一 SVM综述

在[机器学习](https://zh.wikipedia.org/wiki/%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0" \o "机器学习)中，支持向量机（英语：support vector machine，常简称为SVM）是在[分类](https://zh.wikipedia.org/wiki/%E5%88%86%E7%B1%BB%E9%97%AE%E9%A2%98" \o "分类问题)与[回归分析](https://zh.wikipedia.org/wiki/%E8%BF%B4%E6%AD%B8%E5%88%86%E6%9E%90" \o "回归分析)中分析数据的[监督式学习](https://zh.wikipedia.org/wiki/%E7%9B%A3%E7%9D%A3%E5%BC%8F%E5%AD%B8%E7%BF%92" \o "监督式学习)模型与相关的学习[算法](https://zh.wikipedia.org/wiki/%E7%AE%97%E6%B3%95" \o "算法)。给定一组训练实例，每个训练实例被标记为属于两个类别中的一个或另一个，SVM训练算法建立一个将新的实例分配给两个类别之一的模型，使其成为非概率[二元](https://zh.wikipedia.org/w/index.php?title=%E4%BA%8C%E5%85%83%E5%88%86%E7%B1%BB%E5%99%A8&action=edit&redlink=1)[线性分类器](https://zh.wikipedia.org/wiki/%E7%BA%BF%E6%80%A7%E5%88%86%E7%B1%BB%E5%99%A8" \o "线性分类器)。SVM模型是将实例表示为空间中的点，这样映射就使得单独类别的实例被尽可能宽的明显的间隔分开。然后，将新的实例映射到同一空间，并基于它们落在间隔的哪一侧来预测所属类别。

二 SVM算法

支持向量机由简至繁的包含：线性可分支持向量机，线性支持向量机和非线性支持向量机。简单模型是复杂模型的基础，也是复杂模型的特殊情况。当训练数据集线性可分时，通过硬间隔最大化(Hard Margin Maximization)，学习得到线性分类器，又叫硬间隔支持向量机；当数据近似线性可分时候，通过软间隔最大化(Soft Margin Maximization)，也可学习线性分类器，又称软间隔支持向量机；当训练数据线性不可分时，通过核技巧(Kernel Trick)及软间隔最大化，学习非线性支持向量机。

1.1 线性可分支持向量机

我们首先给出一个非常非常简单的分类问题（线性可分），我们先来看什么叫做线性可分？比如图1，直观来看我们可以用一条直线，将图中黑色的点和白色的点分开，并且分开之后没有任何误差，这时候这个数据集就是线性可分。我们可以给线性可分下一个更严谨的定义：线性可分是说可以用一个线性函数把两类样本分开，比如二维空间中的直线、三维空间中的平面以及高维空间中的超平面。所谓可分指可以没有误差地分开；线性不可分指有部分样本用线性分类面划分时会产生分类误差的情况。

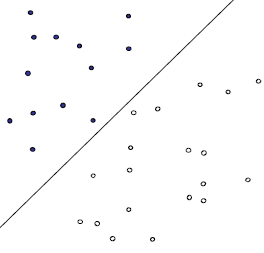


图1 示意图

二维三维我们可以直观的画图来看出，那么高维数据怎么去判断它是不是线性可分呢？答案是检查凸包（convex hull）是否相交。那么什么是凸包呢？简单说凸包就是一个凸的闭合曲线（曲面），而且它刚好包住了所有的数据。图2中蓝色线就是一个恰好包住所有数据的闭合凸曲线。（凸曲线的定义，寻找凸包的算法以及判断是否相交见附录）

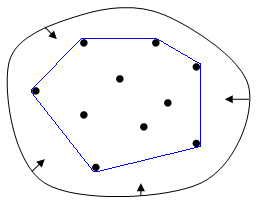


图2 凸包

很显然，图1的这条直线就是我们要求的划分数据集的直线之一（可以有无数条这样的直线）。而线性可分支持向量机利用间隔最大化来求最优分离超平面，此时结果是唯一的。

假如说，我们令黑色的点的labe为-1，白色的点label为+1，直线f(x) = w.x + b，这儿的x、w是向量，当向量x的维度=2的时候，f(x) 表示二维空间中的一条直线， 当x的维度=3的时候，f(x) 表示3维空间中的一个平面，当x的维度=n > 3的时候，表示n维空间中的n-1维超平面。当有一个新的点x需要预测属于哪个分类的时候，我们用sgn(f(x))，就可以预测了，sgn表示符号函数，当f(x) > 0的时候，sgn(f(x)) = +1, 当f(x) < 0的时候sgn(f(x))=–1。

