支持向量机学习笔记

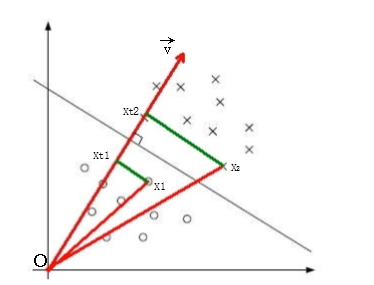
实验平台：Anaconda

理论书籍：《机器学习实战》

原理可以参考：[https://zhuanlan.zhihu.com/p/30487021。但是其推导过程中有一些错误。](https://zhuanlan.zhihu.com/p/30487021。但是其推导过程中国有一些错误。)下面的推导过程已将所有错误纠正了。

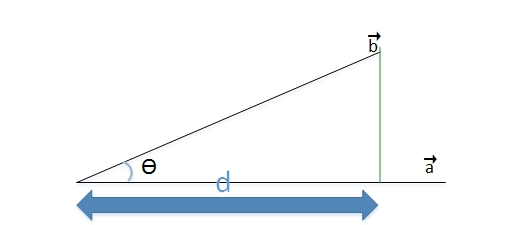
原理：

支持向量：离分隔超平面最近的那些点。需要最大化支持向量到分隔面的距离。



黑色的线是假设的分隔面，x1 x2为支持向量，先找到与分隔面垂直且穿过原点的向量V，将x1 x2分别投影到V上面成为xt1 xt2。原点到xt1的距离小于原点到xt2的距离，可以发现在决策界左边的点到投影到V上后到原点的距离是小于决策界右边的点到原点的距离的。

以下就用公式来表示这个特征：



设向量b在向量a的投影距离为d，两个向量的内积（数量积）表示为：绿色的线表示决策界，向量a表示决策界的法向量（即上面提到的向量v），向量b表示的是决策界上一点x与原点之间的向量。

