# 支持向量机

SVM的一般流程：

1. 收集数据：可使用任意方法；
2. 准备数据：需为数值型数据；
3. 分析数据：有助于可视化分隔超平面；
4. 训练算法：SVM的大部分时间都源自训练，该过程主要实现两个参数的调优
5. 测试算法：十分简单的计算过程就可以实现；
6. 使用算法：几乎所有的分类问题都可以使用SVM，值得一提的是，SVM本身是一个二分类器，对多分类问题应用SVM需要对代码进行一些修改。

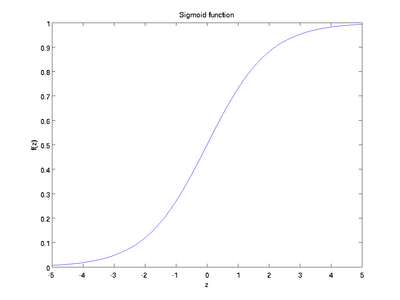
## 1. 逻辑回归

逻辑回归是将特征的线性组合作为输入，通过逻辑函数g(z)学习得到0或1的分类结果。换句话说，由于输入的取值范围从负无穷到正无穷，而输出为0或1，自然会联想到使用逻辑函数来映射。而所求假设函数h就是输出y=1时的后验概率。

假设函数：



其中，x是n维特征向量，g为sigmoid函数，其对应图像为：

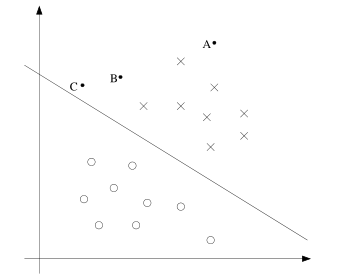


假设函数的结果为y=1时的后验概率：



故当需要判断新的输入x属于哪一类时，只需要计算此时的，若其值大于0.5，则属于类“1”；若小于0.5，则属于类“0”。事实上，的值只与有关，即当时，有；当时，有。若，则，可确信y=1；反之，则，可确信y=0。

逻辑回归最终的目标就是通过学习参数，使得正列（线性组合）远大于0，而负例（线性组合）远小于0。故可引出一个**函数间隔（Functional Margins）**的概念，如下图所示：



中间的直线由确定，其被称为“**分离超平面（separating hyperplane）**”，x表示正列，o表示负例。图中有三个被标注的点A,B,C，从图中我们可以非常确定A属于正例，B比较确定，而C不太确定。实际上，我们总是更加关注靠近分离超平面的点，总是希望它们能够远离分离超平面，而不是所有的样本点，后面会引出**几何间隔（Geometric Margins）**的概念。这正是逻辑回归与支持向量机的不同之处，逻辑回归考虑全局最优（使得特征的线性组合远远大于0，也就是说尽量让更多的样本点远离超平面），而支持向量机考虑局部最优（不考虑已经远离超平面的点）。

## 2. 函数间隔与几何间隔

为方便讨论支持向量机，在此重新定义分类的符号标注。

二元分类问题，类别标签;

若，则；反之，则；

参数为和，即有，代替了之前逻辑回归中的，。

### 2.1 函数间隔

给定一个训练样本集，函数间隔定义为：



当时，只有当的值为很大的正值时，函数间隔的值才会很大；当时，只有当的值为很小的负值时，函数间隔的值才会很大。即使得很大，可理解为函数间隔代表了认为样本点属于正例或负例的确信度。

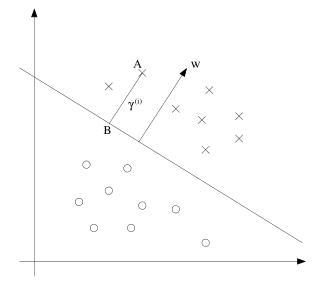
当w和b改变时，如分别增大到原来的2倍（2w,2b），虽然函数间隔的值会增大2倍，却对于假设函数h的值没有影响，因为我们要求的是，同等倍数增加w和b的值并没有影响。为了限制w和b的值唯一，需要加入归一化条件。

