支持向量机
（Support Vector Machine, SVM）

v 2.0

Grissom, 2019/07/15
http://sunbo2019.github.io

目录：

硬间隔（01）

图上是著名的泰国火车市场。假设火车道左侧摊贩是卖青菜的，右侧摊贩是卖水果的。我们现在想让无论多宽的火车都能快速通过，是不是要让两侧的摊贩离火车道越远越好？并且，还有另外一个条件：卖青菜和卖水果的摊贩必须是分别在火车道的两侧。

现在假设其他摊贩都撤到离火车道有100多米远的地方，但有那么几个“钉子户”还是贴在火车道边上摆摊，是不是火车还是没办法快速通过？也就是说，情况能否改善完全取决于离火车道两侧最近的那几个摊位。

好的，那么上面讲到的“离火车道越远越好”就是在求最大间隔。那几个“钉子户”就是SVM中的支持向量。而最大间隔能有多大，完全取决于这几个钉子户的位置。

下面，让我们从数学表达式的角度，再理解一下“求最大间隔”及“支持向量”的概念。

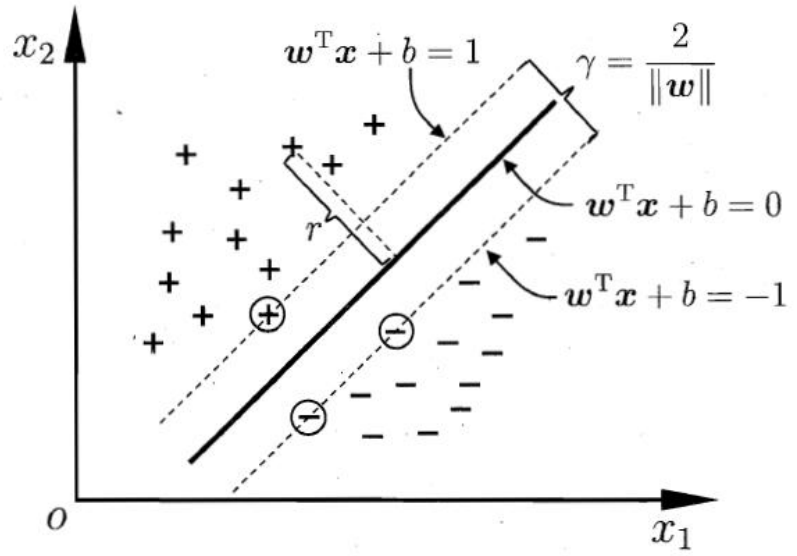
硬间隔（02）

假设我们要构建的最恰当的分类超平面是

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b=0$$

假设现在分类后，一类数据属于y=+1，一类数据属于y=-1，那么可以得到：

$$\begin{cases} \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\geqslant+1, & y_i=+1\ ; \\ \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\leqslant-1, & y_i=-1\ . \end{cases}$$



统一为一个表达式，就是：

$$y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)\geqslant 1,\quad i=1,2,\ldots,m.$$

硬间隔（03）

其实只有“使等式成立”的数据点才对生成变量w、b有意义。这几个对生成w、b的价有贡献的点，被称为“**支持向量**”（我们可以把每一个数据点理解为一个向量）。

而这几个支持向量到超平面的距离为：$\frac{2}{||\boldsymbol{w}||}$

于是，我们构造了SVM的目标函数：

$$\max_{\boldsymbol{w},b} \frac{2}{||\boldsymbol{w}||}$$

$$\text{s.t. } y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geqslant 1, \quad i = 1, 2, \ldots, m.$$

* 使支持向量到超平面的距离最远（即：使间隔最大），同时要满足“分类正确”（即”最大间隔”的约束条件）。

将求max转化为求min，于是得到：

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

$$\text{s.t. } y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geqslant 1, \quad i = 1, 2, \ldots, m.$$