此时可以给出线性可分支持向量机的严格定义：

**定义1**：给定线性可分数据集，通过间隔最大化或等价的求解相应的凸二次规划问题学习得到分离超平面为



以及相应的分类决策函数



称为线性可分支持向量机。

但是，我们怎样才能取得一个最优的划分直线f(x)呢？下图的直线表示几条可能的f(x)

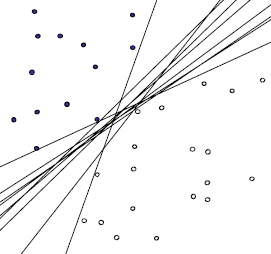


图3 不同的划分直线

划分曲线怎么划分最好呢？从直观上来说，就是分割的间隙越大越好，把两个类别的点分得越开越好。在SVM中，称为Maximum Marginal，是SVM的一个理论基础之一。

1.2 函数间隔和几何间隔

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，比如一个点的label为+，它距离分离平面较远，那么就比较确信预测是正确的，如果离得比较近，此时就不太确信。在分离超平面确定的情况下，表示点到分离平面的距离，而它的符号与类标记的符号是否一致能够表示分类的正确性和确信度，这就是函数间隔(Functional Margin)。给出函数间隔的严格定义：

**定义2（函数间隔）** 对于给定的训练数据集T和超平面(w,b)，定义超平面的关于样本点的函数间隔为



定义超平面(w,b)关于训练数据集T的函数间隔为超平面(w,b)关于T中所有样本点的函数间隔的最小值



函数间隔可以表示分类预测的正确性和确信度，但是只有函数间隔是不够的，只要成比例的改变w和b，超平面并没有改变，但是函数间隔却变了。所以可以对分离超平面做一些约束，例如规范化，使得间隔是确定的，这时函数间隔就成了几何间隔(geometric margin)

**定义3（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,y)的几何间隔为



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中所有样本点的几何间隔的最小值，即



显而易见，函数间隔和几何间隔有以下关系：



1.3 间隔最大化

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进行分类。就是说不仅要将正负实例点分开，而且对最难分的点（离超平面最近的点）也要以足够打的确信度将其分开。

**算法1（线性可分支持向量机学习算法——最大间隔法）**

输入：线性可分数据集，其中，；

输出：最大间隔分离超平面和分类决策函数。

1. 构造并求解约束最优化问题：



求得最优解。

1. 由此得到分离超平面：



分类决策函数



1.4 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与分离超平面距离最近的样本点的实例称为支持向量(support vector)。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。移动支持向量将改变所求的解，而移动其他的点则对结果没有影响。由于支持向量在确定分离超平面中起决定作用，所以将这种分类模型称为支持向量机。支持向量一般个数很少，所以支持向量机由很少的“重要的”训练样本决定。

