
|  | *d*⩾1 |

|  |
| --- |
| 为多项式次数 |
| 高斯核 | *κ*(*xi*,*xj*)=exp(−∥*xi*−*xj*∥22*σ*2) |

|  |  |
| --- | --- |
|  | *σ*>0 |

|  |
| --- |
| 为高斯核的带宽 |
| 拉普拉斯核 | *κ*(*xi*,*xj*)=exp(−∥*xi*−*xj*∥*σ*) |

|  |  |
| --- | --- |
|  | *σ*>0 |

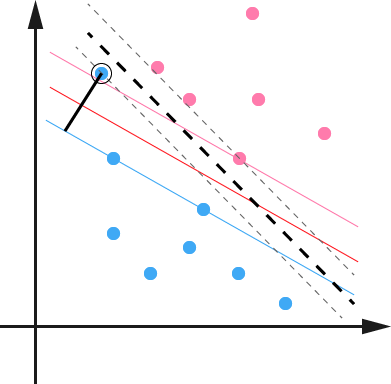
|  |
| --- |
|  |
| Sigmoid核 | *κ*(*xi*,*xj*)=tanh(*βxTixj*+*θ*) |

|  |  |
| --- | --- |
|  | *β*>0,*θ*<0tanh |

|  |
| --- |
| 为双曲正切函数 |

# 5 松弛变量

现实任务中，可能用上核函数还是不能线性可分。或者即使找到线性可分的超平面，也不能判断是不是过拟合。因此，我们将标准放宽一些，允许SVM模型在某些数据点上“出错”，为此，要引入“软间隔”：



前面的推导我们要求 *yi*(*ωTxi*+*b*)⩾1

，现在，我们将条件放宽：

*yi*(*ωTxi*+*b*)⩾1−*ξi*,*i*=1,2,...,*m*(5.1)

但同时，我们希望这个 *ξi*

尽可能小一点，越小不就越接近前面推导的线性可分么。在目标函数中体现这一点，就得到新的优化问题（对比2.2式）：

⎧⎩⎨⎪⎪⎪⎪⎪⎪⎪⎪min*s*.*t*.12∥*ω*∥2+*C*∑*i*=1*mξiyi*(*ωTxi*+*b*)⩾1,*i*=1,2,...,*mξi*>0(5.2)

C是衡量我们“放宽力度”的常数。   
与前面的推导类似，我们得到新的拉格朗日函数：

*L*(*ω*,*b*,*ξ*,*α*,*μ*)=12∥*ω*∥2+*C*∑*i*=1*mξi*+∑*i*=1*mαi*[1−*ξi*−*yi*(*ωTxi*+*b*)]−∑*i*=1*mμiξi*(5.3)

分别令 *L*(*ω*,*b*,*ξ*,*α*,*μ*)

对 *ω*,*b*,*ξi*

的导数等于0：

∂*L*(*ω*,*b*,*ξ*,*α*,*μ*)∂*ω*=*ω*−∑*i*=1*mαiyixi*=0(5.4)

∂*L*(*ω*,*b*,*ξ*,*α*,*μ*)∂*b*=∑*i*=1*mαiyi*=0(5.5)

∂*L*(*ω*,*b*,*ξ*,*α*,*μ*)∂*ξi*=*C*−*αi*−*μi*=0(5.6)

将（5.4）（5.5）（5.6）代入（5.3）得到对偶问题：

⎧⎩⎨⎪⎪⎪⎪⎪⎪⎪⎪⎪⎪⎪⎪⎪⎪max*αs*.*t*.∑*i*=1*mαi*−∑*i*=1*m*∑*j*=1*mαiαjyiyjxTixj*∑*i*=1*mαiyi*=00⩽*αi*⩽*C*(5.7)

KKT条件：

⎧⎩⎨⎪⎪⎪⎪⎪⎪*αi*⩾0,*μi*⩾0*yi*(*ωTxi*+*b*)−1+*ξi*⩾0*αi*[*yi*(*ωTxi*+*b*)−1+*ξi*]=0*ξi*⩾0,*μiξi*=0(5.8)

根据这些条件，用SMO算法求解就可以了，只是在求解相关变量的时候注意有新的范围限制。

从另一个角度观察（5.2），我们可以把 12∥*ω*∥2

看成是一个正则化项，也就是结构风险，描述了我们希望模型具有某些性质，也就是引入了先验知识。 *C*∑*mi*=1*ξi* 项是经验风险，用于描述模型与训练集的契合程度，可以把*ξi*写成一个更一般的形式：*l*(*f*(*xi*),*yi*)，上面推导的模型我们可以认为 *l* 是 hinge损失：*l*(*f*(*xi*),*yi*)=*max*(0,1−*yif*(*xi*))。   
从这里也可以看出，引入先验知识，可以减小模型求解时的搜索空间，因为我们给了它一个目标：间隔最大化。