机器学习算法系列（12）：SVM（1）—线性可分支持向量机

当训练数据线性可分时，通过硬间隔最大化，学习一个线性可分支持向量机。

一般地，当训练数据集线性可分时，存在无穷个分离超平面可将两类数据正确分开。感知机利用误分类最小的策略，求得分离超平面，不过这时的解有无穷多个。线性可分支持向量机利用间隔最大化求分离超平面，解是唯一的。也就是它不仅将正负实例点分开，而且对最难分的实例点（离分离超平面最近的点）也有足够大的确信度将它们分开。这样的超平面应该对未知的新实例有很好的分类预测能力。

一、线性可分支持向量机与硬间隔最大化

1.1 线性可分支持向量机

假设给定一个特征空间上的训练数据集

$$T = \left\{ \left(x_1, y_1 \right), \left(x_2, y_2 \right), \cdot \cdot \cdot, \left(x_N, y_N \right) \right\}$$

其中$x_i \in R^n, y_i \in \{ +1, -1 \}, i = 1, 2, \cdot \cdot \cdot, N$，$x_i$为第$i$个特征向量，也称为实例，$y_i$为$x_i$的类标记，当$y_i = +1$时，称$x_i$为正例；当$y_i = -1$时，称$x_i$为负例，$(x_i, y_i)$称为样本点。再假设训练数据集是线性可分的。

给定线性可分训练数据集，通过**间隔最大化**得到的**分离超平面**为

$$w^T \cdot x + b = 0$$

以及相应的**分类决策函数**

$$f(x) = sign\left(w^T \cdot x + b \right)$$

该决策函数称为线性可分支持向量机

1.2 函数间隔与几何间隔

一般来说，一个点距离分离超平面的远近可以表示分类预测的确信程度。在超平面确定的情况下

$|w^T \cdot x + b|$能够相对地表示点x距离超平面的远近。而$w^T \cdot x + b$的符号与类标记的符号是否一致能够表示分类是否正确，所以可用$y(w^T \cdot x + b)$来表示分类的正确性与确信度，这就是**函数间隔**`functional margin`的概念

但是，函数间隔有一个不足之处，就是在选择分离超平面时，只要成比例地改变w和b，超平面并没有变化，而函数间隔却以同样比例变化了。因此，我们可以对分离超平面的法向量w加上某些约束，使得间隔确定，此时函数间隔成为**几何间隔**`geometric margin`。

对于给定的训练数据集T和超平面(w, b)，定义超平面(w, b)关于样本点(x_i, y_i)的几何间隔为

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

定义超平面(w, b)关于训练数据集T的几何间隔为超平面(w, b)关于T中所有样本点(x_i, y_i)的几何间隔之最小值，即

$$\gamma = \min_{i=1,\cdots,N} \gamma_i$$

超平面(w, b)关于样本点(x_i, y_i)的几何间隔一般是实例点到超平面的带符号的距离，当样本点被超平面正确分类时就是实例点到超平面的距离。

函数间隔与几何间隔的关系为

$$\gamma = \frac{\hat{\gamma}}{||w||}$$

若$||w|| = 1$，那么函数间隔和几何间隔相等。如果超平面参数w和b成比例地改变（超平面没有改变），函数间隔也按此比例改变，而几何间隔不变。

1.3 间隔最大化

支持向量机学习的基本想法是求解能够正确划分训练数据集并且几何间隔最大的分离超平面。几何间隔最大的分离超平面是唯一的。

间隔最大化的直观解释是：对训练数据集找到几何间隔最大的超平面意味着以充分大的确信度对训练数据记性分类。即，不仅将正负实例点分开，而且对最难分的实例点（离超平面最近的点）也有足够大的确信度将它们分开。这样的超平面应该对未知的新实例有很好的分类预测能力。

最大间隔分离超平面

接下来求一个几何间隔最大的分离超平面，即最大间隔分离超平面。具体地，可以表示为下面的约束最优化问题：

$$\max_{w,b} \gamma$$

$$s.t\ \ y_i\left(\frac{w}{||w||}\cdot x_i+\frac{b}{||w||}\right)\geqslant\gamma\ ,\ \ i=1,2,\ \cdot\ \cdot\ \cdot,N$$