针对整个样本集，**函数间隔**取值为：



其为训练样本上分类正列和负例确信度最小的那个函数间隔。

### 2.2 几何间隔



w为分离超平面的法向量，A点为正列，，其到决策边界的距离为，w向量与向量BA的方向相同，w对应单位向量为，故B点可表示为：



又B点在决策边界上，故其满足，即有：



由上式，可的的值为：



结合负例的情况，可将式子写为：



由上式可知，当时，几何间隔实际上就是函数间隔，即归一化后的函数间隔就是几何间隔。同时增大w和b的倍数，值不变，即对最终判断分类的结果没有影响。

针对整个样本集，**几何间隔**取值为：



## 3. 最优间隔分类器（The optimal margin classifier）

基于上述内容，可知只需找到一个决策边界，使其在将正负样本分开的同时，其对应平面到距离最近的正负样本的距离（几何间隔）最大，即可很好地进行二元分类。实际上即为解决如下问题：

**形式1：**



||w||=1，表示几何间隔等于函数间隔，确保了所有几何间隔至少是。但是，它位于球体或圆形的表面，是非凸优化约束，不能保证找到全局最小值。首先，我们可以结合函数间隔与几何间隔的关系来转化这个问题。



可得到优化后的表达式：

**形式2：**



此时，我们需要求解的仍然是几何间隔的最优值，只是不再受||w||=1的约束。然而，这个目标函数仍然是一个非凸优化问题，还需要进一步优化。回顾之前讨论过同时改变w和b的值，对于结果并没有影响，为了最后只能得到唯一的解，需要对函数间隔进行约束，我们增加一个缩放约束：令训练集的函数间隔为1。



即有，距离超平面最近的点到超平面的距离为，由于求解的最大值等价于求解的最小值，故可得到如下表达式：

**形式3：**



目前，我们就将问题转化为可以解决的形式，一个二次函数的优化问题，可以采用QP、二次程序或梯度下降法来求最优值。这样，我们就得到了最优间隔分类器的模型。

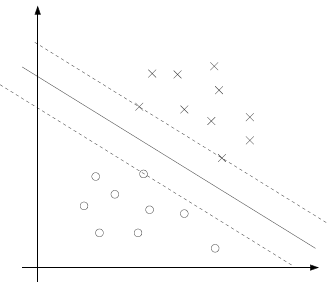
## 4. 对偶最优问题转化

将最优间隔分类器对应式子中的约束条件改写为如下形式：



从KKT（库恩-塔克条件，Karush-Kuhn-Tucker, KKT condition）条件得知函数间隔是1（离超平面最近的点）的线性约束式对应的系数（即对应约束式=0），对于其他不在线上的点（<0），极值不会在他们所在的范围内取得，故其对应的系数。（此处每一个约束式实际上就是一个训练样本）

如下图所示：



实线是最大间隔超平面，假设×号的是正例，圆圈的是负例。在虚线上的点就是函数间隔是1的点，他们对应的系数，其他点均有。这三个点即称为**支持向量**。构造拉格朗日函数如下：



注意到这里只有没有是因为原问题中没有等式约束，只有不等式约束。

结合构造的拉格朗日函数与原最优间隔分类器式子，可将其转化为如下问题：



先求解的最小值，对于固定的，的最小值只与w和b有关。对w和b分别求偏导。



故有：



将上式代入拉格朗日函数中，可得如下式子：



故最后可得：



由于最后一项为0，故其可化简为：



在此将向量内积表示为。

此时拉格朗日函数只包含了变量，且只有求得了才能得到w和b。

求取极大化的过程。



假设函数，且有：



从而引出了核函数的重要概念，对于支持向量机的优化方法必不可少。

同时，在求解模型的过程中会遇到有离群值的干扰，需要对模型做出修正，提出软间隔的概念。

## 5. 核函数

### 5.1 特征映射

问题的原始属性为，在输入学习算法之前需要对它们进行一些处理，从而得到输入特征，假设Φ为原始属性到输入特征的特征映射（feature mapping）：



### 5.2 核函数

**（1）基本定义**

将原始属性的内积换成输入特征的内积，即得到核函数（kernel）的定义：



只要得到，再计算它们的内积就可以很容易地得到核函数，即使是一个高维向量，需要花费很大的计算代价，但是核函数的有效计算可以让支持向量机学习高维空间的特征。

如下列所示。



则有：



最终的结果就是核函数的定义。

由，即可得特征映射为:



计算需要 ，而计算只需要。

**（2）一般形式**

考虑另外一个核函数的形式：



可得特征映射为:



参数c控制了与之间的相对权重。

更一般的核函数式子：



特征映射为(1-n^(d+1))/(1-n)维，计算需要，计算仍只需要，因此不需要明确地表示出在非常高维的特征空间中的特征向量。

**（3）高斯核**

下面考虑一个更为复杂的核函数，由于与是两个向量，如果当它们十分接近时，我们期望十分大；当它们相距甚远时，我们期望很小。换言之，即通过来衡量与之间的相似度或x与z之间的相似度。核函数可以采用以下的形式：



当x与z十分接近时，的值为1；当x与z相距甚远时，的值为0。这个核函数叫做**高斯核（Gaussian kernel）**，它就是支持向量机所采用的核函数。

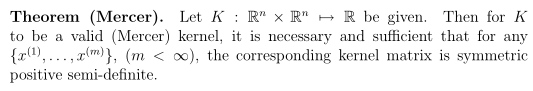
**（4）核函数的合法性**

如何判断存在特征映射Φ，核函数对于所有的x与z都成立，即**核函数的合法性？**

假设K是核函数，同时K也是的方阵，，其称为**核矩阵（Kernel matrix）。**如果核是有效的，K必须是对称矩阵，因为两个输入x,z的内积与两者的前后顺序无关。用表示的第k个值，于是有：



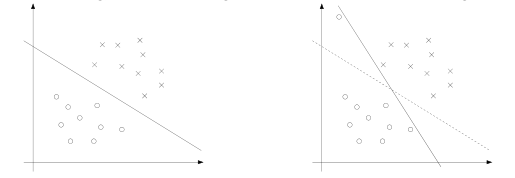
因此，K是半正定矩阵。也就是说，如果K是一个合法的核，那么它所对应的核矩阵是对称的半正定矩阵。事实上，这是一个充分必要条件，符合Mercer定理。



线性分类器对原始空间中并非线性可分离的数据进行分类，SVM输出非线性决策边界的整个过程，是一个求解凸优化问题的过程。许多其他算法可以写成内积的形式，就可以将内积换成核，将特征空间映射到无限维空间，以解决低维空间无法实现可分的情况。

## 6. L1 norm软边界SVM

支持向量机的算法有个前提条件是假设数据集是线性可分的，但是有时并不能保证所有情况下都线性可分或是得到满意的分离超平面。如下图所示，当存在一个较远的离群值（outliers）时，超平面出现了巨大的摆动，使得间隔（margin）较小。



为了满足非线性分类并减小离群值的影响，需要对最优间隔分类器算法进行修正（调整正则化的L1值）：



通过将函数间隔值修正为1-，增加目标函数的惩罚项c，其中c控制目标函数最小值和约束条件中的函数间隔最小值为1这两者之间的相对权重，常数c是一个参数（松弛变量），因此我们可以通过调节该参数得到不同的结果。

拉格朗日函数为：



其对偶形式为：



注意到这里改变的约束条件仅是的取值范围，同时满足KKT互补条件。



L1 norm软边界SVM可以处理非线性分隔情形，包含异常数据的情况时，选择不进行完全正确的分隔来解决这个问题。

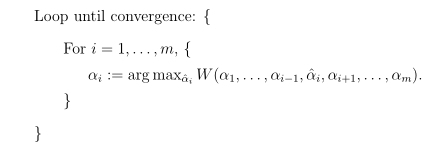
## 7. 序列最小优化算法（SMO）

### 7.1 坐标上升法（Coordinate ascent）

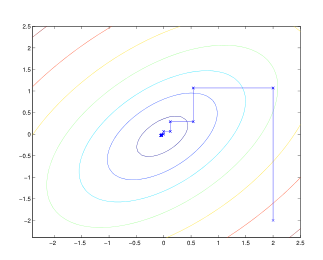
考虑解决无约束最优问题：



联想到之前已经学习过的优化解法，如梯度上升法和牛顿法，现在来介绍一种新的方法——坐标上升法。



固定以外的所有参数，求解函数的最优值，然后再依次分别固定其他参数，循环优化，使得它的增长最大，选择的一般顺序是，。以下是一个含有两个参数的坐标上升法的优化过程，依次沿着坐标平行的方向取最大值。



