06+SupportVectorClassification

2019 年 3 月 13 日

1 支持向量机（分类）

1.1 一、概念

支持向量机是机器学习算法中比较有代表性的一个，它的本质思想是最大间隔分类器，也即我们的任务不仅是将样本分类，而且要使样本到分类线的距离最大。支持向量机的数学推导是机器学习算法中较为复杂的一个，很多互联网算法岗面试的重点之一就是支持向量机的推导（还有快速排序和时间复杂度）。我的数据挖掘老师曾表示，当你学习一门编程语言，如果你能用它实现快速排序，那么你可以算是了解了这门语言，如果你能实现支持向量机，那你可以算是熟悉了这门语言。

支持向量机既可以用于分类任务也可以用于回归任务，在一些机器学习的 python 程序库中通常有 SVC 和 SVR 两个函数，其中 SVC 就是支持向量分类，SVR 则是支持向量回归。

1.1.1 1. 分类器

支持向量分类适用于二分类任务，其标记数据的标签不是先前的 $y \in \{0,1\}$ 而是 $y \in \{-1,1\}$。而分类器为：

$$h_{w,b} = sign(w^T x + b)$$

$$sign(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

1.1.2 2. 函数间隔和几何间隔

定义给定的数据集中的样本（x_i，y_i）和超平面（w，b）的**函数间隔**为：

$$\hat{\gamma}_i = y_i(w^T x = b)$$

而数据集和超平面的函数间隔则是数据集中所有样本和函数间隔的最小值：

$$\hat{\gamma} = \underset{i=1,\ldots,n}{max}\, \hat{\gamma}_i$$

支持向量分类的本意是最大间隔分类器，即是指希望间隔越大越好，因此当 y_i 是 1 时，希望 w^Tx+b 越大越好，当 y_i 是-1 时，希望 w^Tx+b（负数）越小越好。并且如果 $y_i(x^Tx+b)>0$ 时，预测结果才是正确的。

函数间隔可以表示预测的正确性，但并不能确切地反应样本点到超平面的距离。譬如（w,b）和（2w，2b）是同一个超平面，而函数间隔后者却是前者的二倍。

因此需要将函数距离进行**单位化**，得到的就是**几何距离**：

$$\gamma_i = y_i\left(\frac{w^T}{||w||}x_i+\frac{b}{||w||}\right)$$

整个样本的集合距离也就是：

$$\gamma = \underset{i=1,...n}{max}\gamma_i$$

值得一提的是，上述间隔均是针对正确分类的样本，也即 y_i 和 w^Tx_i+b 同号。

$\frac{b}{||w||}$ 乍一看没有道理，但（2w，b）与（2w，2b）单位化之后并不相同，而（w，b）与（2w，2b）单位化之后相同，所以单位化时只需要除以 w 的 L2 范数。

1.1.3 3. 间隔最大化

支持向量机的目的是求的一个几何间隔最大的分离超平面，可以表示为以下有约束的最优化问题：

$$\begin{aligned}&\underset{w,b}{max} \quad \gamma\\ &s.t. \quad y_i\left(\frac{w^T}{||w||}x_i+\frac{b}{||w||}\right)\geq\gamma, i=1,2,...n\end{aligned}$$

根据几何间隔和函数间隔的关系 $\gamma=\frac{\hat{\gamma}}{||w||}$

则优化问题可以化为以函数间隔为表示的等价形式：

$$\begin{aligned}&\underset{w,b}{max} \quad \frac{\hat{\gamma}}{||w||}\\ &s.t. \quad y_i\left(w^Tx_i+b\right)\geq\hat{\gamma}, i=1,2,...n\end{aligned}$$

而根据之前对函数间隔的探讨，函数间隔的大小本身对优化问题没有影响，于是可以令 $\hat{\gamma}=1$，优化问题又可以转化为等价形式：

$$\begin{aligned}&\underset{w,b}{min} \quad \frac{1}{2}||w||^2\\ &s.t. \quad y_i\left(w^Tx_i+b\right)\geq 1, i=1,2,...n\end{aligned}$$

1.1.4 4. 拉格朗日对偶问题（Lagrange duality）

先把之前的优化问题放在一边，考虑如下优化问题：

$$\begin{aligned}&\underset{w}{min} \quad f(w)\\ &s.t. \quad h_i(w)=0, i=1,2,...n\end{aligned}$$

