

模式识别与机器学习 Assignment-1 报告

生成数据

见 source.py 中的 generate_data() 函数。

数据量为每个标号300，数据组织方式 data 为 $(n, 3)$ 的矩阵，其中前两列表示坐标，第三列表示标号。

discriminative model

见 source.py 中的 class discriminative_model .

该模型可以训练三组参数，每组参数三个，分别对应 x , y 坐标以及常数。每组参数都根据坐标给出对应标号的概率，储存在对象 self.w 中。

construct_matrix 方法返回计算、训练所需的矩阵。我们将 x 从 $(n, 2)$ 扩展为 $(n, 3)$, 最后一列恒为 1 , 以训练常数项。同时将标号用 *one-hot* 编码，返回一个 $(n, 3)$ 的矩阵 y .

model 方法对于给定的一组坐标和参数，返回一个 $(n, 3)$ 的矩阵，每一行表示一个数据点对应三种标号的概率。

gradient 方法对于给定的 x , y 和当前参数 Theta 返回一个 $(3, 3)$ 的梯度矩阵。

train 方法对模型进行训练，才用随机梯度下降方法。目前采用的参数 epoch, batch_size, learning_rate 分别为 1000, 50, 0.1 .

predict 方法对数据进行测试，并绘图。

generative model

见 `source.py` 中的 `class generative_model` .

该模型通过输入数据得到三个高斯分布（即他们的均值，协方差）以及三种标号的先验概率，不需要迭代地训练，分别储存在对象 `self.e` , `self.cov` , `self.theta` 。

`train` 方法可以得到上述三个参数。

`classify` 方法对于给定的坐标返回一个标号

`predict` 方法对数据进行测试并绘图。

对比

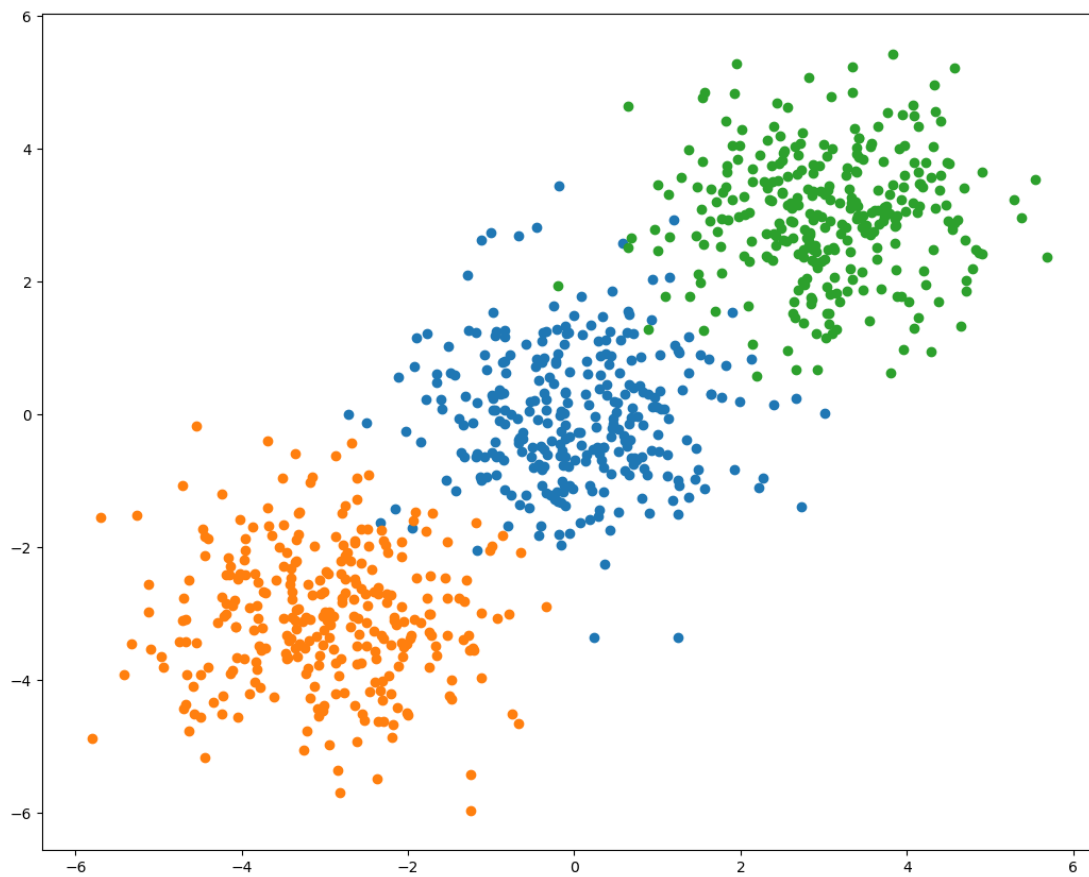
当数据完全线性可分时，`discriminative model`在 `epoch` 不够大时不能全部预测正确，而 `generative model`的几乎总是全对。

