# THUCTC

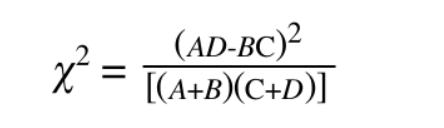
THUCTC(THU Chinese Text Classification)是由清华大学自然语言处理实验室推出的中文文本分类工具包，能够自动高效地实现用户自定义的文本分类语料的训练、评测、分类功能。采用Java语言编写

文本分类通常包括特征选取、特征降维、分类模型学习三个步骤。如何选取合适的文本特征并进行降维，是中文文本分类的挑战性问题。

在THUCTC中选取二字串bigram作为特征单元，特征降维方法为Chi-square，权重计算方法为tfidf，分类模型使用的是LibSVM或LibLinear。THUCTC对于开放领域的长文本具有良好的普适性，不依赖于任何中文分词工具的性能，具有准确率高、测试速度快的优点。

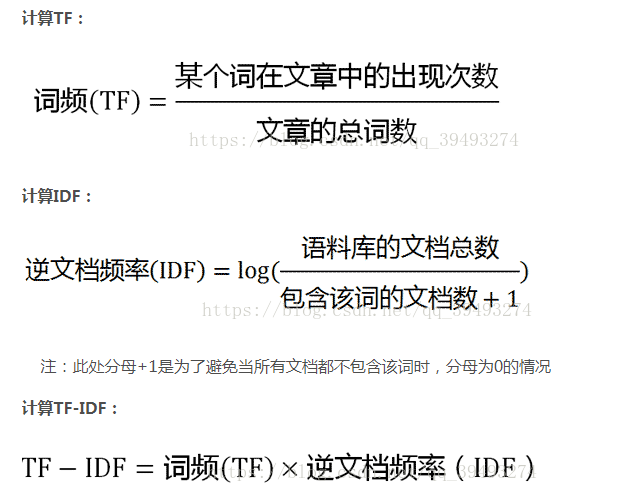
二字串bigram：分词模式中的一种。unigram 一元分词，把句子分成一个一个的汉字； bigram 二元分词，把句子从头到尾每两个字组成一个词语； trigram 三元分词，把句子从头到尾每三个字组成一个词语.

Chi-square：卡方检验-计算每个特征的卡方值，取最大卡方值的前K个特征。卡方检验在特征筛选的应用中，通常采用四格表法计算，即直接套用公式。



即使是多分类的数据也可以转换成四格表计算。其中A为某一特征词在某一篇文档中出现的次数，B、C、D需要计算得到。

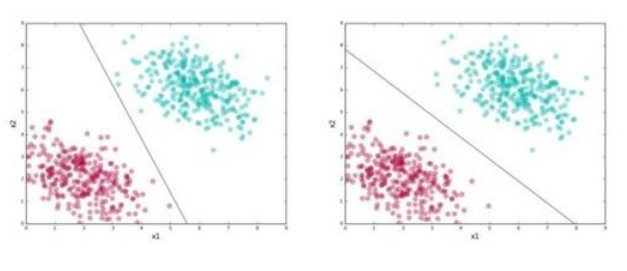
Tf-idf：根据文档计算每个特征的权重。该值为一个NxK的矩阵。N为文档数。



# Svm

<https://wenku.baidu.com/view/35ec70029ec3d5bbfc0a74ab.html>

目标：找到一个超平面，使得它能够尽可能多的将两类数据点正确的分开，同时使分开的两类数据点距离分类面最远。



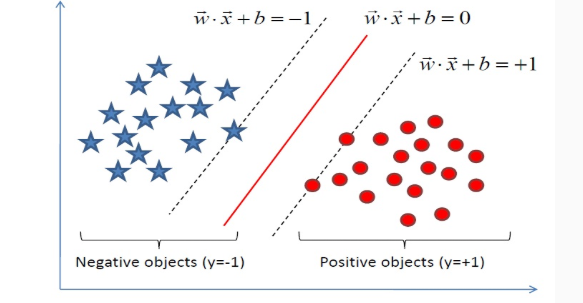
支持向量机可以分为线性、非线性两大类。其核心是线性可分，但是现实中往往大部分数据都是线性不可以分的。根据cover定理，复杂的线性不可分问题通过映射到高维空间可能是线性可分的，因此只要特征空间的维数足够高。所以大部分问题都是围绕线性可分探讨的。