这是微积分典型的条件极值，于是可以应用拉格朗日数乘法得到拉格朗日函数：

$$L(w,\beta)=f(w)+\sum_{i=1}^{n}\beta_i h_i(w), i=1,2,...n$$

其中系数 β_i 是拉格朗日乘子，求解拉格朗日的极值只对参数求偏导令其为零即可：

$$\frac{\partial L}{\partial w_i}=0;\ \frac{\partial L}{\partial \beta_i}=0$$

在此基础上加入含有不等式约束条件的优化问题，我们称之为**原始问题**：

$$\begin{aligned} \min_{w} \quad & f(w) \\ s.t. \quad & h_i(w)=0, i=1,2,...n \\ & g_j(w)\le 0, j=1,2,...m \end{aligned}$$

为了求解这个问题，我们定义**广义拉格朗日数乘法**：

$$L(w,\alpha,\beta)=f(w)+\sum_{j=1}^{m}\alpha_j g_j(w)+\sum_{i=1}^{n}\beta_i h_i(w)$$

设原始问题的最大值为：

$$\begin{aligned} \Theta_P(w) &= \max_{\alpha,\beta:\alpha_i\ge 0} L(w,\alpha,\beta) \\ &= \max_{\alpha,\beta:\alpha_i\ge 0} f(w)+\sum_{j=1}^{m}\alpha_j g_j(w)+\sum_{i=1}^{n}\beta_i h_i(w) \end{aligned}$$

易知，当 w 满足约束条件时，最大值为 $f(w)$ 也就是目标函数，当 w 不满足条件时最大值是 ∞。因此当我们考虑最小化 Θ_P 时我们会发现这和原始问题是同一个问题，或者是同解的。

$$\min_{w}\Theta_P(w)=\min_{w}\max_{\alpha,\beta:\alpha_i\ge 0} L(w,\alpha,\beta)$$

定义对偶问题为：

$$\Theta_D(\alpha,\beta)=\min_{w} L(w,\alpha,\beta)$$

对偶的优化问题为：

$$\max_{\alpha,\beta:\alpha_i\ge 0}\Theta_D(\alpha,\beta)=\max_{\alpha,\beta:\alpha_i\ge 0}\min_{w} L(w,\alpha,\beta)$$

由于 max(min) 总是小于等于 min(max)，譬如下面这个例子：

$$\max_{y\in\{0,1\}}\left(\min_{x\in\{0,1\}} I(x=y)\right)\le\min_{x\in\{0,1\}}\left(\max_{y\in\{0,1\}} I(x=y)\right)$$

于是就有了对偶问题的解小于等于原始问题的解：

$$d^*=\max_{\alpha,\beta:\alpha_i\ge 0}\min_{w} L(w,\alpha,\beta)\le\min_{w}\max_{\alpha,\beta:\alpha_i\ge 0} L(w,\alpha,\beta)=p^*$$

等号成立的条件便是 **KKT 条件（Karush-Kuhn-Tucker condition）**：

$$\frac{\partial}{\partial w_i}L(w,\alpha,\beta)=0, i=1,2,...,n$$
$$\frac{\partial}{\partial \beta_i}L(w,\alpha,\beta)=0, i=1,2,...,p$$
$$\alpha_i g_i(w)=0, i=1,2,...,q$$
$$g_i(w)\le 0, i=1,2,...,q$$
$$\alpha_i \le 0, i=1,2,...,q$$

值得一提的是，原始问题是先求解关于 ，的函数最大值，得到的是关于 W 的函数，再进行求解；而对偶问题是先求解关于 W 的函数，得到的是关于于 ，的函数，再进行求解。

1.1.5 5. 最优间隔分类器

根据前述的推广的拉格朗日数乘法，原始优化问题可以写成以下形式：

$$\begin{aligned}\min_{\gamma,w,b}\ & \frac{1}{2}||w||^2\\ s.t.\ & g_i(w)=-y_i(w^Tx_i+b)+1\le 0, i=1,2,...,n\end{aligned}$$