拉格朗日乘子法

前面，我们将最大间隔问题转化为“带约束的求极值问题”。

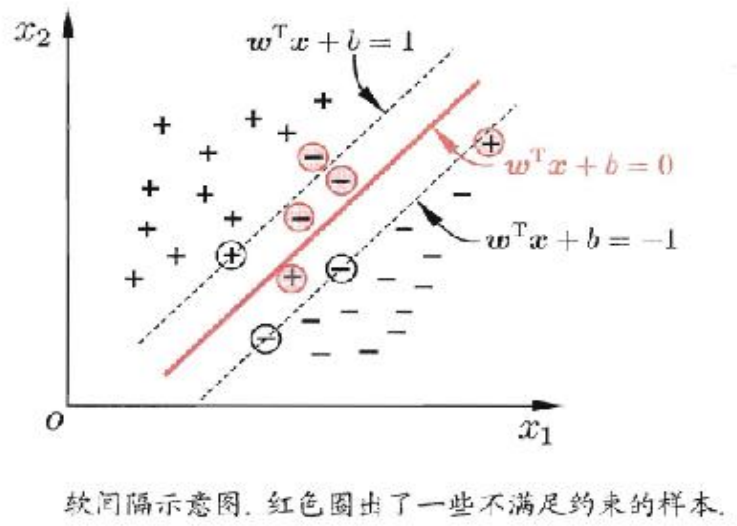
为了方便求解，我们在这里根据拉格朗日乘子法将约束条件融合到主式中，构成拉格朗日函数。

$$L(w,b)=\min_{\omega,b}\max_{\lambda}\frac{1}{2}\|w\|^2+\sum_{i=0}^{N}\lambda^{(i)}\left[1-y^{(i)}\left(w^{\top}k^{(i)}+b\right)\right]$$

其中，λ 是所谓的拉格朗日乘子。它后面“[ ]”中的式子来自原式的约束条件。这就是构造拉格朗日函数的方法。我们姑且不求甚解，按这个“套路”来就行。

软间隔（01）

现在的问题是：现实生活中，大部分数据都**不是完全可划分的**。
会有少量不同类的数据混杂在一起（见右图）。



软间隔示意图，红色圈出了一些不满足约束的样本。

我们能做的，就是容忍一定量的数据被划分错。用公式来表示分错就是

$$y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)-1<0$$

那么，我们来建构一个表达式，做两件事：
1、使间隔最大
2、使被错误分类的数据点尽可能少

也就是：

$$\min_{\boldsymbol{w},b}\ \frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{m}\ell_{0/1}\left(y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)-1\right)$$

这里的C，是一个正数，用来表示对数据的重视程度：
C越大，表示对数据越重视，这些的数据不能被分错。它会迫使w变小，当C足够大时就会退化为“硬间隔”；
C越小，表示这些数据分错了也没太关系。（理论上讲，也可以为数据点分配不同的C）

这里的 $\ell_{0/1}(z)$ ，是一个标识器函数。如果这个数据点被划分错了，得到1，否则为0。前面的 $\sum_{i=1}^{m}$ 是在累计分错个数。

软间隔（02）

道理是这么个道理。但是在数学处理上还有一个问题：这里的 $\ell_{0/1}(z)$ 阶是跃函数（图像呈阶梯状），它是不可导的。基于求导的各种运算就玩不起来了。

怎么办？通常我们选择如下三个函数之一进行**近似**表示：

$$\text{hinge 损失: } \ell_{hinge}(z) = \max(0, 1-z) \ ;$$

$$\text{指数损失(exponential loss): } \ell_{exp}(z) = \exp(-z) \ ;$$

$$\text{对率损失(logistic loss): } \ell_{log}(z) = \log(1+\exp(-z)) \ .$$

假设我们选择了hinge损失作为替代函数，那么我们得到

$$\min_{\boldsymbol{w},b} \ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \max\left(0, 1-y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)\right)$$

我们令 $\xi_i = 1-y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)$ 且 $\xi_i \geqslant 0$ ，那么上式又可以写为：

$$\min_{\boldsymbol{w},b,\xi_i} \ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \xi_i$$

* 这里的 ξ_i 就叫“**松驰变量**”。

软间隔（03）

将约束条件代入前面的表达式，得到SVM在“软间隔”条件下的目标函数：

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\xi_i + \sum_{i=1}^{m}\alpha_i\left(1 - \xi_i - y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right)\right) - \sum_{i=1}^{m}\mu_i\xi_i$$

* 这里的 α_i 和 μ_i 都是拉格朗日乘子。

另外，再啰嗦一句：**“松弛变量”的本质是数据点到软间隔边界线的距离，**所以它**一定是大于或等于0的**。

对偶（04）

这个拉格朗日函数看起来好凶呀！不好求，怎么办？我们试着去找它的“对偶式”吧！
那么，什么是对偶式呢？如何找到它的对偶式呢？

（1）什么是对偶式？

数学中的对偶，表达的是“在两个表达式中存在的某一种对应关系”。在SVM上，我们可以简单理解对偶式为低配版的原式。它相对好求解，并且解集与原式接近甚至完全一样。对偶式又分为两种：

a. 弱对偶性（weak duality）

$$min\ max\ L(w,b,\lambda) >= max\ min\ L(w,b,\lambda)$$

即：不附加任何约束条件的情况下，“先取最大值再从最大值集中取最小值”所得值，总是等于或大于“先取最小值再从最小值集中取最大值”所得值。也就是说，通过对偶式可以得到原问题的下界解。

b. 强对偶性（strong duality）

如果我们为对偶式赋加一定的约束（Slater条件 + KKT条件），便可得到与原式一样的解。

对偶（05）

在SVM算法中，我们选择了“强对偶性”。

Slater条件：保证对偶式必然有一个或多个解，但不一定是最优值。
KKT条件：保证了对偶式的解必定是最优解。

那么，什么是KKT条件呢？如右图所示：

$$(1)\ \lambda_i \left(1 - y_i \left(w^T x_i + b\right)\right) = 0$$
$$(2)\ \lambda_i \geqslant 0$$
$$(3)\ 1 - y_i \left(w^T x_i + b\right) \leq 0$$

- 条件（1）被称为互补松弛条件。
- 条件（2）、条件（3）是拉格朗日函数成立的要求。

“互补松弛条件”是什么意思呢？

- 当lamdba >= 0 时，那么1 - y(Wx+b)可以为0，也可以不为0。即：允许有分类不正确的情况存在。（即“软间隔”）
- 当lamdmab != 0时，那么1- y(Wx+b)必须为0，即所有的分类都必须正确。

KKT条件中，式（2）、（3）其一成立为“软间隔”，两者都成立则为“硬间隔”。

对偶（06）

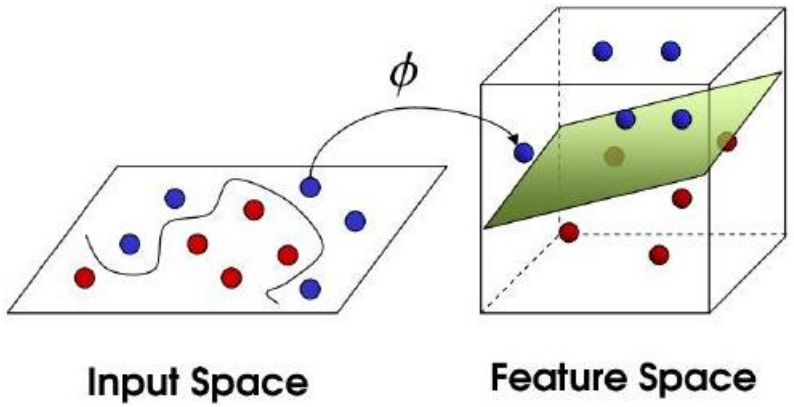
那么，如何求得对偶式？

Step 01 - 计算原式除“拉格朗日乘子lamda”之外的所有变量的偏导数。
Step 02 - 将得到的值代入原式，使得原式只剩下一个变量lamdba。
此式即为原式的对偶式。

优点：
原式要求三个变量：w、b、lamda，而在对偶式中只需要求解lamda的值即可，降低了计算复杂度。

核函数

在讲“核函数”之前，我们先来看一个神奇的现象：



在某些场景下，若我们将**线性不可分**的数据按照**一定的非线性规则**投射到**更高维空间**中，数据就有可能变的**线性可分**了。

核函数

但是，现在有两个很坑的事：
1，这种非线性投射规则很难找到
2，根据SVM的目标函数要求，我们还要计算两两数据（向量）之间的内积。
（为了达到数据线性可分的目的，有可能会将原数据投射到非常高的维度，这时再计算它们的内积时，计算复杂度很高了。）

怎么办？使用**核函数**！

核函数是一种**人为设计好的函数**，一般来讲它是半正定的。它的优点是：
将原数据代入到函数中，就可以直接得到它们通过对应规则在高维空间中的生成的新向量之间的内积值。这样我们就避开了（1）、（2）两个难点。

常用的核函数如下：

名称	表达式	参数
线性核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j$	
多项式核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j)^d$	$d \geqslant 1$ 为多项式的次数
高斯核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right)$	$\sigma > 0$ 为高斯核的带宽(width)
拉普拉斯核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\sigma}\right)$	$\sigma > 0$
Sigmoid 核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\beta \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j + \theta)$	tanh 为双曲正切函数, $\beta > 0, \theta < 0$