当数据线性不可分时，两者的正确率基本相同，分别为 0.981 和 0.979 。

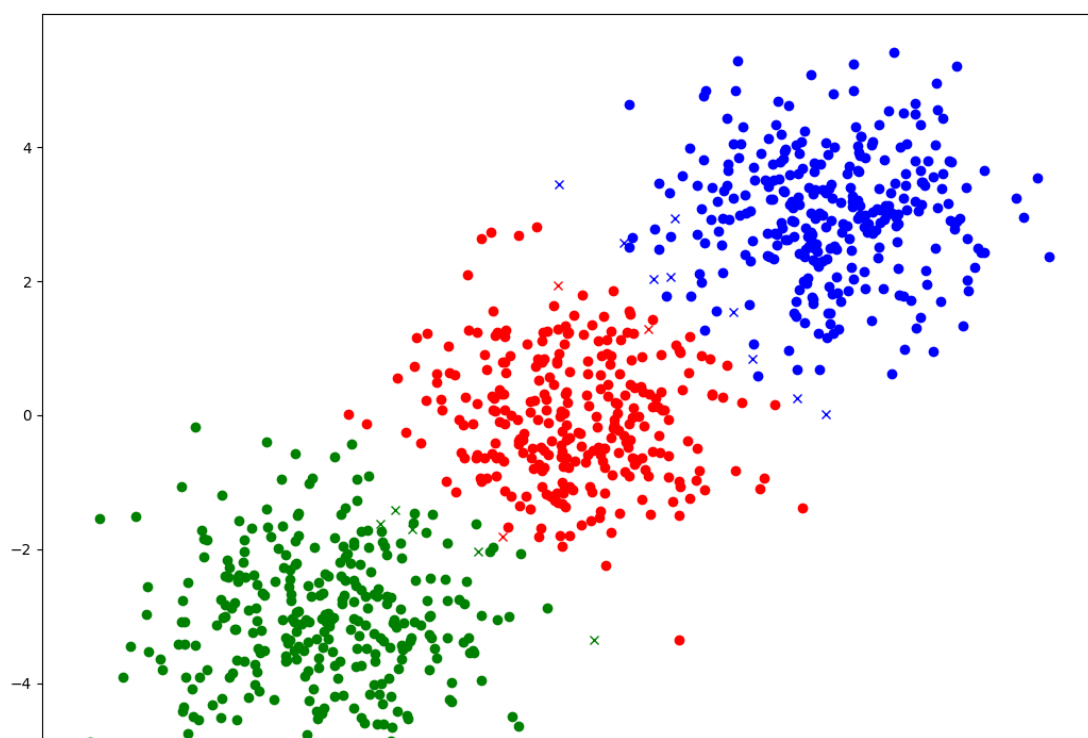
下面三个图例分别为：数据集，`discriminative model`的结果，`generative model`的结果。

