数据二分类实验

1 实验目标

给定一个可二分类的二维数据集，通过使用机器学习方法求出一个模型（即分类直线），具有良好的泛化能力，对测试集内的数据也进行正确分类。

2 实验结构

AI-homework 项目文件夹

——-+ main.py 总流程控制

——-+ data.py 数据集准备

——-+ data_visual.py 数据可视化

——-+ perceptron_algor.py 感知机算法

——-+ choose_model.py 选择最佳模型

——-+ svm_algor.py 支持向量机算法

3 人员分配

杨军典：实验设计。对实验的目标、结构、可行性等方面进行分析、制定，对 python 语言进行支持。

熊春艳：准备数据集。对实验所需的数据集构造、处理、标记，将结果返回给 main 文件。

周世年：感知机算法设计。设计训练集的感知机算法，并在测试集上测试，可视化并分析结果。

尚大伟：SVM 算法设计。采用 SVM 算法对原始数据进行分类测试，可视化并分析结果。

司远：辅助 ppt 制作。

4 实验流程

如图 1所示

5 数据集准备

输入：数据集个数

输出：标记过的可二分类的训练集和测试集 (数量比例 4:1)

设计数据集数据结构：$dataSets = [[], [], ..., []]$，整体是一个列表，其中每个数据元素点也是一个列表。

数据元素点列表结构为 1*3, 其中最后一个元素代表标记值。

设计算法:

1、随机产生 N 组 2 维数据点（代表平面上的一个点）, 限定所有数据点的 x 和 y 轴范围均在 [-100,100]。

即生成一个 $N * 2$ 的矩阵，其中每个元素的值范围都在 [-100, 100]。

2、设定标记模型：人为设定分类线为 $x - y = 0$，法向量为 $w = [1, -1]$,$b = 0$。

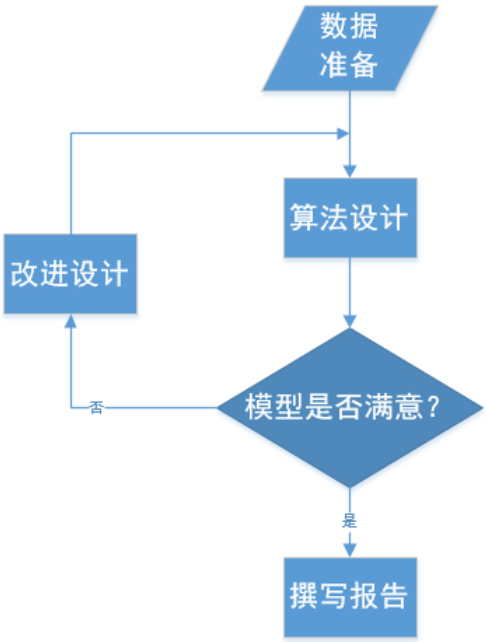

图 1: 实验流程

3、标记数据点：将第 1 步产生的数据点根据第 2 步的模型进行标记正负，正实例为 1, 负实例为-1。

4、将标记过的数据永久保存成磁盘上的 csv 文件。4、将在内存中标记过的数据集按照 4:1 的比例分割为训练集和测试集进行返回。

6 感知机算法

输入：训练集数据，K 折参数

输出：模型（分类直线参数和模型得分)

因为训练集的数量较少和为了提高泛化能力，将训练集按照 K 折交叉验证的方法进行模型的训练。

每折采用感知机算法，即模型误分类数最小和随机梯度下降更新参数。最后返回每折的模型参数（直线的法向量和截距）和模型的验证分数。最佳模型取验证分数最小的模型参数。

7 可视化模型结果并分析

某次实验结果中模型在可视化如图 2所示。由图 2可以看出，模型对训练集和测试集的分类效果并不是非常好。分类线距离训练集中正负实例点的距离非常近。因此，考虑换一个策略，使用 SVM 算法使训练集中的数据类别间隔最大化，而且允许一定的误分类点。

8 支持向量机算法

考虑到 SVM 中 SMO 算法比较复杂，而且有成熟的库模块可以使用。因此基于避免重复造轮子的考虑，直接使用 scikit-learn 库中的 SVM 算法模型，重点在于库 API 的使用和结果的分析。另外本模块不再使用之前的方法通过函数参数获得数据，而是换个思路使用 pands 来处理 data.py 中生成的 csv 数据。

