SVM及核函数

Support Vector Machines (SVMs) are a type of supervised learning algorithm that **can be used for classification or regression tasks**. The main idea behind SVMs is to find a **hyperplane** that **maximally separates the different classes in the training data**. This is done by finding the hyperplane that has the largest margin, which is defined as the distance between the hyperplane and the closest data points from each class. Once the hyperplane is determined, new data can be classified by determining on which side of the hyperplane it falls. SVMs are particularly useful when the data has many features, and/or when there is a clear margin of separation in the data.[2]

总之，SVM支持向量机是一种二分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器。通过其他的技巧，例如核函数的引入使得支持向量机本质上变为一个可以处理非线性可分问题的分类器，支持向量机包括硬间隔线性支持向量机、软间隔线性支持向量机和非线性支持向量机。

线性可分支持向量机与硬间隔最大化

线性可分支持向量机

线性可分支持向量机是一种分类模型，它的学习目的是找出线性可分的样本在特征空间中的最大间隔超平面。简单来说，它用于求解线性可分问题的分类问题。

首先，假定在某个特征空间上的训练数据集如下

$$T = \{(X_1, Y_1), (X_2, Y_2), (X_3, Y_3)..., (X_N, Y_N)\}$$

其中，$x_i \in \chi = \mathbf{R}^n$，$y_i \in \gamma = \{+1, -1\}, i = 1, 2, 3, ..., N$，$x_i$为第$i$个特征向量，也成为实例，$y_i$为$x_i$的类标记，当$y_i$为+1时称$x_i$为正例，反之，$x_i$为负例。

(x_i, y_i)称为样本点。

在数据集线性可分的条件下，存在着无数个超平面可以将两类数据正确地分开，而感知机利用误分类最小的策略，可以得到无限个超平面用于解决分类问题，而线性可分支持向量机利用间隔最大化求最优分离超平面，便可得到唯一的解。

线性可分支持向量机的定义

在给定线性可分训练数据集，通过间隔最大化或等价地求解相应的凸二次优化问题学习得到的分离超平面为

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

对应的分类决策函数为

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

称为线性可分支持向量机

支持向量机中的两类间隔——函数间隔和几何间隔

函数间隔

几何间隔是函数间隔定义的衍生，一般来说，一个样本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$w \cdot x + b = 0$确定的情况下，$|w \cdot x + b = 0|$可以表示样本点x距离分类超平面的远近。

而在$w \cdot x + b$的符合和类标记y的符号同号的情况下，表示分类正确，反之分类错误，所以能够使用$y(w \cdot x + b)$表示分类的正确性以及确信度。

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)的函数间隔为

$$\hat{\gamma_i} = y_i(w \cdot x_i + b)$$

定义T中所有样本点到超平面的函数间隔的最小值为$\hat{\gamma}$

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

几何间隔

观察函数距离的表达式，$\hat{\gamma_i} = y_i(w \cdot x_i + b)$可以看出，仅仅使用函数距离来选择超平面的话是不够的，因为成比例的缩放样本的时候，虽然超平面$w \cdot x_i + b$保持不变，但是函数间隔会随着比例的变化而变化，也就是说，函数间隔并不是实际意义上样本点到超平面的距离，如果我们对分离超平面的法向量w进行约束，将函数间隔固定，这时函数间隔就变为了几何间隔。

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

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

$||w||$表示的为法向量的模长，这样无论w和b如何成比例的变化，最终除以法向量的模长后，样本点到分离超平面的距离都是固定的

倘若T中所有样本点到超平面的函数间隔的最小值为γ

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

通过简单的运算，可以发现

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

同样的，对于每个样本点

$$\gamma_i = \frac{\hat{\gamma}_i}{||w||} \ (i = 1, ..., N)$$

从式子$\gamma_i = \frac{y_i(w \cdot x_i + b)}{||w||}$中可以看出，几何间隔实际上是样本点到分类超平面的带符号距离，当样本点被分类超平面正确分类的时候，就是实例点(样本点)到超平面的距离，若样本但被误分类的时候，此时表示误分类点到超平面的距离且符号为"-"，该式子巧妙地统一了“距离”与“分类的正确情况”于一式。

如果$||w|| = 1$，则表示，函数间隔和几何间隔是相等的。

总的来说：几何间隔就是函数间隔的规范化，函数间隔要么等于几何间隔，要么是几何间隔的$||w||$倍，这里的$||w||$表示的是超平面法向量的模长，也叫L_2范数

间隔最大化

对于线性可分的数据集而言，线性可分的超平面有无穷多个（试想一下感知机的分类过程），而如何确定最优的且唯一的分离超平面呢?