什么是线性可分？就是可以用一条或几条直线把不同类别的样本点分开，在3维及以上特征空间，即就是求解超平面。

通常情况考虑的是一个两分类问题，数据点用 x 来表示，这是一个 n 维向量，而类别用 y 来表示，可以取 1 或者 -1 ，分别代表两个不同的类。一个线性分类器就是要在 n 维的数据空间中找到一个超平面，其方程可以表示为：



下面举个简单的例子，一个二维平面(一个超平面，在二维空间中的例子就是一条直线)，如下图所示，平面上有两种不同的点，分别用两种不同的颜色表示，一种为红颜色的点，另一种则为蓝颜色的点，红颜色的线表示一个可行的超平面。



从上图中我们可以看出，这条红颜色的线把红颜色的点和蓝颜色的点分开来了。而这条红颜色的线就是我们上面所说的超平面，也就是说，这个所谓的超平面的的确确便把这两种不同颜色的数据点分隔开来，在超平面一边的数据点所对应的 y 全是 -1 ，而在另一边全是 1 。

接着，我们可以令分类函数：



显然，如果 f(x)=0 ，那么 x 是位于超平面上的点。我们不妨要求对于所有满足 f(x)<0 的点，其对应的 y 等于 -1 ，而 f(x)>0 则对应 y=1 的数据点。

因此，我们在进行分类的时候，将数据点 x代入 f(x) 中，如果得到的结果小于 0 ，则赋予其类别 -1 ，如果大于 0 则赋予类别 1 。如果 f(x)=0，则很难办了，分到哪一类都不是。

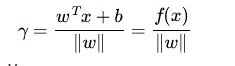
如何确定超平面：咱们就要确定上述分类函数f(x) = w.x + b（w.x表示w与x的内积）中的两个参数w和b，通俗理解的话w是法向量，b是截距。那如何确定w和b呢？答案是寻找两条边界端或极端划分直线中间的最大间隔（之所以要寻最大间隔是为了能更好的划分不同类的点）。也就是点到平面的距离。

在超平面w\*x+b=0确定的情况下，|w\*x+b|能够相对的表示点x到距离超平面的远近，而w\*x+b的符号与类标记y的符号是否一致表示分类是否正确，所以，可以用量y\*(w\*x+b)的正负性来判定或表示分类的正确性和确信度。于此，我们便引出了定义样本到分类间隔距离的函数间隔functional margin的概念。

我们定义函数间隔functional margin 为：



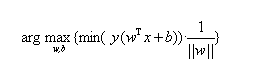
但是函数间隔存在一个问题就是当W、b成比例的改变时，其实超平面并没有改变，到那时函数间隔却变成了原来的成比例倍。所以我们需要对函数间隔规范化，也就是真正的点到超平面的距离即几何间隔。





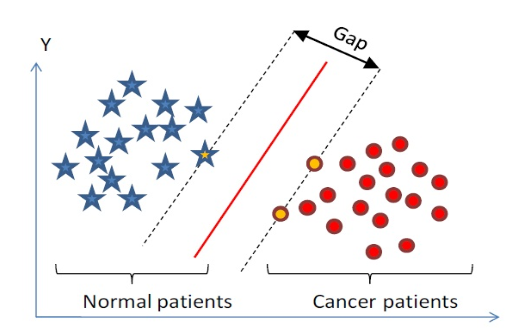
按照我们前面的分析，对一个数据点进行分类，当它的间隔越大的时候，分类的 置信度越大。对于一个包含 n 个点的数据集，我们可以很自然地定义它的间隔为所有这 n 个点的间隔值中最小的那个。于是，为了使得分类的置信度高，我们希望所选择的超平面能够最大化这个间隔值。因此最大化间隔分类的目标函数为：

 即



什么是支持向量机？

可以看到两个支撑着中间的 gap 的超平面，它们到中间的纯红线separating hyper plane 的距离相等，即我们所能得到的最大的 geometrical margin，而“支撑”这两个超平面的必定会有一些点，而这些“支撑”的点便叫做支持向量Support Vector。因而，该分类问题的样本点就只考虑这些支持向量。



由于使用了几何间隔，不管W、b成比例怎么变化，间隔都不会发生变化。因此，我们令所有支持向量机所在的超平面为：

 或

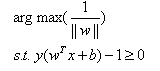


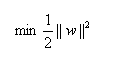
即就是：



为什么是等于1呢，就是方便求解方便，而不是所有的点都满足等于1，但是可以通过等比例的调节W、b的值使得支持向量机满足该条件，而并没有改变超平面方程。

因此求解的目标函数就可以简化为：



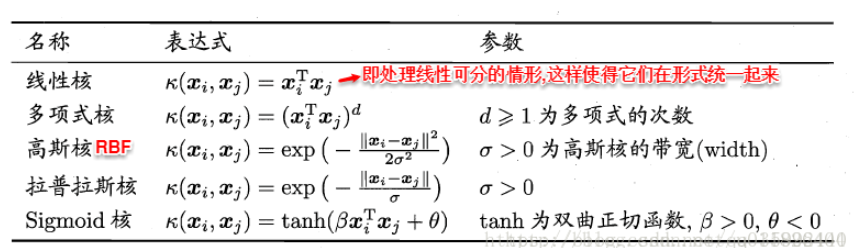
为了后面计算的方便，我们将目标函数等价替换为：

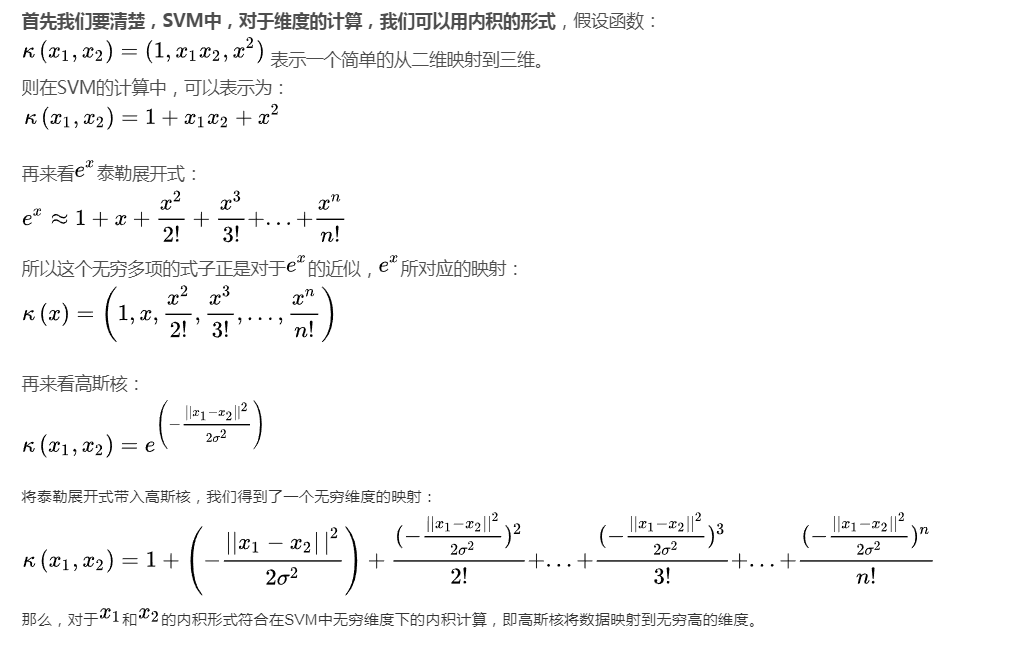


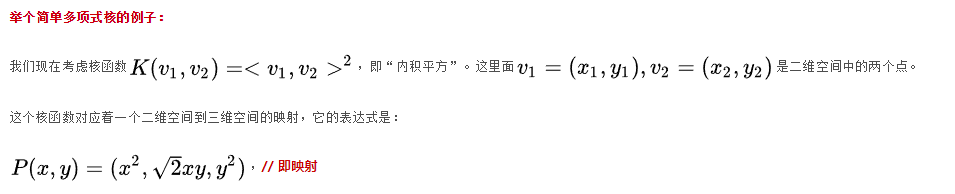
线性不可以分？

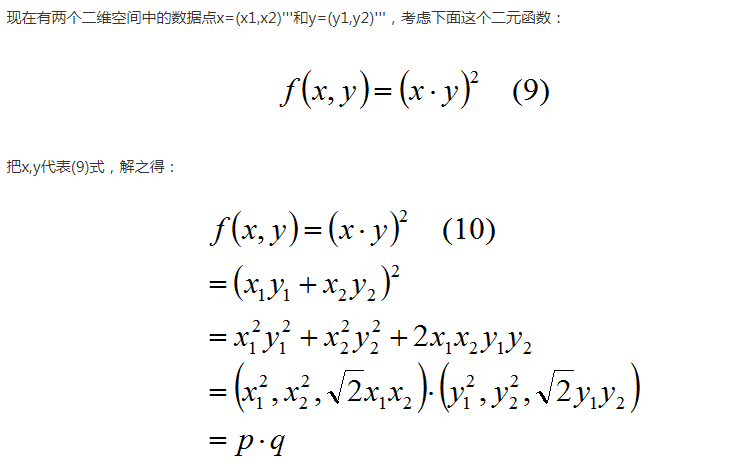
前面讲到过线性不可以分的情况下需要将样本映射到高维特征空间中，因此引入核函数的概念来实现这一功能。

简而言之：在线性不可分的情况下，支持向量机通过某种事先选择的非线性映射(核函数)将输入变量映射到一个高维特征空间，在这个空间中构造最优分类超平面。常用的核函数表：









那么怎样的函数才可以作为核函数呢？下面的一个定理可以帮助我们判断。

Mercer定理：任何半正定的函数都可以作为核函数。其中所谓半正定函数

是指拥有训练集数据集合，我们定义一个矩阵的元素 ，这个矩阵是nxn的矩阵，如果这个矩阵是半正定的，那么 就称为半正定函数。

值得注意的是，上述定理中所给出的条件是充分条件而非充要条件。因为有些非正定函数也可以作为核函数。

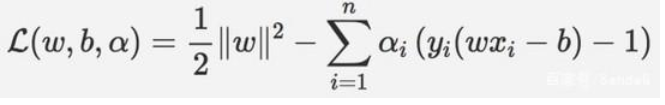
————————————————————————————————

目标函数的求解：

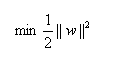
对于此类问题的优化求解，我们可以利用凸优化的凸二次规划来求解。本问题采用拉格朗日乘子法将其转换为对偶变量（dual variable） 优化问题。

关于拉格朗日和对偶问题可以看：<http://blog.sina.com.cn/s/blog_628cc2b70102xyed.html>

该问题的拉格朗日函数可表示为：



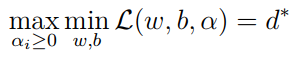
则问题就转换成：



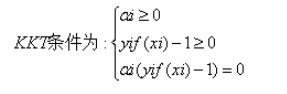
这里用p\*表示这个问题的最优值，且和最初的问题是等价的。

接下来，我们进行第二步：将不易求解的优化问题转化为易求解的优化：

我们看一下我们的新目标函数，先求最大值，再求最小值。这样的话，我们首先就要面对带有需要 求解的参数w和b的方程，而αi又是不等式约束，这个求解过程不好做。所以，我们需要使用拉格朗日函数对偶性，将最小和最大的位置交换一下，这样就变成了：

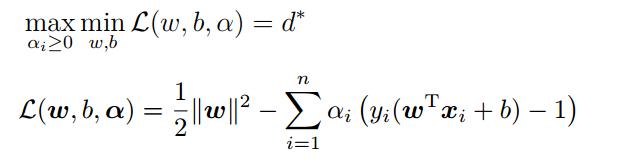


交换以后的新问题是原始问题的对偶问题，这个新问题的最优值用d\*来表示。而且d\*<=p\*。我们关心的是d=p的时候，这才是我们要的解。需要什么条件才能让d=p呢？那就是要满足KKT条件。

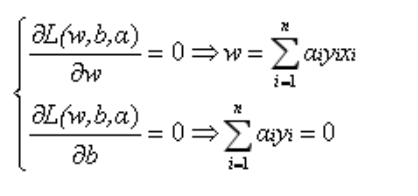


对偶问题求解？

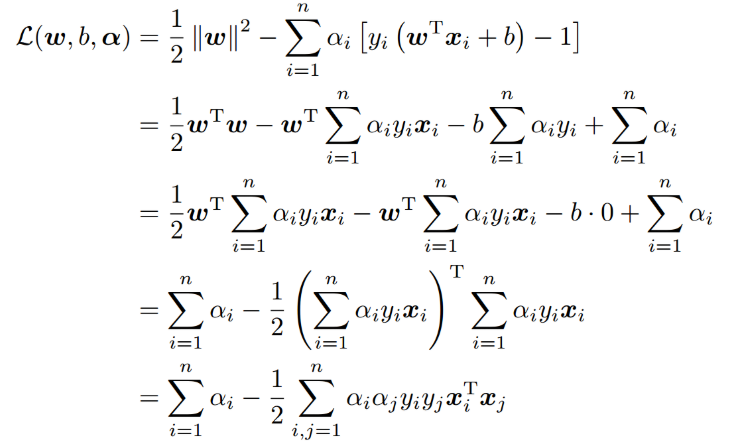
第一步：根据上述推导已知：



对拉格朗日式子分别求编导得到：



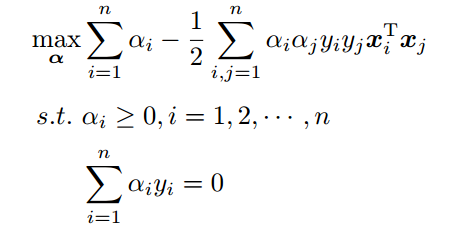
从而原问题化简得：



从上面的最后一个式子，我们可以看出，此时的L(w,b,α)函数只含有一个变量，即αi。

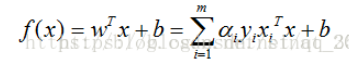
第二步：

现在内侧的最小值求解完成，我们求解外侧的最大值，从上面的式子得到：



现在我们的优化问题变成了如上的形式。对于这个问题，我们有更高效的优化算法，即序列最小优化（SMO）算法。我们通过这个优化算法能得到α，再根据α，我们就可以求解出w和b，进而求得我们最初的目的：找到超平面，即"决策平面"。<http://blog.sina.com.cn/s/blog_62970c250102xfok.html>

<https://wenku.baidu.com/view/9f6f99567dd184254b35eefdc8d376eeaeaa1780.html>

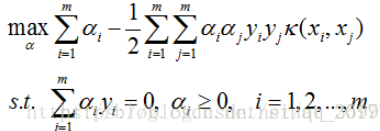


**我们可以将SMO算法过程归纳如下：**

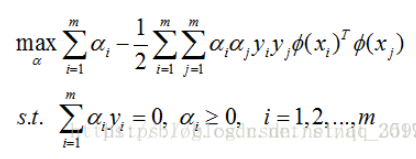
1. 初始化所有变量的解αi=0
2. 选取优化变量α1, α2，解析求解这两个变量的二次规划问题，得到最优解α1’, α2’
3. 验证，所有变量的解αi是否满足KKT条件，若不满足，继续step2，否则step4
4. 得到a=α1’, α2’…

非线性可分？

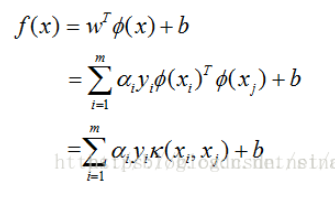
针对第二步目标表达式 ，其中 不在是样本集，而是通过核函数映射后的结果，即 。因而，上述表达式可表示为：



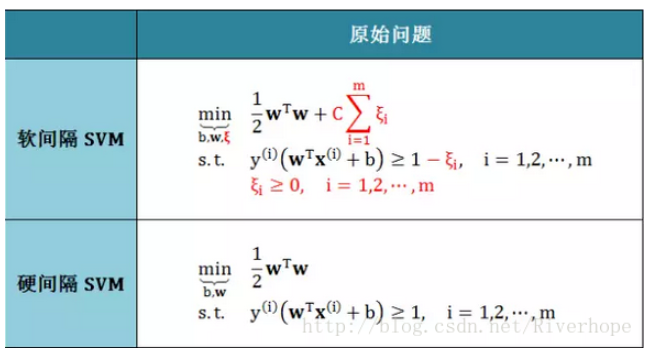




求解后：



其实，最后求解出来，会发现：不是支持向量的样本的拉格朗日乘子全为0.只有支持向量的样本系数不为0.



# Linear

# 朴素贝叶斯（NB）

# K-近邻

# 决策树

# Rocchio算法

# 神经网络

# 集成学习

# 迁移学习

# 强化学习