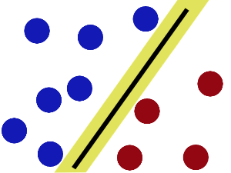
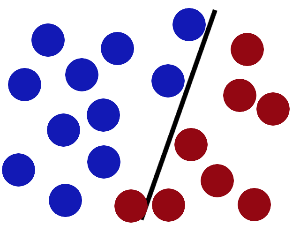
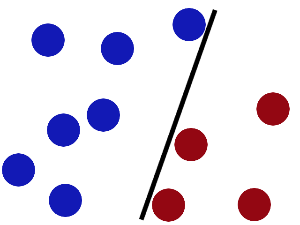
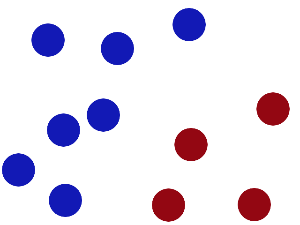
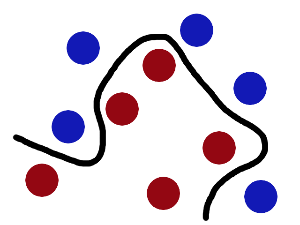
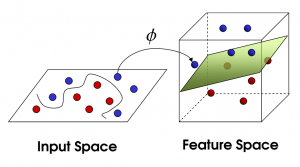
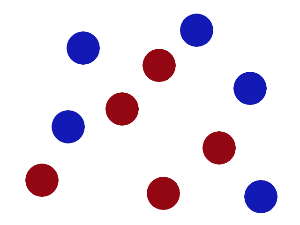
支持向量机（Support Vector Machine, SVM）

支持向量机可作为一种分类/回归方法，在2012年前一直具有不错的口碑。SVM依据输入数据集是否线性可分，可分为线性SVM和非线性SVM。下面有一个形象的“天使拍桌子”故事可以描述SVM可实现的效果及中间的关键技术/概念。

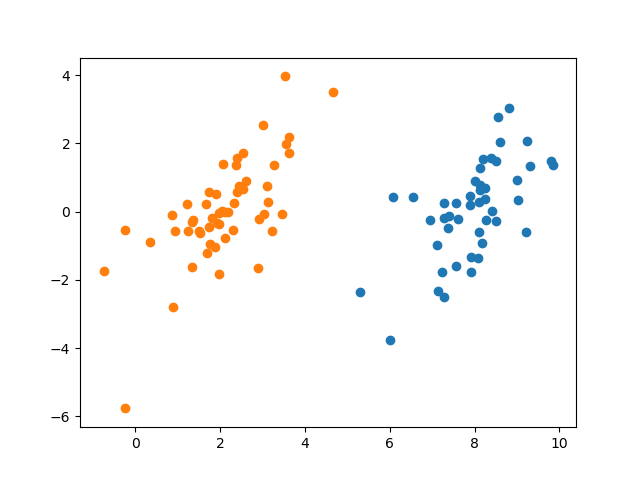




SVM如何解决多分类问题？

线性SVM原理及实践：

参考博客地址：https://cuijiahua.com/blog/2017/11/ml\_8\_svm\_1.html



（1）线性可分与线性不可分：如何判别？

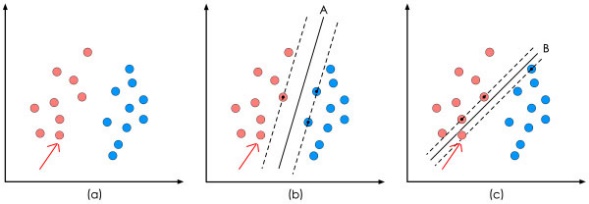
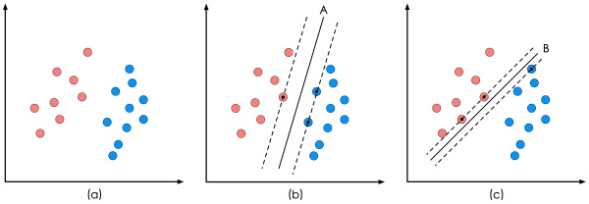
（2）如果线性可分，如何求解决策面方程/超平面方程？

围绕上述问题展开如下分析与建模过程。

最优决策面方程具有的特点：分类间隔最大。所谓分类间隔指决策面到两类点的最短距离。如下图所示，分类器B优于分类器C的原因在于分类间隔比分类器C大，好处在于如果添加箭头所指的红点之后，分类器B仍然能正确的分隔两类点，分类器C则不能。也就是说，分类间隔最大的决策面可以使得分类器容错性能最优。

两类点对于决策面的分类间隔存在竞争关系，即都希望决策面到自身所在类的分类间隔最大。那么从公平的角度，最优的决策面对每一类的分类间隔相等，即应该到两类点的最短距离相等。

综上所述，最优决策面方程应该满足的特点，到每类的分类间隔最大且相等。



建模过程：决策面方程与分类间隔、输入集与输出集、约束条件、优化问题描述、模型求解。

决策面方程与分类间隔：

决策面方程：，输入集合，其中N为样本数目，为每个样本的数据集，输出集合，为每个样本的标签。

最优化问题描述为





建模过程分两步，分类间隔最大，分类并归一化分类间隔

分类间隔可以表示为，问题目标函数为，其中表示支持向量的样本点。

分类依据如下：，当且仅当支持向量上的样本取等号。

不等式两边同除以，分类依据变为，注意与为同一决策面。所以将符号定义交换一下，变为仍成立，即

注意此时原目标函数变为，为了后续求导方便，将目标函数更改为。综合上述，优化问题为



问题求解，采用拉格朗日对偶+KKT条件的方法，将原优化问题转化为对偶问题求解，分两步：（1）将带不等式约束的优化问题变为无约束的构造优化问题。（2）将不易求解的优化问题变为易求解的优化问题。

（1）面对不等式约束的优化问题，拉格朗日对偶的方法思路是利用拉格朗日函数将约束条件放入目标函数中，构造新的优化问题，使得该目标函数在可行域内与原目标函数一致，在可行域外为无穷大。这样就可转化为无约束的优化问题。

拉格朗日函数为：，其中

其对偶函数为，可以发现当时，，当时，。原优化问题可以等价转化为如下

（2）面对上述问题，下一步将其转化为容易求解的问题，利用拉格朗日函数对偶性，可将其变成并假设该问题最优值为，原优化问题最优值为，由凸优化理论可知，，当且仅当原优化问题为凸优化问题且满足KKT条件时取等号。

KKT条件如下所示



由此推导可得







将其带入对偶问题中可得：



原对偶问题转化为：





上述优化问题的解法待定，目前采用SMO算法比较高效。

上述模型是基于数据百分之百可分的情况，而对于现实中的数据来说并不那么理想，这时我们可以通过引入松弛变量和惩罚参数，来允许有些数据点可以处于超平面的错误的一侧。

原优化问题变为







对应拉格朗日函数为：



对偶函数为

KKT条件为：

（1）

（2）

（3）

（4）

（5）

由（3）得知，满足。

令，称为样本点的函数间隔，由它可以判断样本点*i*与超平面（最优决策面）、间隔边界平面（支持向量的样本点所在平面）之间的关系。

下面我们由KKT条件来推导与函数间隔之间的关系。

当时，，同时由（3）和（5）可得，，，所以。

当时，由（3）和（5）可得，，，所以。

当时，由（3）和（5）可得，，，所以，样本点处于间隔边界平面上。

对于的情况，当时，样本点处于间隔边界平面与超平面之间，还能正确分类，当时，样本点处于超平面上。当时，样本点处于超平面错误分类的一侧。

所以由KKT条件可得（将（1）代入拉格朗日函数，并结合（3）消去C项），原优化问题转化为





由公式（2）可得，可假设，其中B为固定常数，为任意两个拉格朗日乘子。SMO算法思路：通过成对更新两个乘子。一旦找到了一对合适的拉格朗日乘子，那么增大其中一个同时减小另一个。这里的合适就是指两个乘子必须满足以下两个条件，条件是两个乘子要在间隔边界之外，而且第二个条件则是两个乘子还没有进行过区间化处理或者不在边界上。

选择两个乘子，满足，因为两个乘子不容易同时求解，可先通过求解第二个乘子的解，然后再用第二个乘子表示第一个。在两个步骤之间，应该需要确定第二个乘子的边界，以免求得的第二个乘子超出边界。

假设满足，下面综合以下两个条件来确定它的上下界

（1）

（2）

由式（2），可根据的取值情况可分为两种情况。当同号的时候，，结合式（1）可得，即，结合（1）可得，的上下界可表示为；当异号的时候，，结合式（1）可得，即，结合（1）可得，的上下界可表示为；因为在编程过程中，我们无法用B来更新新的拉格朗日乘子，只能通过旧的拉格朗日乘子（上一次拉格朗日乘子的值来更新）所以，结合（2）式，我们可以表示在编程中的上下界如下，



在的上下界确定之后，下面面临的问题是如何计算并根据它的值来计算。

由式（2）我们可以得到之间的关系，通过两边同时乘以，我们可以得到，即，其中。然后准备将其代入目标函数中，通过对目标函数求导可得的计算公式。



其中

对目标函数求导可得



上面可以发现，根据数据集和标签集以及上次的拉格朗日乘子可以求出，将其继续简化可得



其中，（误差函数），

（学习速度）

根据前面求出的上下界，可以得出

下面根据可以求出

下面根据两个拉格朗日乘子计算阈值b，因为它关系到了的计算，也就关系到了（误差）的计算。

当两个乘子在0和C之间的时候，根据KKT条件可知，这两个样本点是支持向量上的点，即满足如下公式



由上述可知，



所以可以得到



从而我们可以梳理出简化版SMO算法的步骤：初始化两个乘子以及相应的中间变量误差、学习速度等，随机选择两个乘子，按照公式分别更新两个乘子和b，收敛条件：最大迭代次数。

伪代码如下：

从上面我们可以发现每次选择两个乘子的算法需要进一步研究，而不是随机的选择，这会影响最后的误差与收敛速度。