当固定其他参数而求解某个参数的最优值时，坐标上升法收敛速度要好于牛顿法。

### 7.2 SMO(sequential minimal optimization)

回归L1 norm软边界SVM模型的求解问题，实际上是求出满足两个约束条件，同时使目标函数最大时的值，如果采用上升坐标法来解决这个问题，固定的值，来求最大时的值，这是行不通的。



因为KKT条件约束的存在，当其他值确定时，的值也就固定了，它们之间是固定的相关关系，此时就无法来优化的值。改进的方法当然存在，只需要同时改变两个即可解决。基本步骤为：

* 使用**试探法（heuristic）**选择两个变量和用于更新值；
* 固定其他的值，反复地利用和两个值来优化W(α)；
* 重复以上步骤直到算法收敛，找到的全局最大值；
* 通过检验**收敛容忍参数**（convergence tolerance parameter）**tol**（0.01~0.001）是否满足KKT条件来判断是否收敛。

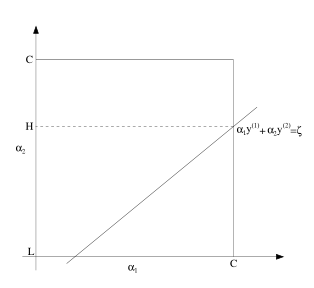
SMO算法之所以是个有效的算法，重点在于更新计算和的值十分快捷。

下面来推到出和的更新式子。由约束条件知：



可以转化为以下式子和示意图：





一定存在下限L和上限H使得满足[0,C]，同理存在（直线）使得满足[0,C]，使它落在直线上或L上。通过与之间的关系可以减少参数的个数：



更新参数后，W(α)可以表达为的二次函数：

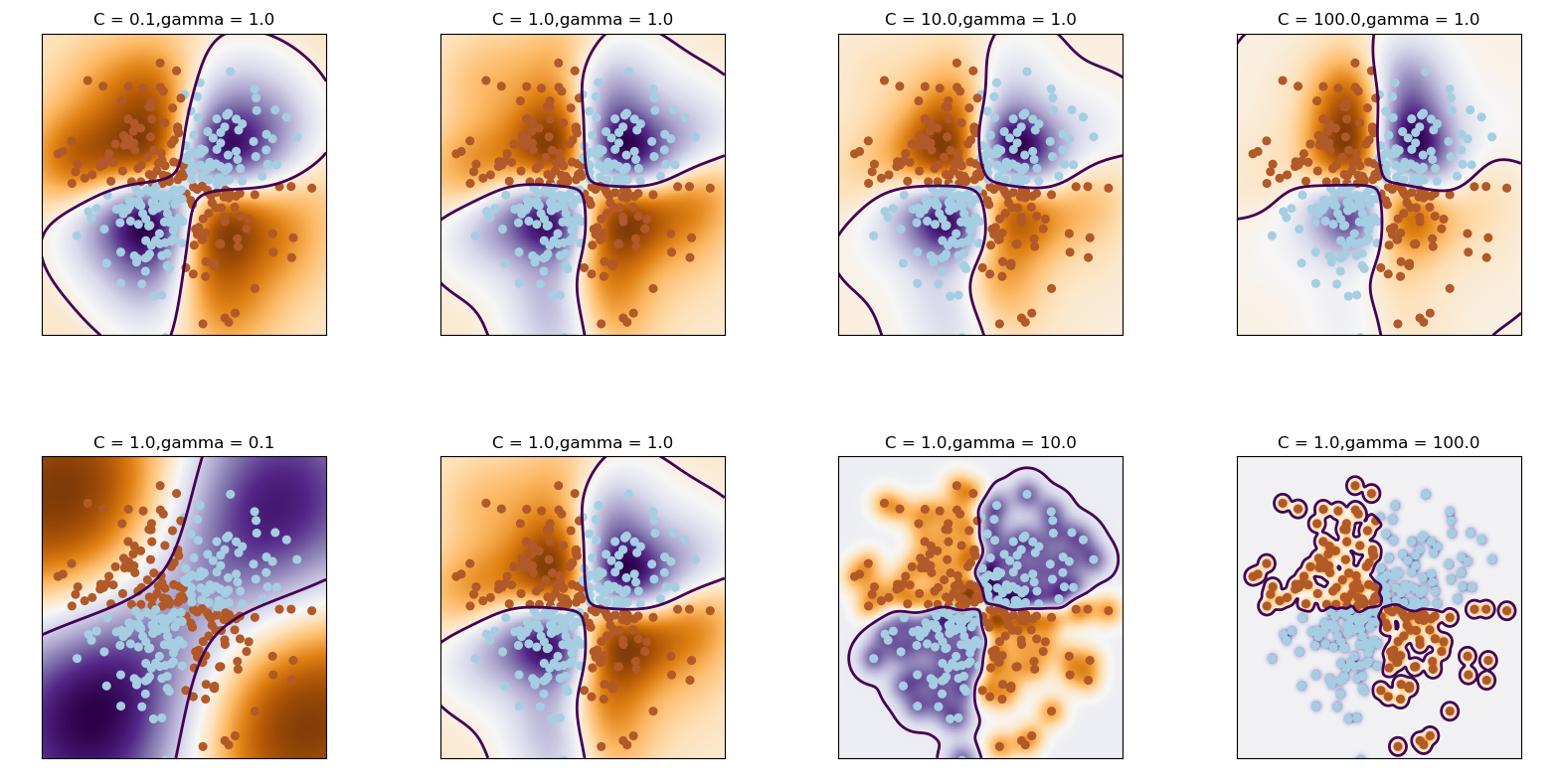




通过选择合适的值优化二次函数就可以得到W(α)的最大值。



（使用高斯核RBF时）参数的调节对分类效果的影响：



令，上面四个图是固定 γ，选取不同的C，可以发现C越大时，允许被错误分类的样本数量就越少；C越小时，允许被错误分类的样本数量就可能就会越多。