根据 KKT 条件第三个等式，事实上只有函数间隔为最小值 1 的样本点 α_i 不为 0，这些样本点即被称为支持向量。

此问题的拉格朗日函数则是：

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

拉格朗日函数对 w，b 求梯度再令其为零：

$$\nabla_w L(w,b,\alpha)=w-\sum_{i=1}^{n}\alpha_i y_i x_i$$
$$w^*=\sum_{i=1}^{n}\alpha_i y_i x_i$$
$$\frac{\partial}{\partial b}L(w,b,\alpha)=\sum_{i=1}^{n}\alpha_i y_i=0$$

将 w^* 回代到公式（22）：

$$
\begin{aligned}
L(w^*, b, \alpha) &= \frac{1}{2} w^{*T} w^* - \sum_{i=1}^{n} \alpha_i [y_i (w^T x_i + b) - 1] \\
&= \frac{1}{2} \left(\sum_{i=1}^{n} \alpha_i y_i x_i \right)^T \left(\sum_{i=1}^{n} \alpha_i y_i x_i \right) - \sum_{i=1}^{n} \alpha_i \left[y_i \left(\left(\sum_{i=1}^{n} \alpha_i y_i x_i \right)^T x_i + b \right) - 1 \right] \\
&= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j - \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i b \\
&= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j
\end{aligned}
$$

此时的优化问题变成：

$$
\begin{aligned}
&\max_{\alpha} \ W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j \\
&s.t. \ \alpha_i \geq 0, i = 1, 2, ..., n \\
&\sum_{i=1}^{n} \alpha_i y_i = 0
\end{aligned}
$$

这就印证了我们之前对对偶问题的描述，即先求关于 W 的函数的极值，得到关于 和 的函数，再求解极值。

b 的求解则是通过反证法获得：如果所有的 α_i 都等于 0，则根据公式（21）$W = 0$，分类器就变成 $f(x) = sign(W^T x + b) = sign(b)$，对于任意新的输入，输出恒为 +1 或-1。于是必有 $\alpha_i > 0$，也即支持向量一定存在。则对任意的 $\alpha_j > 0$，根据 $g_i(w)$ 的定义，$b^* = y_j - \sum_{i=1}^{n} \alpha_i y_i x_i^T x_j$。又根据先前的公式 〇 我们可以知道对 b 求导的结果恒为零，故 b 为一常数，对于任意支持向量 $b^* = y_j - \sum_{i=1}^{n} \alpha_i y_i x_i^T x_j$ 结果将是一样的。

b^* 另有一种解：

$$
b^* = -\frac{\max\limits_{i:y_i=-1} W^{*T} x_i + \min\limits_{i:y_i=1} W^{*T} x_i}{2}
$$

这是斯坦福 CS229 中给到的一种解法，根据结构推测是将一个正例得来的解和一个负例得来的解进行平均得到的。

假设我们得到了参数 W^*, b^*，当我们用支持向量机预测时：

$$
\begin{aligned}
W^{*T} x + b^* &= \left(\sum_{i=1}^{n} \alpha_i y_i x_i \right)^T x + b^* \\
&= \sum_{i=1}^{n} \alpha_i y_i x_i^T x + b^*
\end{aligned}
$$

由于之前讨论过，只有支持向量的拉格朗日乘数 α_i 大于零，其余的拉格朗日乘数均为零，所以预测时将会减少很多不必要的计算。

1.1.6 6. 核（Kernels）

除了优化求解 W 和 b 的算法外，线性可分支持向量机的推导已经全部完成，针对用非线性分类器可以完全分类的数据，则需要核方法。核方法的本质就是特征工程，将特征映射到更高维的空间上，为数据加入非线性的因素。

假设有如下映射 $\phi(x)$：

$$\phi(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}$$

则核定义如下：

$$K(x,z) = \phi(x)^T \phi(z)$$

这是从映射到核函数的方法，另有从核函数到映射的方法：

$$\begin{aligned} K(x,z) &= (x^T z)^2 \\ &= \left(\sum_{i=1}^{n} x_i z_i\right)\left(\sum_{i=1}^{n} x_i z_i\right) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j z_i z_j \\ &= \sum_{i,j=1}^{n} (x_i x_j)(z_i z_j) \end{aligned}$$

对应的映射则是：

$$\phi(x) = \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \end{bmatrix}$$