即最大化超平面(w,b)关于训练数据集的几何间隔γ，约束条件表示的是超平面(w,b)关于每个训练样本点的几何间隔至少是γ

根据几何间隔和函数间隔的关系，可以将此问题改写为

$$\max\ _{w,b}\ \frac{\hat{\gamma}}{||w||}$$

$$s.t\ \ y_i\left(w\cdot x_i+b\right)\geqslant\hat{\gamma}\ ,\ \ i=1,2,\ \cdot\ \cdot\ \cdot,N$$

函数间隔$\hat{\gamma}$的取值不影响最优化的解。函数间隔因为w，b按比例改变为λw ， λb而成为$\lambda\hat{\gamma}$，但是对最优化问题中的不等式约束没有影响，对目标函数的优化也没有影响，即两者等价。这样，我们可以取$\hat{\gamma}=1$，代入后注意到最大化$\frac{1}{||w||}$和最小化$\frac{1}{2}||w||^2$是等价的，因为我们关心的并不是最优情况下目标函数的具体数值。于是就得到下面的线性可分支持向量机学习的最优化问题。

1. 构造并求解约束最优化问题：

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

$$s.t\ \ y_i\left(w\cdot x_i+b\right)-1\geqslant 0,\ \ i=1,2,\ \cdot\ \cdot\ \cdot,N$$

   求得最优解w^*,b^*
2. 由此得到分割超平面：

$$w^*\cdot x+b^*=0$$

   分类决策函数

$$f(x)=sign\left(w^*\cdot x+b\right)$$

这其实是一个凸二次规划`convex quadratic programming`问题，凸优化问题是指约束最优化问题

$$\min\ _w\ \ f(w)$$

$$s.t\ \ \ g_i(w)\leq 0\ ,\ i=1,2,\ \cdot\ \cdot\ \cdot,k$$

$$h_i(w) = 0\ ,\ i = 1, 2,\ \cdot\ \cdot\ \cdot, l$$

其中，目标函数$f(w)$和约束函数$g_i(w)$都是R^n上的连续可微的凸函数，约束函数$h_i(w)$是R^n上的仿射函数。当目标函数$f(w)$是二次函数且约束函数$g_i(w)$是仿射函数时，上述凸优化问题成为凸二次规划问题。

支持向量和间隔边界

在线性可分情况下，训练数据集的样本点中与分离超平面距离最近的样本点的实例成为支持向量`support vector`。支持向量是使约束条件式等号成立的点，即

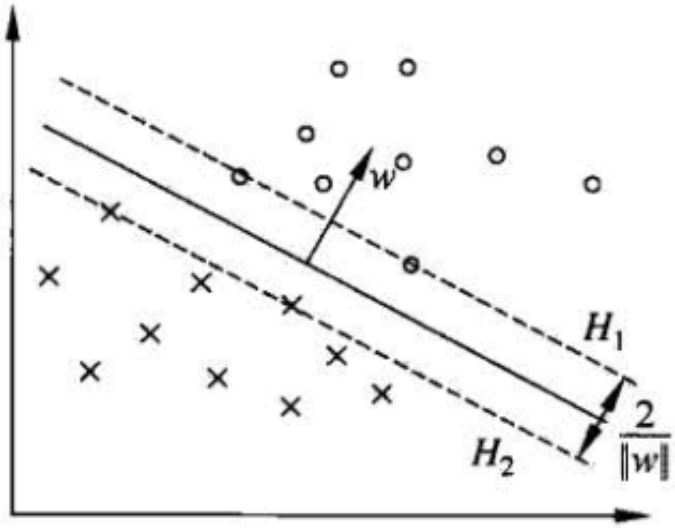
$$y_i\left(w \cdot x_i + b\right) - 1 = 0$$

对$y_i = +1$的实例点，支持向量在超平面

$$H_1:\ \ w \cdot x_i + b = 1$$

对$y_i = -1$的负例点，支持向量在超平面

$$H_2:\ \ w \cdot x_i + b = -1$$



可以看到两个支撑着中间的长带的超平面，它们到中间的分离超平面的距离相等，为什么一定是相等的呢？，即我们所能得到的最大的几何间隔$\tilde{\gamma}$。而“支撑”这两个超平面的必定会有一些点，试想，如果某超平面没有碰到任意一个点的话，那么我就可以进一步地扩充中间的 gap ，于是这个就不是最大的 margin 了。由于在 n 维向量空间里一个点实际上是和以原点为起点，该点为终点的一个向量是等价的，所以这些“支撑”的点便叫做支持向量。