下面四个图是固定C，选取不同的 γ，会影响高斯核函数的分布， γ 越大，高斯核函数分布就会越陡峭； γ越小，高斯核函数分布就会越平缓； 因此当γ越大时，截取等高面时，样本就只会在自己周围形成分布；γ越小时，截取等高面时，样本就可以在自己周围较大的范围内形成分布，因此同类样本就有可能连接在一起。

SVM算法最初是为二分类问题设计，当处理多分类问题时，需要构造合适的多类分类器。

**1.one-versus-rest（一对多法）：**

训练时，依次把某个类别的样本归为一类，其他剩余的样本归为另一类，有k个类别的样本就构造出k个SVM；预测时，将未知样本分类为具有最大分类函数值的那类。

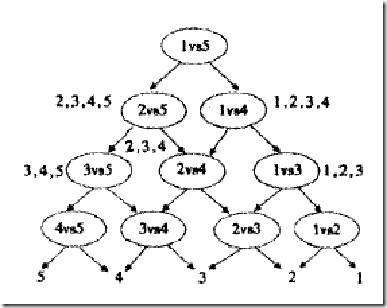
缺陷：会存在数据倾斜；分类结果出现重叠（属于多个分类器）或者不可分类（不属于任何一个分类器）。

**2.one-versus-one（一对一法）：**

训练时，选择一个类的样本作为正样本，负样本则只选择一个类，又k个类别的样本，就构造出 k(k−1)/2个分类器，虽然分类器的数组增加了，但是训练阶段所用的总时间却比"one-versus-rest"方法少很多。预测时，每个分类器都会预测出一个结果，然后统计最后的预测结果。尽管这个方法也有分类重叠现象，但是不会有不可分类的现象，因为不可能所有类别的票数都是0。

**3.DAG SVM：**

类似于"one-versus-one"方法，只是在对一个样本进行分类之前，先按照下图的结构组织分类器（这是一个有向无环图，因此被称作DAG SVM），



在预测时，可以先问分类器"1 vs 5"，如果回答是5，就往左走；再问分类器"2 vs 5"，如果还回答5，就继续往左走，一直问下去，就可以得到分类结果，如果有k个类别，那么只调用k-1个，分类速度快，且没有分类重叠和不可分类现象。

缺陷：如果一开始分类器回答错误，那么后面的分类器是无法纠正的。