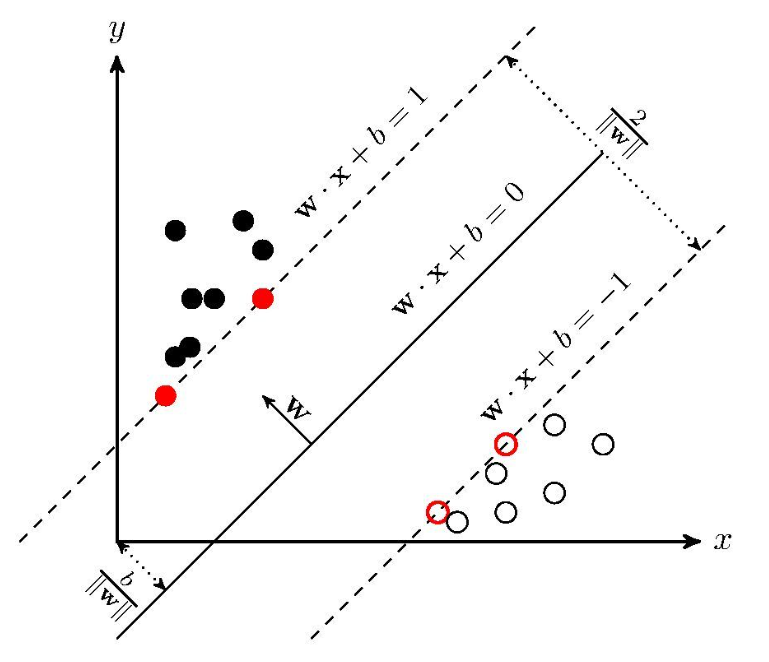


图4 支持向量

1.5 拉格朗日乘数法

在约束最优化问题中，常常利用拉格朗日对偶性(Lagrange duality)将原始的问题转化为对偶问题。对于线性可分支持向量机来说，这样做的优点：一是对偶问题往往更容易求解；二是自然引入核函数，进而推广到非线性分类问题中。

拉格朗数乘法用来求解有约束条件的极值问题。它的基本思想就是通过引入拉格朗日乘子来将含有n个变量和k个约束条件的约束优化问题转化为含有（n+k）个变量的无约束优化问题。拉格朗日乘子背后的数学意义是其为约束方程梯度线性组合中每个向量的系数。

我们以一个例子引出拉格朗日乘数法。

1.5.1 实例

求双曲线xy=3上离远点最近的点。首先，我们根据问题的描述来提炼出问题对应的数学模型，即：



两点之间的欧氏距离应该还要进行开方，但是这并不影响最终的结果，所以进行了简化，去掉了平方。

根据上式我们可以知道这是一个典型的约束优化问题，解这个问题时最简单的解法就是通过约束条件将其中的一个变量用另外一个变量进行替换，然后代入优化的函数就可以求出极值。我们在这里为了引出拉格朗日乘数法，所以我们采用拉格朗日乘数法的思想进行求解。

首先将x2+y2=c的曲线族画出来，如下图所示，当曲线族中的圆与xy=3曲线进行相切时，切点到原点的距离最短。也就是说，当f(x,y)=c的等高线和双曲线g(x,y)相切时，我们可以得到上述优化问题的一个极值（注意：如果不进一步计算，在这里我们并不知道是极大值还是极小值）。

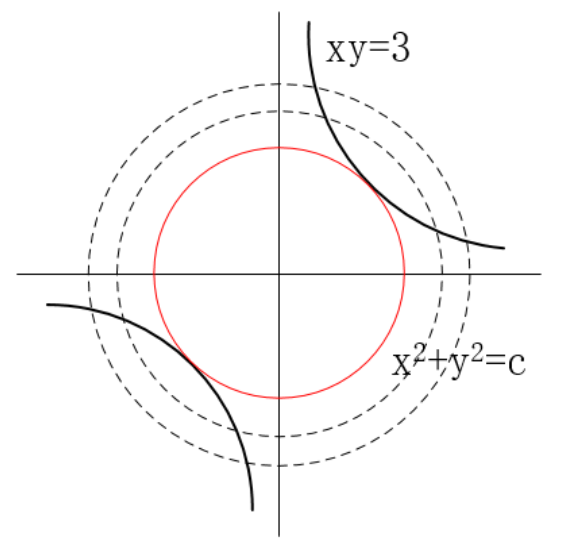


图5 约束曲线与问题曲线

现在原问题可以转化为求当f(x,y)和g(x,y)相切时，x,y的值是多少？如果两个曲线相切，那么它们的切线相同，即法向量是相互平行的，▽f//▽g。由▽f//▽g可以得到，▽f=λ\*▽g。这时，我们将原有的约束优化问题转化为了一种对偶的无约束的优化问题，如下所示：

原问题：min f(x,y)=x2+y2               对偶问题：由▽f=λ\*▽g得，

　　　　　　s.t. xy=3                                       fx=λ\*gx，

　　                                                           fy=λ\*gy，

                                                                 xy=3.