注意到H_1和H_2平行，并且没有实例点落在他们中间。在H_1和H_2之间形成一条长带，分离超平面与他们平行且位于他们中间。长带的宽度，即H_1与H_2之间的距离成为间隔`margin`，间隔依赖于分离超平面的法向量w，等于$\frac{2}{||w||}$。H_1和H_2称为间隔边界。

在决定分离超平面时只有支持向量起作用，而其他实例点并不起作用。如果移动支持向量将改变所求的解；但是如果在将俄边界以外移动其他实例点，甚至去掉这些点，则解是不会改变的。由于支持向量在确定分离超平面中起着决定性作用，所以将这种分类模型称为支持向量机。支持向量机的个数一般都很少，所以支持向量机由很少的“重要的”训练样本确定。

很显然，由于这些 supporting vector 刚好在边界上，所以它们是满足 $y(w^Tx+b)=1$ ，而对于所有不是支持向量的点，也就是在“阵地后方”的点，则显然有$y(w^Tx+b)>1$ 。事实上，当最优的超平面确定下来之后，这些后方的点就完全成了路人甲了，它们可以在自己的边界后方随便飘来飘去都不会对超平面产生任何影响。这样的特性在实际中有一个最直接的好处就在于存储和计算上的优越性，例如，如果使用 100 万个点求出一个最优的超平面，其中是支持向量的有 100 个，那么我只需要记住这 100 个点的信息即可，对于后续分类也只需要利用这 100 个点而不是全部 100 万个点来做计算。

1.4 学习的对偶算法

为了求解线性可分支持向量机的最优化问题，将它作为原始最优化问题，应用拉格朗日对偶性，通过求解对偶问题得到原始问题的最优解，这就是线性可分支持向量机的对偶算法`dual algorithm`。

这样做的优点是，一是对偶问题往往更容易求解；二是引入核函数，进而推广到非线性分类问题。

首先构建拉格朗日函数，为此，对每一个不等式约束引进拉格朗日乘子$a_i \geq 0$ ， $i=1,2,\cdots,N$ 定义拉格朗日函数：

$$L(w,b,a)=\frac{1}{2}||w||^2-\sum_{i=1}^{N}a_i(y_i\Big(w\cdot x_i+b)-1\Big)$$

其中，$a=\Big(a_1,a_2,\cdots,a_N\Big)^T$
然后我们令

$$\theta(w)=max a_i\geq 0 L \qquad (w,b,\alpha)$$

容易验证，当某个约束条件不满足时，例如 $y_i(w^Tx_i+b)<1$，那么我们显然有 $\theta(w)=\infty$ （只要令$\alpha_i=\infty$即可）。而当所有约束条件都满足时，则有 $\theta(w)=\frac{1}{2}||w||_2$ ，亦即我们最初要最小化的量。因此，在要求约束条件得到满足的情况下最小化 $\frac{1}{2}||w||_2$ 实际上等价于直接最小化 $\theta(w)$

（当然，这里也有约束条件，就是 $\alpha_i \geq 0, i = 1, \ldots, n$），因为如果约束条件没有得到满足，$\theta(w)$ 会等于无穷大，自然不会是我们所要求的最小值。 具体写出来，我们现在的目标函数变成了：

$$min_{w,b}\theta(w) = \min_{w,b} \max_{a_i \geq 0} L(w, b, a) = p^*$$

这里用 p^* 表示这个问题的最优值，这个问题和我们最初的问题是等价的。不过，现在我们来把最小和最大的位置交换一下：

$$\max_{a_i \geq 0} \min_{w,b} L(w, b, a) = d^*$$

当然，交换以后的问题不再等价于原问题，这个新问题的最优值用 d^* 来表示。并，我们有 $d^* \leq p^*$，这在直观上也不难理解，最大值中最小的一个总也比最小值中最大的一个要大吧！总之，第二个问题的最优值d^* 在这里提供了一个第一个问题的最优值p^* 的一个下界，在满足某些条件的情况下，这两者相等，这个时候我们就可以通过求解第二个问题来间接地求解第一个问题。具体来说，就是要满足 KKT 条件，这里暂且先略过不说，直接给结论：我们这里的问题是满足 KKT 条件的，因此现在我们便转化为求解第二个问题。