这样的映射通常是很难构造或者难以想到的，而这样的核函数却往往是在情理之中或者易于想到的，而且诸如此来的核函数不必计算映射再求向量内积，而是直接在原始向量上进行计算，于是对于高维数据而言，内存和时间得到了兼顾。

应用核方法时只需要将所有的 $X_i^T X_j$ 替换成相应的核函数 $K(X_i^T, X_j)$ 即可。

1.1.7 7. 正则化和不可分情形

无论线性还是非线性，以上讨论的都是数据完全可分的情况。然而这样的实际数据少之又少，哪怕映射到更高维的空间之后，不能完全可分的案例也比比皆是。有时即便是可以线性可分，但一些极端样本点的出现会使支持向量的几何间隔大大减小，这种情况就弱化了支持向量机最大分类间隔的本意，为了应对以上问题，我们便提出了正则化的思想，对目标函数进行一定的 ‘‘惩罚’’，优化问题就变成：

$$\begin{aligned} \min_{\gamma,w,b} \ & \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i \\ s.t. \ & y_i(w^T x + b) > 1 - \xi_i, i = 1,2,...,n \\ & \xi_i \geq 0, i = 1,2,...,n \end{aligned}$$

其中 ξ_i 是针对每个样本加的约束，它的存在允许函数间隔小于 1，同时又会使目标函数增加 $C\xi_i$，由于目标是最小化目标函数，所以这是某种意义上的 ‘‘损失’’ 或者说 ‘‘惩罚’’。C 则是一个超参数，权衡了函数间隔和目标函数的关系。C 越大，惩罚作用越强，函数间隔小于 1 的越少；C 越小，惩罚作用越弱，函数间隔小于 1 的越多。

对应的拉格朗日函数就是：

$$L(w,b,\xi,\alpha,r) = \frac{1}{2}w^T w + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i[y_i(w^T x + b) - 1 + \xi_i] - \sum_{i=1}^{n} r_i\xi_i$$

前几项好理解，最后一项 $-\sum_{i=1}^{n} r_i\xi_i$ 则是因为 ξ 本身也是变量，所以在拉格朗日函数里也需要对它进行约束。

其对偶问题为：

$$\begin{aligned} \max_{\alpha} \ & W(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j \\ s.t. \ & 0 \leq \alpha_i \leq C, i = 1,2,...,n \\ & \sum_{i=1}^{n}\alpha_i y_i = 0 \end{aligned}$$

关于 不能大于 C 可以参见拉格朗日函数，将中间两项合并，得到 $-\sum_{i=1}^{n}\{\alpha_i[y_i(w^T x + b) - 1] + (\alpha_i - C)\xi_i\}$，若 大于 C 则相当于使函数间隔大于一，这与我们施加惩罚项允许函数间隔小于 1 的本意背道而驰。

类似于可完全分类的情况下的 KKT 条件第三个等式，在非完全可分条件下也有类似的关系：

$$\begin{aligned} \alpha_i = 0 \ &\Rightarrow \ y_i(w^T x_i + b) \geq 1 \\ \alpha_i = C \ &\Rightarrow \ y_i(w^T x_i + b) \leq 1 \\ 0 < \alpha_i < 0 \ &\Rightarrow \ y_i(w^T x_i + b) = 1 \end{aligned}$$

1.1.8 7.SMO 算法

迄今为止我们已经讨论了线性可完全分类、非线性可完全分类、（非）线性不可完全分类的情况，除了如何求解 w^{*T} 之外（b 可以在此基础上求出），其他的数学推导已经完成。

最后我们要推导的即是如何求解参数。

1）坐标上升法（Coordinate ascent） 坐标上升法是一种简单直接的优化方法，对于无约束的优化问题：

$$\max_{\alpha} \ W(\alpha_1, \alpha_2, ..., \alpha_n)$$

W 是关于 的函数，坐标上升则是针对各个变量分别作为唯一变量求最值，循环至收敛：

Loop until convergence:{

```
For i = 1,...,m,{
    _i := argmax_{_i} W(_1,_2,...,_m)
}
```

}

2)SMO（sequential minimal optimization） 我们想要求解的优化问题为：

$$\max_{\alpha} \ W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j$$

$$s.t. \ 0 \leq \alpha_i \leq C, i = 1, 2, ..., n$$

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

In [ ]: