

第20课：SVM——对偶学习算法

### 对偶问题

上一篇我们用 x 和 y 各代表一个维度，用 *z*=*f*(*x*,*y*) 和 *g*(*x*,*y*)=0 分别代表一个二元函数和一个一元函数。这样做是为了和图形对比的时候能看得清楚，为了可视化方便。

一般情况下，我们就用 *x* 代表一个函数的自变量。这个 *x* 本身可以是多维的。

而且，同一个函数可能同时既有等式约束条件，又有不等式约束条件。

#### 主问题

现在我们考虑在 *d* 维空间上有 *m* 个等式约束条件和 *n* 个不等式约束条件的极小化问题。这样的问题可以写作：

*minf*(*x*)，其中*x*为*d*维。

*s*.*t*.*hi*(*x*)=0,*i*=1,2,…,*m*;*gj*(*x*)⩽0,*j*=1,2,…,*n*

我们把上述问题称为“****原始最优化问题****”，也可以叫做“****原始问题****”或“****主问题****”。

为了解决原始问题，我们引入拉格朗日乘子 *λ*=(*λ*1,*λ*2,…,*λm*)*T* 和 *μ*=(*μ*1,*μ*2,…,*μn*)*T*，构造拉格朗日函数为：

*L*(*x*,*λ*,*μ*)=*f*(*x*)+∑*mi*=1*λihi*(*x*)+∑*nj*=1*μjgj*(*x*)

然后，再设：

Γ(*λ*,*μ*)=*infx*∈*D*(*f*(*x*)+∑*mi*=1*λihi*(*x*)+∑*nj*=1*μjgj*(*x*))

其中，*x*∈*D*，*D* 为主问题可行域；*inf*(*L*) 表示 *L* 函数的下确界，*inf*(*L*(*x*,*λ*,*μ*)) 表示小于或者等于 *L*(*x*,*λ*,*μ*) 的极大值。

*hi*(*x*)=0，因此对于任意 *λi*，必然有：

∑*mi*=1*λihi*(*x*)=0， 其中 *i*=1，2，…,*m*。

又因为 *gj*(*x*)⩽0，因此对于 *μj* 均为非负的情况：*μj*⩾0，必然有：

∑*nj*=1*μjgj*(*x*)⩽0， 其中 *j*=1，2，…,*n*。

假设 ˆ*x* 是主问题可行域中的一个点，则对于任意 *μj*⩾0,*j*=1，2，…,*n* 和任意 *λi*,*i*=1,2,…,*m*，有：

Γ(*λ*，*μ*)⩽*L*(ˆ*x*,*λ*,*μ*)⩽*f*(ˆ*x*)

我们假设主问题的最优解是 *p*∗，也就是说 *p*∗ 是 *f*(ˆ*x*) 所有取值中极小的那个。

又因为所有 ˆ*x* 对于任意 *μj*⩾0,*j*=1,2,…,*n* 和任意 *λi*,*i*=1,2,…,*m*，有：

Γ(*λ*，*μ*)⩽*f*(ˆ*x*)

因此，对于任意 *μj*⩾0,*j*=1,2,…,*n* 和任意 *λi*,*i*=1,2,…,*m* 有 Γ(*λ*，*μ*)⩽*p*∗，也就是说，Γ(*λ*,*μ*) ****是主问题最优解的下确界。****

#### 对偶函数和对偶问题

在此，****我们把** Γ(*λ*,*μ*） **称为对偶函数****。

对偶函数和目标函数最优解（极小值）的关系如下：

Γ*λ*,*μ*;*μj*⩾0(*λ*,*μ*)⩽*p*∗

由上式子得出：

*max*(Γ*λ*,*μ*;*μj*⩾0(*λ*,*μ*))⩽*p*∗

这里的：

*max*(Γ*λ*,*μ*;*μj*⩾0(*λ*,*μ*))

我们称为主问题的****对偶问题****，*λ* 和 *μ* 称为****对偶变量****。

### 强对偶性及求解对偶问题

设对偶问题的最优解为 *d*∗，显然有 *d*∗⩽*p*∗。

若 *d*∗==*p*∗，则我们将主问题和对偶问题的关系称为****强对偶性****，否则称为弱对偶性。

显然，强对偶性如果成立，我们就可以通过最优化对偶问题来达到最优化主问题的目的了。

那么什么时候强对偶性成立呢？

如果****主问题是凸优化问题****，也就是说当：

1. 拉格朗日函数中的 *f*(*x*)和*gj*(*x*) 都是凸函数；
2. *hi*(*x*) 是仿射函数；
3. 主问题可行域中至少有一点使得不等式约束严格成立。即存在 *x*，对所有 *j*，均有 *gj*(*x*)<0。