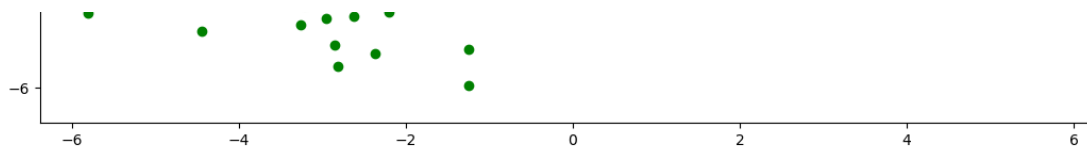
运行代码

`python source.py`

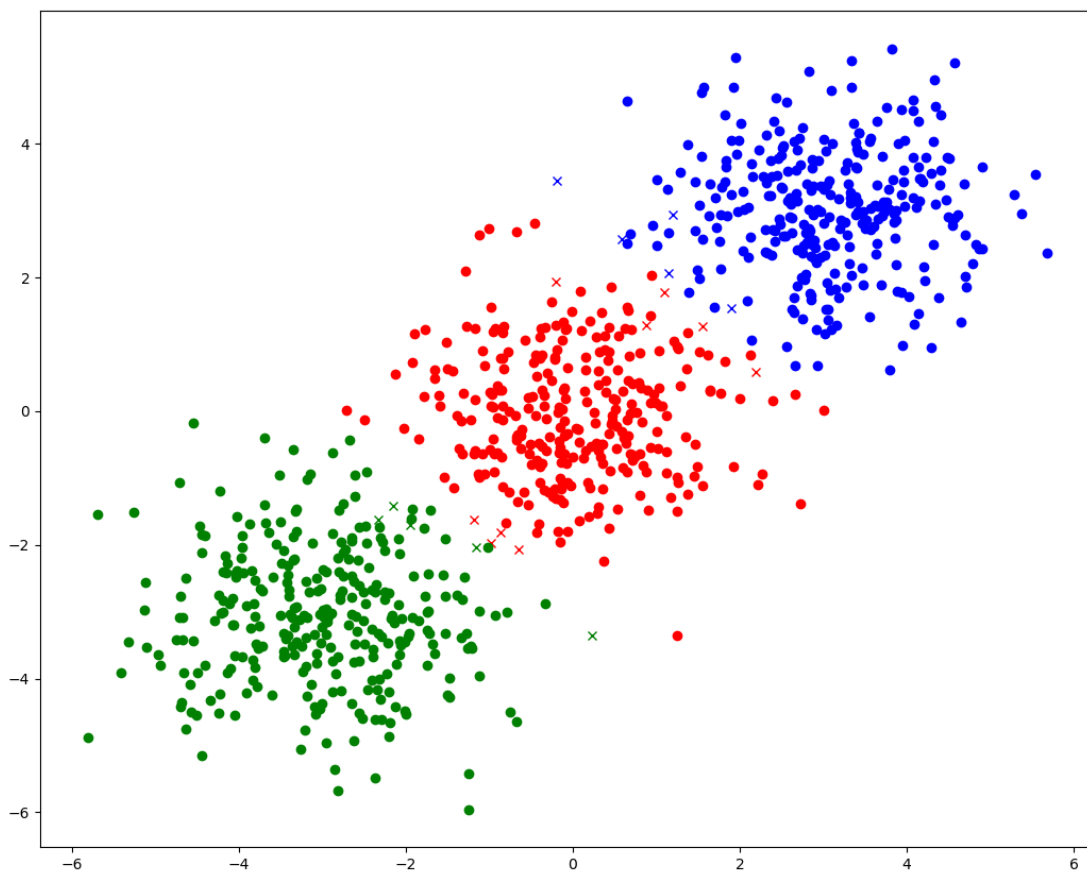


<https://blog.csdn.net/u012076197>





<https://blog.csdn.net/u012076197>



<https://blog.csdn.net/u012076197>