                约束优化问题                              无约束方程组问题

通过求解右边的方程组我们可以获取原问题的解，即

　　2x=λ\*y

　　2y=λ\*x

　　xy=3

通过求解上式可得，λ=2或者是-2；当λ=2时，(x,y)=(sqrt(3), sqrt(3))或者(-sqrt(3), -sqrt(3))，而当λ=-2时，无解。所以原问题的解为(x,y)=(sqrt(3), sqrt(3))或者(-sqrt(3), -sqrt(3))。

通过举上述这个简单的例子就是为了体会拉格朗日乘数法的思想，即通过引入拉格朗日乘子(λ)将原来的约束优化问题转化为无约束的方程组问题。

1.5.2拉格朗日乘数法的基本形态

求函数在满足下的条件极值，可以转化为函数的无条件极值问题。

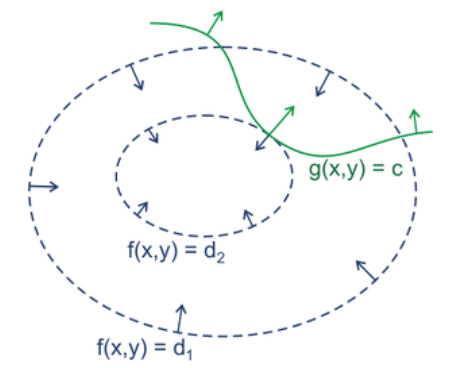


图6 有约束的极值问题

绿线标出的是约束g(x,y)=c的点的轨迹。蓝线是f(x,y)的等高线。箭头表示斜率，和等高线的法线平行。从图上可以直观地看到在最优解处，f和g的斜率平行。▽[f(x,y)+λ(g(x,y)−1)]=0, λ≠0。

一旦求出λ的值，将其套入下式，易求在无约束极值和极值所对应的点。F(x,y)=f(x,y)+λ(g(x,y)−c)，新方程F(x,y)在达到极值时与f(x,y)相等，因为F(x,y)达到极值时g(x,y)−c总等于零。上述式子取得极小值时其导数为0，即▽f(x)+▽∑λigi(x)=0，也就是说f(x)和g(x)的梯度共线。

1.6 拉格朗日对偶性

1.6.1 原始问题

假设是定义在上的连续可微函数。考虑最优化问题：



称这个约束最优化问题为原始最优化问题或原始问题。

引入广义拉格朗日函数(generalized Lagrange function)



这里，，是拉格朗日乘子，，考虑x的函数：



这里，下标P表示原始问题。

假设给定某个x，如果x违反原始问题的约束条件，即存在某个i使得或者存在某个j使得，那么就有



因为若某个i使约束，则可令，若某个j使，则可令使得，而将其余的均取0。

相反地，如果x满足原始问题的约束条件，则可知，因此，



x满足原始问题约束。所以如果考虑极小化问题



它是与原始最优化问题等价的，他们有相同的解。该问题称为广义拉格朗日函数的极小极大问题，这样一来，就把原始的最优化问题表示为广义拉格朗日函数的极小极大问题，为了方便，定义原始问题的最优值



称为原始问题的值。

1.6.2 对偶问题

定义



再考虑极大化，即



问题称为广义拉格朗日函数的极大极小问题。

可以将广义拉格朗日函数的极大极小问题表示为约束最优化问题：



称为原始问题的对偶问题，定义对偶问题的最优值



称为对偶问题的值。

1.7 学习的对偶算法

首先构建拉格朗日函数，为此，对

3 最优间隔分类器

4 核函数

5 SMO算法

6 其他衍生算法

三 应用实例

1数据集简介

2 数据处理

3 结果调优

重点问题

1就是在svm分类中的是怎么计算样本属于哪一类的概率的（为啥只有线性核的时候可以用距离来算概率，非线性核就不行）。

2对偶问题，

附录：

凸闭曲线(convex closed curves)：若一平面曲线总是位于它的每一点切线的同一侧，则此曲线称为凸曲线

quickhull算法来找到数据的凸包

sweepline算法判断凸包边缘是否有相交