1、2、3 同时成立时，强对偶性成立。

注意：当主问题和对偶问题存在强对偶性时，存在 *x*∗，*λ*∗ 和 *μ*∗分别为主问题的解和对偶问题的解的充分必要条件是：它们满足 KKT 条件！

### 通过对偶问题求解主问题

当强对偶性成立时，为了解决主问题，我们可以这样做：

1. 构造拉格朗日函数，引入非负参数的拉格朗日算子去给目标函数加上限制；
2. 求拉格朗日函数对主变量的极小——将拉格朗日函数对主变量求偏导，令其为零后得出主变量与对偶变量的数值关系，由此把对主变量进行极小化的拉格朗日函数转化为一个对偶变量的函数；
3. 求上面第2步得出的函数对对偶变量的极大

由此一来，就****将求解主问题转化成了极大极小问题****。

下面，我们就用这个方法来求解线性可分 SVM 的目标函数。

### 线性可分 SVM 的对偶问题

#### 主问题

根据第18课，我们知道线性可分 SVM 的主问题为：

*minw*,*b*||*w*||22minw,b||w||22

*s*.*t*.*gi*(*w*,*b*)=1−*yi*(*wxi*+*b*)⩽0,*i*=1,2,...,*m*s.t.gi(w,b)=1−yi(wxi+b)⩽0,i=1,2,...,m

#### 主问题的强对偶性

我们需要判断一下线性可分 SVM 主问题是否是强对偶的。

因为：

1. *f*(*w*,*b*)=||*w*||22 是凸函数；
2. *gi*(*w*,*b*)=1–*yi*(*wxi*+*b*) 也是凸函数（没错，线性函数是凸函数！）；
3. 想想我们是如何构造不等式约束条件的——对于所有位于最大分割超平面两侧，距离最大分割超平面距离为 ||*w*|| 的辅助超平面上的点 *x*∗，有 1−*y*∗(*wx*∗+*b*)=0， 而对这两个辅助平面之外的点 *x*∗∗，则有 1–*y*∗∗(*wx*∗∗+*b*)<0。因此，主问题可行域中，至少有一点使得不等式条件严格成立。

所以，****线性可分 SVM 的目标函数可以通过求解其对偶问题来求解****。

### 使用对偶算法求解线性可分 SVM 的步骤

****步骤1：对主问题构造拉格朗日函数。****

引入拉格朗日乘子 *αi*⩾0，其中*i*=1，2，…,*m*，得到拉格朗日函数：

*L*(*w*,*b*,*α*)=12||*w*||2+∑*mi*=1*αi*[1−*yi*(*wxi*+*b*)]

****步骤2：求拉格朗日函数对于** *w***，***b* **的极小。****

注意：这里要用到一个向量求导的小知识—— 有向量 *X*=(*x*1,*x*2,...,*xn*)，则 *X* 的函数 *f*(*X*) 对 *X* 的导数为：

*df*(*X*)*dX*=(*dfdx*1,*dfdx*2,...,*dfdxn*)

因为 *w* 为向量，假设 *w*=(*w*1,*w*2,…,*wn*)，有：

||*w*||=√(*w*21+*w*22+…+*w*2*n*)

则：

||*w*||2=*w*21+*w*22+…+*w*2*n*

因此，*f*(*w*)=||*w*||22 对 *w* 求导结果为：*df*(*w*)*dw*=(*w*1,*w*2,…,*wn*)=*w*

我们先将拉格朗日函数对 *w* 和 *b* 求偏导，然后分别令两个偏导结果为0，之后得出了下列数值关系：

*w*=∑*mi*=1*αiyixi*

0=∑*mi*=1*αiyi*

将这两个等式带入拉格朗日函数，得：

*L*(*w*,*b*,*α*)=12∑*mi*=1∑*mj*=1*αiαjyiyj*(*xi*⋅*xj*)+∑*mi*=1*αi*−∑*mi*=1*αiyi*((∑*mj*=1*αjyjxj*)⋅*xi*+*b*)=∑*mi*=1*αi*−12∑*mi*=1∑*mj*=1*αiαjyiyj*(*xi*⋅*xj*)L(w,b,α)=12∑mi=1∑mj=1αiαjyiyj(xi⋅xj)+∑mi=1αi−∑mi=1αiyi((∑mj=1αjyjxj)⋅xi+b)=∑mi=1αi−12∑mi=1∑mj=1αiαjyiyj(xi⋅xj)

也就是：

*minw*,*bL*(*w*,*b*,*α*)=∑*mi*=1*αi*−12∑*mi*=1∑*mj*=1*αiαjyiyj*(*xi*⋅*xj*)

****步骤3：求** *minw*,*bL*(*w*,*b*,*α*) **对** *α* **的极大。****

也就是对偶问题：

*maxαminw*,*bL*(*w*,*b*,*α*)

*s*.*t*.∑*mi*=1*αiyi*=0

*αi*⩾0,*i*=1,2,...,*m*

又因为：

*maxαminw*,*bL*(*w*,*b*,*α*)=*maxα*[∑*mi*=1*αi*−12∑*mi*=1∑*mj*=1*αiαjyiyj*(*xi*⋅*xj*)]=*minα*[12∑*mi*=1∑*mj*=1*αiαjyiyj*(*xi*⋅*xj*)−∑*mi*=1*αi*]maxαminw,bL(w,b,α)=maxα[∑mi=1αi−12∑mi=1∑mj=1αiαjyiyj(xi⋅xj)]=minα[12∑mi=1∑mj=1αiαjyiyj(xi⋅xj)−∑mi=1αi]

因此对偶最优化问题变成了：

*minα*[12∑*mi*=1∑*mj*=1*αiαjyiyj*(*xi*⋅*xj*)−∑*mi*=1*αi*]

*s*.*t*.∑*mi*=1*αiyi*=0

*αi*⩾0,*i*=1,2,...,*m*

****步骤4：由对偶问题求** *α*1,*α*2,…,*αm***。

设：*T*(*α*1,*α*2,...,*αm*)=12∑*mi*=1∑*mj*=1*αiαjyiyj*(*xi*⋅*xj*)−∑*mi*=1*αi*

注意：上面这个函数中，*xi*,*xj*,*yi*,*yj* 都是训练样本的 *x* 和 *y* 值，都是定值我们只需带入即可，因此这是一个关于 *α*1,…,*αm* 的函数。

要最小化 *T*(*α*1,*α*2,…,*αm*)，我们可以把 *α* 看作一个向量：*α*=(*α*1,*α*2,…,*αm*)，我们要通过基于约束条件 ∑(*aiyi*）=0 最小化 *f*(*α*)，来求 *α* 的最优解 *α*∗。

我们可以对 *α*1,*α*2,…,*αm* 分别求偏导，然后令偏导为0 ，再结合约束条件来求 *α*的最优解：*α*∗=(*α*∗1,*α*∗2,…,*α*∗*m*)。

此处可以采取 SMO 算法，SMO 的具体内容我们在本文最后进行讲解，此处跳过，总之到这一步，我们已经求出了 *α*∗。

****步骤5：由** *α*∗**求** *w***。

由步骤1已知：

*w*=∑*mi*=1*αiyixi*

*xi*，*yi* 已知， *α*∗*i* 已由上一步求出，将它们带入上式，求 *w*。

****步骤6：由** *w***求** *b***。

*α*1∗,*α*2∗,…,*αm*∗ 都已经求出来了。

因为 *αi*（1−*yi*(*wxi*+*b*)）=0;*i*=1，2,…,*m* 是整体约束条件；又因为对于所有支持向量 (*xs*，*ys*)，都有 1−*ys*(*wxs*+*b*)=0，因此，所有大于0的 *α*∗*k* 所对应的 (*xk*,*yk*) 必然是支持向量。

否则，如果 *α*∗*k*>0,1−*yk*(*wxk*+*b*)<0，则 *α*∗*k*(1−*yk*(*wxk*+*b*))<0，不符合约束条件。

注意：我们再推想一下，会不会所有的 *α*∗*i* 都等于0呢？

如果那样的话，根据步骤5中的 *w* 计算公式，得 *w*=0，||*w*||=0，则 2||*w*|| 趋近正无穷，而 2||*w*|| 的物理意义是两个线性可分数据集之间的最大距离。

我们希望这个距离尽量大是希望两个集合被分得尽量清楚，而如果两个集合之间的距离都是无穷了，又怎么能说它们处在相同的特征空间里呢？

还有，我们原本定义的两个辅助超平面是 *wx*+*b*=1 和 *wx*+*b*=−1，如果 *w*=0，则 *b*=1 和 *b*=−1同时成立，这显然矛盾了。所以 *w* 肯定不为0，因此必然存在 *α*∗*k*>0。

那么既然哪些 (*x*,*y*) 对是支持向量都已经清楚了，理论上讲，我们随便找一个支持向量 (*xs*,*ys*)，把它和 *w* 带入：*ys*(*wxs*+*b*)=1，求出 *b* 即可。

*ys*(*wxs*)+*ysb*=1，两边乘以 *ys*。

*y*2*s*(*wxs*)+*y*2*sb*=*ys*，因为 *y*2*s*=1，所以：*b*=*ys*–*wxs*。

为了更加鲁棒，我们可以求所有支持向量的均值：

*b*=1|*S*|∑*s*∈*S*(*ys*−*wxs*)

****步骤7： 求最终结果****。

构造最大分割超平面：*wx*+*b*=0。

构造分类决策函数：*f*(*x*)=*sign*(*wx*+*b*)。

其中，*sign*(⋅) 全称为 Signum Function。其定义为：

sign(*x*)={−1:*x*<00:*x*=01:*x*>0

### SMO（Sequential Minimal Optimization）算法

先来看一下我们的优化目标：

*T*(*α*1,*α*2,...,*αm*)=12∑*mi*=1∑*mj*=1*αiαjyiyj*(*xi*⋅*xj*)−∑*mi*=1*αi*

*minαT*(*α*1,*α*2,...,*αm*)

*s*.*t*.∑*mi*=1*αiyi*=0

*αi*⩾0,*i*=1,2,...,*m*

一共有 *m* 个参数需优化。

这是一个典型的二次规划问题，我们可以直接用二次规划方法求解。或者，为了节约开销我们也可以用 SMO 算法。

SMO 是一种动态规划算法，它的****基本思想****非常简单：每次只优化一个参数，其他参数先固定住，仅求当前这一个优化参数的极值。

可惜，我们的优化目标有约束条件：∑(*αiyi*)=0，其中*i*=1，2，…,*m*。如果我们一次只优化一个参数，就没法体现约束条件了。

于是，我们这样做：

****1.**** 选择两个需要更新的变量 *αi* 和 *αj*，固定他们以外的其他变量。

这样，约束条件就变成了：

*αiyi*+*αjyj*=*c*,*αi*⩾0,*αj*⩾0

其中：

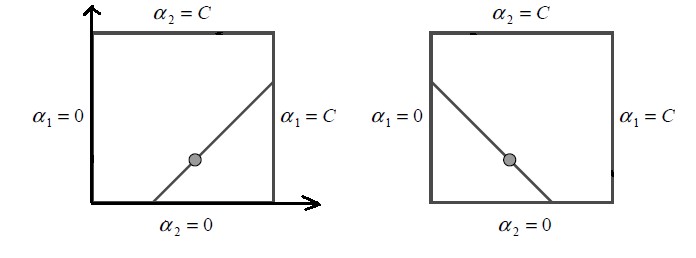
*c*=−∑*k*≠*i*,*jαkyk*

这样由此，可得出 *αj*=(*c*–*αiyi*)*yj*，也就是我们可以用 *αi* 的表达式代替 *αj*。

将这个替代式带入优化目标函数。就相当于把目标问题转化成了一个单变量的二次规划问题，仅有的约束是 *c*⩾*αi*⩾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 *αi* 上，对问题函数 *T*(*αi*) 求（偏）导，令导数为零，从而求出变量值 *αinew*。

注意：需要注意的是求出的变量值 *αinew* 还要考虑限制条件 0⩽*αi*⩽*c*，*αi* 的取值要满足直线 *αiyi*+*αjyi*=*yic* 落在 [0,*c*]×[0,*c*]的矩形中的部分（参照下图）：



检查 *αinew* 是否在这个区域内，如果不在，只能取这个区域边界上让问题函数达到相对最小的点作为新值。

确定新值后，更新 *αi* 和 *αj*。

****3.**** 多次迭代上面1-2步， 直至收敛。

SMO 算法本身还有许多值得讲得，比如：它的具体推导过程；如何选择每次 *αi*,*αj* 来提高效率等。因为本文重点是 SVM，就不赘述了。

[邀请好友一起学，获得 25% 返现奖励](https://gitbook.cn/m/mazi/columns/5bc6ac7442d7d32f50f19a98/topics/5bf28298fd72950cafdcd0a4)

[IMG_258](https://gitbook.cn/m/mazi/comp/column?columnId=5bc6ac7442d7d32f50f19a98%26tag=2#catalog)