

机器学习第一次作业

——线性判别分析（LDA）

2052902 韩意

一、 问题描述

请用 Python 编程实现线性判别分析 LDA，并给出下面数据集上的结果及说明。

编号	属性 1	属性 2	类别
1	0.666	0.091	正例
2	0.243	0.267	正例
3	0.244	0.056	正例
4	0.342	0.098	正例
5	0.638	0.16	正例
6	0.656	0.197	正例
7	0.359	0.369	正例
8	0.592	0.041	正例
9	0.718	0.102	正例
10	0.697	0.46	反例
11	0.774	0.376	反例
12	0.633	0.263	反例
13	0.607	0.317	反例
14	0.555	0.214	反例
15	0.402	0.236	反例
16	0.481	0.149	反例
17	0.436	0.21	反例
18	0.557	0.216	反例

二、 算法实现

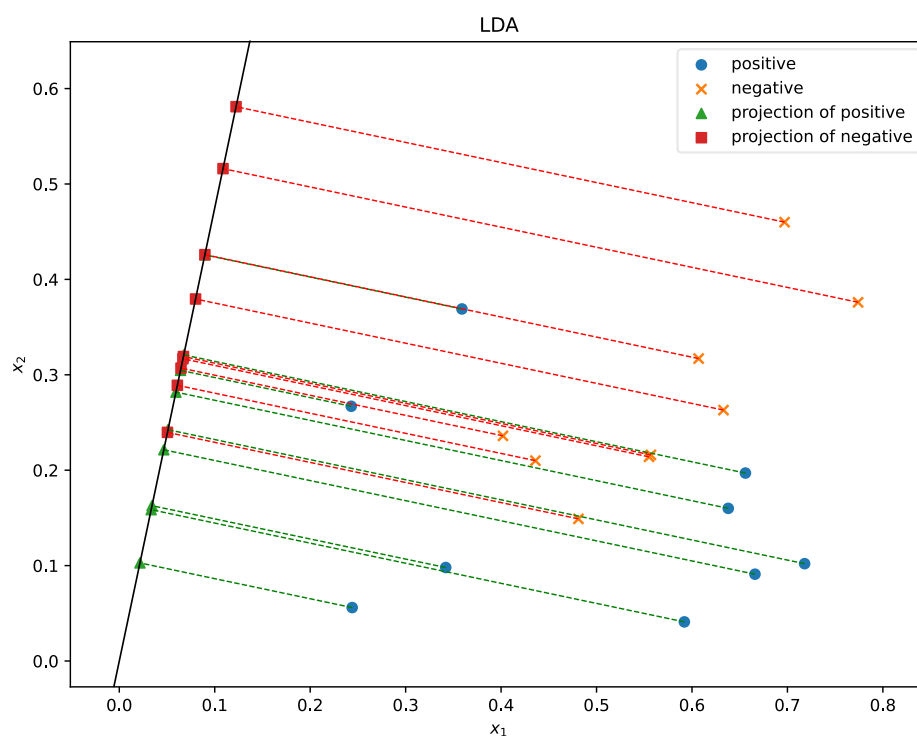
代码中将 LDA 模型实现为一个类，运用公式

$$\mathbf{w} = \mathbf{S}_w^{-1}(\mu_0 - \mu_1)$$

求得 LDA 对应的直线。代码中已将 \mathbf{w} 进行了归一化得到单位向量。

训练模型时，未对数据划分训练集和测试集，统一将 18 个点均用于训练，

得到的模型绘图如下：



上图有几个注意点：

- 1) 投影判别区域发生重合，说明 LDA 算法不是完美的，不能保证两类数据投影到直线上是两个互不相交的区域，因此可能存在错判。这也是线性模型的弊端。
- 2) 为了反映投影的垂直关系，横纵坐标的尺度应当是一致的（即单位长度要相同），在 Python 中使用语句 `plt.axis('equal')` 保证。

最终模型用于预测时，只要求数据 \mathbf{x} 的投影 $\mathbf{w}^T \mathbf{x}$ ，并与 $\mathbf{w}^T \boldsymbol{\mu}_0$ 和 $\mathbf{w}^T \boldsymbol{\mu}_1$ 进行比较，得到对应分类。