为了得到对偶问题的解，需要先求$L(w, b, a)$对w, b的极小，再求对a的极大。

1. 求 $\min_{w,b} L(w, b, a)$
   将拉格朗日函数$L(w, b, a)$分别对w, b求偏导数并令其为0。

$$\nabla_w L(w, b, a) = w - \sum_{i=1}^{N} a_i y_i x_i = 0$$

$$\nabla_b L(w, b, a) = \sum_{i=1}^{N} a_i y_i = 0$$

   得到

$$w = \sum_{i=1}^{N} a_i y_i x_i$$

$$\sum_{i=1}^{N} a_i y_i = 0$$

   将其代入拉格朗日函数，得到

$$L(w, b, a) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j y_i y_j \left(x_i \cdot x_j \right) - \sum_{i=1}^{N} a_i y_i \left(\left(\sum_{j=1}^{N} a_j y_j x_j \right) \cdot x_i + b \right) + \sum_{i=1}^{N} a_i$$

$$= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}a_i a_j y_i y_j\left(x_i \cdot x_j\right) + \sum_{i=1}^{N}a_i$$

2. 求 $\min_{w,b} L(w, b, a)$对a的极大，即是对偶问题

$$\max_a \sum_{i=1}^{N}a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}a_i a_j y_i y_j\left(x_i \cdot x_j\right)$$

$$s.t. \sum_{i=1}^{N}a_i y_i = 0$$

$$a_i \geqslant 0, \quad i = 1, 2, \cdots, N$$

将目标函数由求极大转换为极小，就得到下面与之等价的对偶最优化问题。

$$\min_a \quad \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}a_i a_j y_i y_j\left(x_i \cdot x_j\right) - \sum_{i=1}^{N}a_i$$

$$s.t. \sum_{i=1}^{N}a_i y_i = 0$$

$$a_i \geqslant 0, \quad i = 1, 2, \cdots, N$$

让我们先来看看推导过程中得到的一些有趣的形式。首先就是关于我们的 hyper plane ，对于一个数据点 x 进行分类，实际上是通过把 x 带入到 $f(x) = w^T x + b$ 算出结果然后根据其正负号来进行类别划分的。而前面的推导中我们得到$f(w) = \sum_{i=1}^{n}\alpha_i y_i x_i$ ，因此

$$f(x) = (\sum_{i=1}^{n}\alpha_i y_i x_i)^T x + b = \sum_{i=1}^{n}\alpha_i y_i \langle x_i, x\rangle + b$$

这里的形式的有趣之处在于，对于新点 x 的预测，只需要计算它与训练数据点的内积即可（这里 $\langle\cdot,\cdot\rangle$ 表示向量内积），这一点至关重要，是之后使用 Kernel 进行非线性推广的基本前提。此外，所谓 Supporting Vector 也在这里显示出来——事实上，所有非 Supporting Vector 所对应的系数 α 都是等于零的，因此对于新点的内积计算实际上只要针对少量的“支持向量”而不是所有的训练数据即可。

为什么非支持向量对应的 α 等于零呢？直观上来理解的话，就是这些“后方”的点——正如我们之前分析过的一样，对超平面是没有影响的，由于分类完全有超平面决定，所以这些无关的点并不会参与分类问题的计算，因而也就不会产生任何影响了。

这个结论也可由刚才的推导中得出，回忆一下我们刚才通过 Lagrange multiplier 得到的目标函数：

$$max_{a_i \geq 0} L(w, b, a) = max_{a_i \geq 0} \frac{1}{2} ||w||^2 - \sum_{i=1}^{n} a_i (y_i (w^T x_i + b) - 1)$$

注意到如果 x_i 是支持向量的话，上式中$(y_i(w^T x_i + b) - 1)$部分是等于 0 的（因为支持向量的 functional margin 等于 1 ），而对于非支持向量来说，函数间隔会大于 1 ，因此这个部分是大于零的，而 a_i 又是非负的，为了满足最大化，α_i 必须等于 0 。

线性可分支持向量机学习算法

- 输入：线性可分训练数据集$T = \left\{ \left(x_1, y_1 \right), \left(x_2, y_2 \right), \cdot \cdot \cdot, \left(x_N, y_N \right) \right\}$,其中，$x_i \in R^n, y_i \in \{ +1, -1 \}, i = 1, 2, \cdot \cdot \cdot, N$
- 输出：最大间隔分离超平面和分类决策函数

步骤如下

1. 构造并求解约束最优化问题

$$\min_a \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j y_i y_j \left(x_i \cdot x_j \right) - \sum_{i=1}^{N} a_i$$

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

$$a_i \geqslant 0, \ i = 1, 2, \cdot \cdot \cdot, N$$

   求得最优解$a^* = (a_1^*, a_2^*, \cdot \cdot \cdot, a_N^*)$

2. 计算

$$w^* = \sum_i a_i^* y_i x_i$$

   并选择a^*的一个正分量$a_j^* > 0$，计算

$$b^* = y_j - \sum_{i=1}^{N} a_i^* y_i \left(x_i \cdot x_j \right)$$

3. 求得分离超平面

$$w^* \cdot x + b^* = 0$$

   分类决策函数：

$$f(x) = sign(w^* \cdot x + b^*)$$

在线性可分支持向量机中，w^* 和 b^*只依赖于训练数据中对应于$a_i^* > 0$的样本点x_i, y_i,而其他样本点对w^* 和 b^*没有影响。我们将训练数据中对应于$a_i^* > 0$的实例点$x_i \in R^n$称为**支持向量**。

对于线性可分问题，上述线性可分支持向量机的学习（硬间隔最大化）算法是完美的。但是，训练数据集线性可分是理想的情形。在现实问题中，训练数据集往往是线性不可分的，即在样本中出现噪声或特异点。此时，有更一般的学习算法。

1.5 KKT条件

对于包含等式和不等式约束的一般优化问题

$$\min f(\mathbf{x})$$
$$s.t. g_j(\mathbf{x}) \leq 0 (j = 1, 2, \cdots, m)$$
$$h_k(\mathbf{x}) = 0 (k = 1, 2, \cdots, l)$$

KKT条件（x^*是最优解的必要条件）为

$$\begin{cases} \frac{\partial f}{\partial x_i} + \sum_{j=1}^{m} \mu_j \frac{\partial g_j}{\partial x_i} + \sum_{k=1}^{l} \lambda_k \frac{\partial h_k}{\partial x_i} = 0, \ (i = 1, 2, ..., n) \\ h_k(\mathbf{x}) = 0, (k = 1, 2, \cdots, l) \\ \mu_j g_j(\mathbf{x}) = 0, (j = 1, 2, \cdots, m) \\ \mu_j \geq 0. \end{cases}$$

上式便称为不等式约束优化问题的KKT（Karush-Kuhn-Tucker）条件.μ_j称为KKT乘子，当约束起作用时$\mu_1 > 0, g_1(x) = 0$，当约束不起作用时$\mu_1 = 0, g_1(x) < 0$

更细致的推导可以看这篇文章：浅谈最优化问题的KKT条件

证明可以将线性可分支持向量机的原始问题和对偶问题等同起来的充分必要条件KKT条件，即得

$$\nabla_w L\left(w^*, b^*, a^*\right) = w^* - \sum_{i=1}^{N} a_i y_i x_i = 0$$

$$\nabla_b L\left(w^*, b^*, a^*\right) = -\sum_{i=1}^{N} a_i^* y_i = 0$$

$$a_i^* \left(y_i\left(w^* \cdot x_i + b^*\right) - 1\right) = 0 \ , \ i = 1, 2, \cdots, N$$

$$y_i\left(w^* \cdot x_i + b^*\right) - 1 \geqslant 0 \ , \ 1, 2, \cdots, N$$

$$a_i^* \geqslant 0 \ , \ i = 1, 2, \cdots, N$$

由此得

$$w^* = \sum_i a_i^* y_i x_i$$

其中至少有一个$a_j^* > 0$(反证法，假设$a^* = 0$，由上可知$w^* = 0$，而$w^* = 0$不是原始最优化问题的解，产生矛盾)，对此j有

$$y_j\left(w^* \cdot x_j + b^*\right) - 1 = 0$$

$$a_j^* y_j x_j \cdot x_i + b^* = 1/y_j = y_j$$

$$b^* = y_j - \sum_{i=1}^{N} a_i^* y_i \left(x_i \cdot x_j\right)$$

综上所述，对于给定的线性可分训练数据集，可以首先求对偶问题的解a^*;再利用求得原始问题的解w^*, b^*,从而得到分离超平面及分类决策函数。这种算法称为线性可分支持向量机的对偶学习算法，是线性可分支持向量机学习的基本算法。