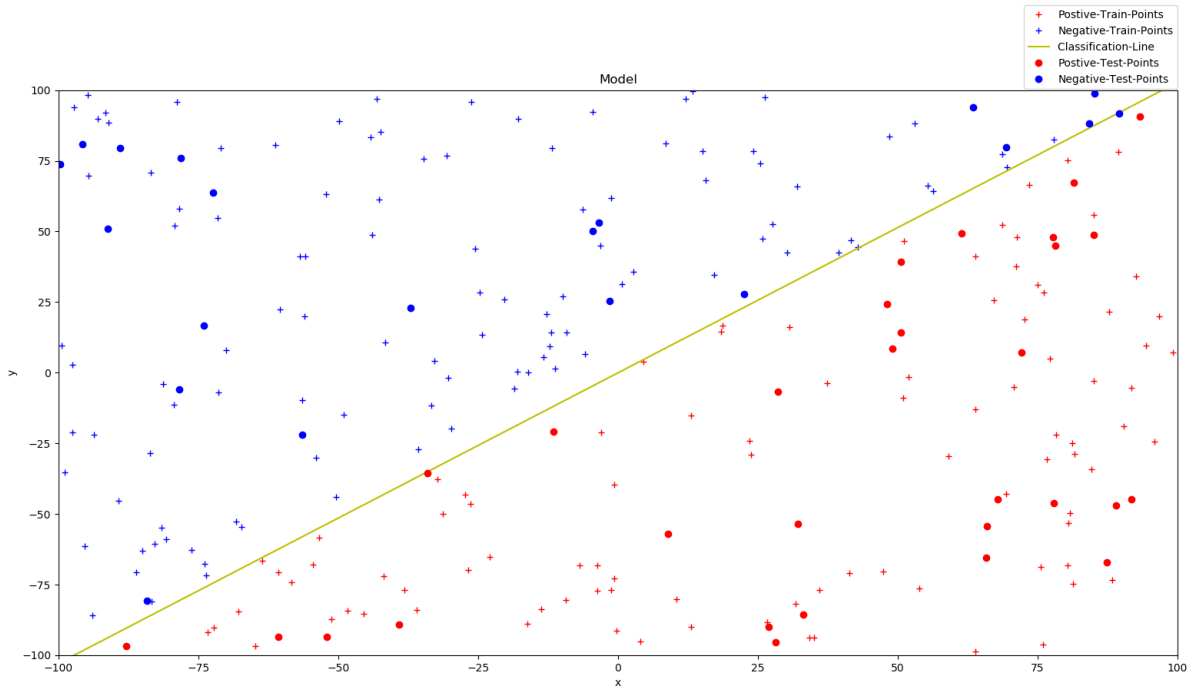

图 2: 感知机算法生成的模型可视化结果

在同样的数据下数据可视化结果如图 3所示 实验结果分析：从图 3可以看出，分类线直接比感知机算法得出的结果要更好，可以将测试集的数据正确分类。

但是, 同样存在一些问题，分类线与正负实例的距离过近。问题的源头来源于数据集，在 data.py 中，我们采用的是 $0-1$ 均匀分布生成位于 [0,1] 之间的数据，然后将这些数据通过线性运算扩展区域到 [-100,100] 之内，均匀分布的特点造成了数据在直线 $x-y=0$ 附近会出现这样的密集度问题，从而在直线附近会出现标记的正负实例点。因此造成上述所说的问题。

9 实验总结

从上面的陈述中可以得出以下结论：

1、数据的正确性对后续算法、模型的产生非常重要。一定要对原始数据进行一定的处理，包括去噪声、清洗等。如果输入算法的数据有问题，那么会对之后的结果分析产生非常大的干扰。

2、一个问题通过不同的算法求解可以得出不同的结果，通过对结果的分析，找到不足的原因然后进行优化选择。

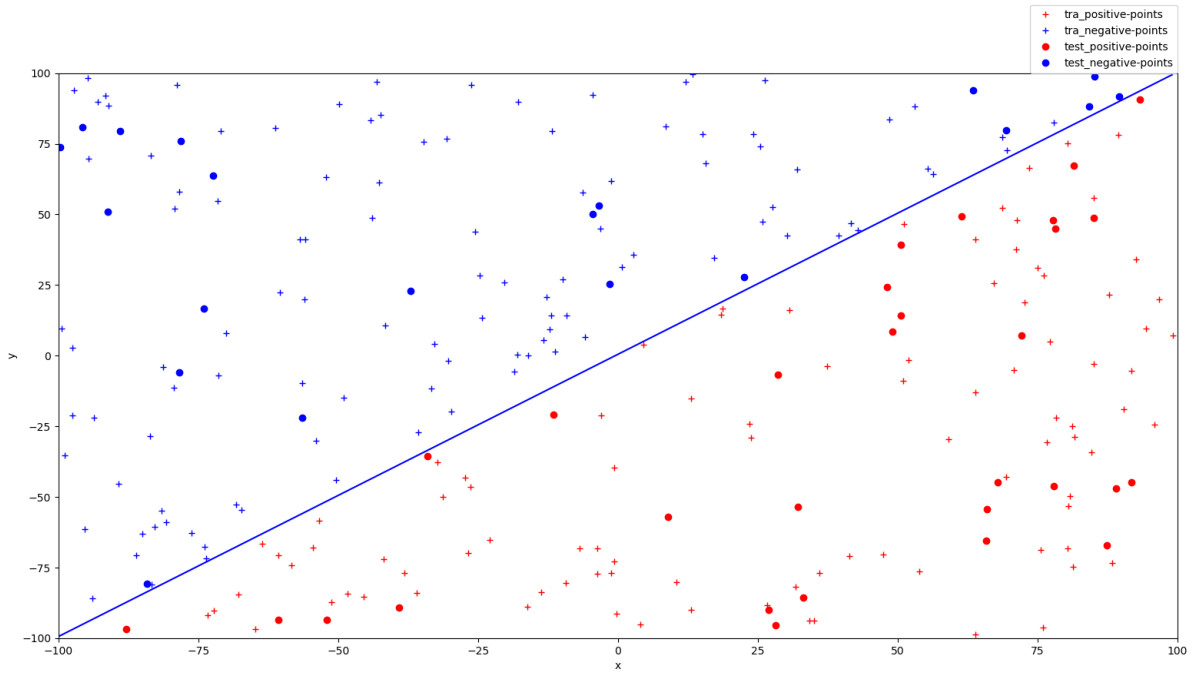

图 3: SVM 算法生成的模型可视化结果