回顾刚刚的两类间隔，我们可以发现，在完全正确分类的情况下，距离分离超平面最远的点表示最有可能分对，而距离分离超平面最近的点则表示没有很足够的确信度将其分对。

所以，间隔最大化的意思实际上就是，对于十分靠近分离超平面的点，有足够大的确信度将最难分的这些点分开，从而找到一个唯一的，最优的超平面。

在这样的情况下，即使对于新实例，也能够表现出很好的分类预测能力。

最大间隔分离超平面

根据上面的讨论，我们的目标是找到一个最优的分离超平面，而“最优”这个程度副词如何去量化呢，没错，就是使用“几何间隔最大”这个工具，从而能够找到分离超平面。

这个问题可以使用数学语言进行转述，即求解下列的约束最优化问题：

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

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

也就是说，在给定的线性可分的数据集T下，在约束条件下为了找到最大的几何间隔γ，式子中的$s.t$的含义是$subject \ to$ 受....的约束

由于函数间隔对这一最优化条件的不等式没有影响（成倍的放大和缩小下函数间隔也成倍的放大和缩小，不影响优化问题的解），故式子等价于

$$\max_{w,b} \frac{\hat{\gamma}}{||w||}$$

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

取$\hat{\gamma} = 1$，带入上面的式子中，得到线性可分的支持向量机的最优化问题

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

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

注：$max \frac{1}{||w||}$和$min \frac{1}{2}{||w||}^2$是等价的

这是一个凸二次规划问题，如果求出了上述的问题的解w^*, b^*，那么就可以得到最大间隔分离超平面$w^* \cdot x + b^* = 0$以及分类决策函数$f(x) = sign(w^* \cdot x + b^*)$，即线性可分支持向量机模型

凸优化问题

凸函数

凸函数是指在其定义域内，任意区间的中点处的函数值不超过该区间端点处函数值的算术平均数的连续函数，更一般地说，一个函数$f(x)$在区间$[a,b]$上是凸的，如果对于任何两个点x_1和x_2在

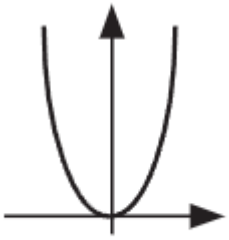
$[a,b]$ 和任何λ，当$0<\lambda<1$的时候，都有[3]

$$f\left[\lambda x_1+(1-\lambda) x_2\right] \leq \lambda f\left(x_1\right)+(1-\lambda) f\left(x_2\right)$$

这里的λ实际上表示的是函数段上的任意点，λ表示开头则$1-\lambda$表示结尾，组合起来就能够表示函数段上的任意一点

同时可以证明，在区间内，凸函数的二阶导数是大于等于0的

所以，在凸函数中，如果能够找到一个局部最优点，即表示在满足凸性的函数上该点为全局最优，如果找不到那就在函数端点处取值



convex function

对于一元函数，如果它在定义域内具有二阶导数，那么可以通过检查其**二阶导数是否大于等于0**来判断它是否为凸函数。对于多元函数，可以使用海森矩阵来判断。如果一个多元函数在定义域内具有二阶偏导数且其海森矩阵半正定，则该函数为凸函数。

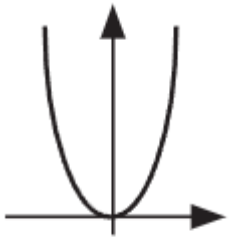
凸优化

凸优化问题是指约束最优化问题，在式子中，目标函数$f(w)$和约束函数$g_i(w)$都是R^n上连续可微的凸函数，约束函数$h_i(w)$是R^n上的仿射函数。

$$\min_w f(w)$$

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

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



convex
function

凸二次优化问题

当目标函数$f(w)$是二次函数且约束函数$g_i(w)$是仿射函数的时候，上述凸最优化问题成为凸二次规划问题

这里可以看出，一开始为什么要使用函数间隔和几何间隔去表示支持向量机的分类情况，$\hat{\gamma}_i = y_i(w\cdot x_i + b)$，如上所述，可以避免求解此问题时使用绝对值表示距离而造成的不可求导问题，这也许是支持向量机巧妙的地方之一

线性可分支持向量机学习算法——最大间隔法

由上所述，该算法的过程如下

对于输入是线性可分的训练数据集$T = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$，其中，$x_i \in \chi = R^n$，$y_i \in \gamma = -1, +1$，$i = 1, 2, 3, ..., N$

输出：最大间隔分离超平面和分类决策函数

- 构造并求解约束最优化问题
- 得到分离超平面

支持向量和间隔边界

在线性可分的情况下，训练数据集的样本点中与分离超平面距离最近的样本点实例称为支持向量，支持向量是指使约束条件等号成立的点

对于$y_i = +1$的点，支持向量在超平面H_1

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

对于$y_i = -1$的点，支持向量在超平面H_2

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

H_1, H_2是两条平行的线，之间形成一条长带，长带的宽度，也就是H_1, H_2的距离叫做间隔，长带之间没有任何的实例点，长带依赖于法向量w，即$|d_{H_1-H2}| = \frac{2}{||w||}$，只有支持向量对超平面起支持作用，其余的点不影响超平面的选择，也就是说，支持向量机其实是由很少的“重要的”训练样本所确定的