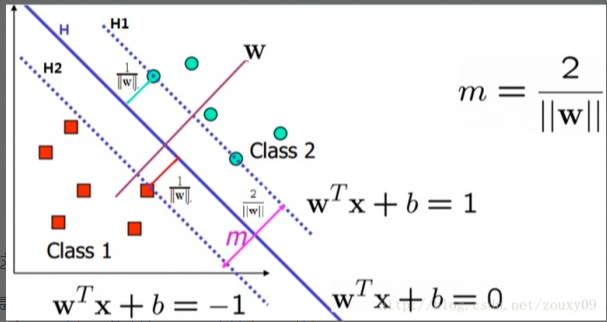
这个式子是为了计算余弦值



现将向量a看作是决策边界的法向量，当向量a是单位法向量的时候，其模为1，此时

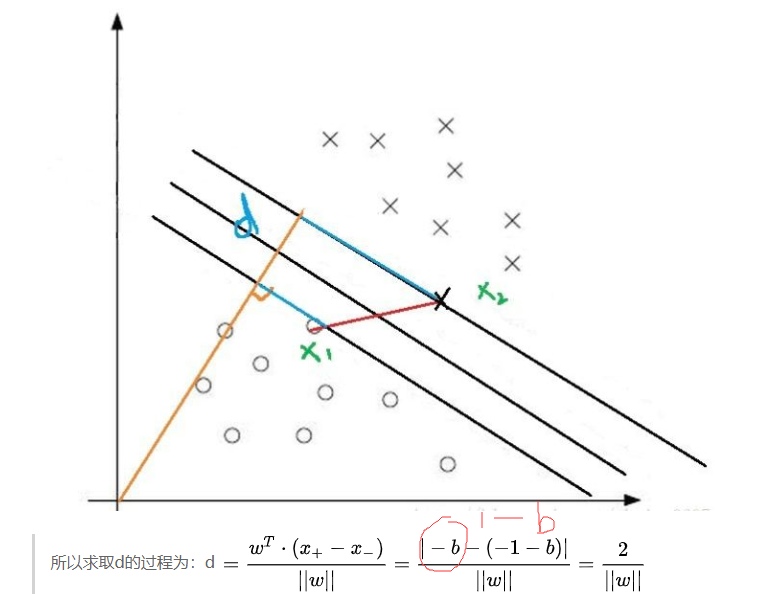
。映射到上面的实际情况就可以得到xt1 xt2（即d的长度大小）：

此时就能找到一个常数c来分来两类数据，使得：vx+c>0为一侧数据，vx+c<0为另外一侧数据。



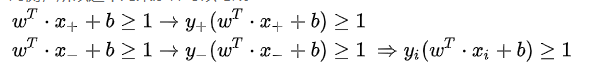
上图可以看出，我们将决策边界法向量相关的方程写为了wx+b=0，那么在决策界两遍的两类数据分别结果就是正负一。这一点就对应上面的vx+c的决策方程。

那么最大间隔就是m，可以通过以下来证明



图中的推导过程中-b应该是1-b，这幅图是直接对应上面那副图的。X+属于上面的数据集，应该采用wx+b=1的方程计算。

因为推导的过程就有假设条件是两个支持向量要求在两侧，所以这个约束条件可以写成：



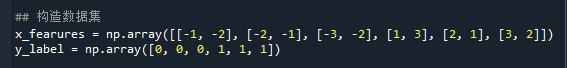
因为y表示的是label，y的取值为正负一。当x在决策界上方的时候，y=1;否则，y=-1。这样就可以保证最终的约束条件：y(wx+b)大于等于1。

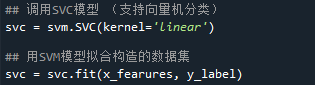
由于对乘积优化比较困难，因此一般采用固定其他值优化一个值的方法来做。如果令所有支持向量的label\*(wx+b)都为1，那么就可以通过求的最大值来得到最终解。但实际上只有那些离分隔超平面最近的点得到的值才为1。而离超平面越远的数据点，其值也就越大。

应用：

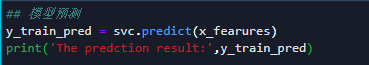


主要是从sklearn库里导入svm！

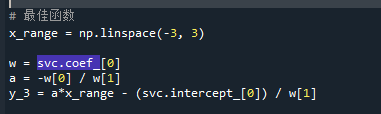




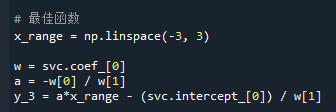
这两步是将坐标点和其特征值配对。



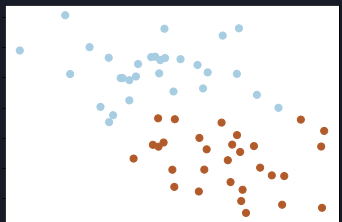
拟合完成以后就可以进行预测了！



以上代码是为了画出决策界。coef\_是用于提出特征的权重，intercept\_是为了提出决策函数中的常数。

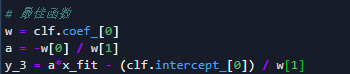


因为超平面的方程形式为：w1\*y+w0\*x+b=0，所以y=(-w0/w1)\*x-b/w1。Coef\_中存储的就是特征的x和y的权重，intercept中是截距。

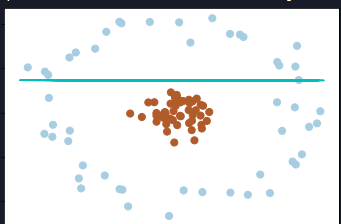


这种情况不容易找到最大间隔，于是可以利用软间隔进行约束，允许个别数据出现在间隔带中。在SVC中有一个参数C是惩罚参数，对分错的数据进行惩罚。C越小，说明对数据容忍性越大。

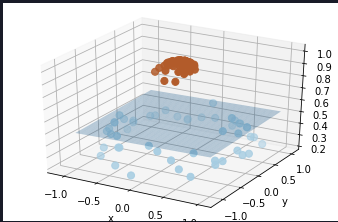




最佳函数写法不变。



如果遇到这种无法进行线性划分的数据，就可以将二维（低维）数据映射到三维（高维）空间去进行划分。





利用高斯核函数来实现分类。