学习的对偶算法

上面对于支持向量机的凸二次优化问题进行了讨论，我们大概知道支持向量机最终求解的目标是什么，那么如何求解支持向量机的参数呢？对于原始的最优化问题，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，这样做的优点一是为了使得对偶问题更容易求解，二是自然地引入核函数，使得支持向量机能够处理非线性可分的数据

Working with the constraints can be cumbersome and challenging to manipulate, and it would be ideal if we could somehow turn this constrained optimization problem into an unconstrained one. One idea is to re-express the optimization problem into $\min_x L(x)$. [4]

也就是说，对于原约束问题我们可能很难求解，所以我们尝试将其转换为一个非约束问题

回顾一下原问题的公式

$$\min_x f(x)$$

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

$$h_i(x) = 0, \ \ i = 1, 2, ..., l$$

当约束条件$g_i(w) \leq 0, \ \ i = 1, 2, ..., k$和$h_i(w) = 0, \ \ i = 1, 2, ..., l$成立的时候，对$w$取$f(w)$的最小值$\min_w f(w)$，也就是说，对于满足约束条件的情况下，$\min_w f(w) \leq f(w)$，**即最大值是$f(w)$本身**，而不满足约束条件的情况下，式子的解可能是无穷大，所以我们把约束最优化问题转换为非约束最优化问题

设非约束最优化问题为

$$\min_x L(x)$$

$$L(x) = \begin{cases} f(x) \ \ g_i(x) \leq 0, \ \ i = 1, 2, ..., k \ \ and \ \ h_i(x) = 0, \ \ i = 1, 2, ..., l \\ \infty \ \ 其余情况 \end{cases}$$

注意：$\min_x f(x)$与$\min_x L(x)$是等价的!

虽然我们成功的把约束最优化问题变为非约束最优化问题（也就是我们无需考虑约束不成立的部分），但是我们如何解决这个问题呢？答案是，对这个原始问题$\min_x L(x)$，构造拉格朗日对偶式，于是我们得到了下面的式子

$$L(x) = \max_{\alpha, \beta} L(x, \alpha, \beta) = \max_{\alpha, \beta} [f_0(x) + \sum_{i=1}^{k} \alpha_i f_i(x) + \sum_{j=1}^{l} \beta_j(x)] \ \ s.t \ \ \alpha_i \geq 0$$

$$\min_x L(x) = \min_x \max_{\alpha, \beta} L(x, \alpha, \beta) \ \ s.t \ \ \alpha_i \geq 0$$

将$\alpha_i \geq 0$吸入，得

$$\min_x L(x) = \min_x \max_{\alpha, \beta; \alpha_i \geq 0} L(x, \alpha, \beta)$$

This optimization problem above is otherwise known as the primal (not to be confused with the primal variables), and its optimal value is indeed equivalent to that of the original constrained optimization problem.[4]上面这个优化问题也就是等价于原始问题，他的最优值等价于原始约束最优化问题。

原始问题

$$\min_x L(x) = \min_x \max_{\alpha, \beta; \alpha_i \geq 0} L(x, \alpha, \beta)$$

原始问题的解

$$p* = \min_x \max_{\alpha, \beta; \alpha_i \geq 0} L(x, \alpha, \beta)$$

对偶问题

$$\max_{\alpha, \beta; \alpha_i \geq 0} L(x, \alpha, \beta)$$

对偶问题是在**对偶变量**上求极大值的问题，当对**原问题的解**求极小值的时候

$$d^* = \max_{\alpha,\beta;\alpha_i \geq 0} \min_x L(x,\alpha,\beta) = \max_{\alpha,\beta;\alpha_i \geq 0} L_D(\alpha,\beta)$$

弱对偶性

在线性规划中，原问题的最优解通常会大于等于对偶问题的最优解，这叫做弱对偶性。原始问题是最小化损失函数$L(x)$，而对偶问题是最大化对偶损失函数$L_D(\alpha,\beta)$

$$p^* = \min_x L(x) \geq d^* = \max_{\alpha,\beta;\alpha_i \geq 0} L_D(\alpha,\beta)$$

强对偶性

当满足KKT条件时，原始问题和对偶问题之间存在强对偶性。这意味着原始问题的最优解等于对偶问题的最优解。

> 证明部分参照《A Comprehensive Guide to Machine Learning》一书中CHAPTER 7. DUALITY, NEARESET NEIGHBORS, SPARSITY——Strong Duality and KKT Conditions部分

线性支持向量机与软间隔最大化

非线性支持向量机与核函数

文献来源：

[1] 《统计机器学习：第七章 支持向量机》——李航

[2] Introduction to Support Vector Machines (SVM) - GeeksforGeeks

[3] Convex Function -- from Wolfram MathWorld

[4] 《A Comprehensive Guide to Machine Learning》—— Soroush Nasiriany, Garrett Thomas, William Wei Wang, Alex Yang Department of Electrical Engineering and Computer Sciences University of California, Berkeley February 9, 2018

[5]

[6]