Report

问题描述

part1：设计三个高斯分布，分别标记为类别 A、B、C，从这三个高斯分布中抽样得到数据集。

part2：构建 2 个线性分类模型：生成模型和判别模型。比较它们之间的区别。

part3：重新组织数据集的规模或调整高斯分布之间的重叠。

代码说明

1、代码运行方式：

在命令行中定位到相应文件夹下，输入命令 python source.py，即可运行程序。运行程序后会按顺序展示三张图，依次为正确的分类、生成模型的分类结果、判别模型的分类结果。同时，命令行中会显示生成模型和判别模型的分类正确率。

2、大数据集链接：

大数据集存放在百度网盘中

链接：https://pan.baidu.com/s/1CIKj-CHFl0uw3kcROnVcXw

提取码：45wu

3、其它说明：

生成数据集的函数也包含在了 source.py 中，由于已经提供了小数

据集 gauss.data，生成数据部分的函数不会在运行程序时被调用。

Part 1

为了便于作图观察，生成数据集时选择使用二维高斯分布。

对每个二维高斯分布，定义 mean、cov、num 三个参数，分别代表它的均值、协方差矩阵、样本数量。在大数据集中，三个高斯分布的均值分别为(0,0)，(10,10)，(20,0)，协方差矩阵均为$\begin{bmatrix}10 & 0\\ 0 & 10\end{bmatrix}$，样本数量均为 1000。在抽样完成后，对所有的样本进行随机排列，使不同类别的样本均匀分布在数据集中。

提交的小数据集中，三个高斯分布的均值和协方差矩阵与大数据集相同，样本数量修改为各 50 个样本点。

在之后的模型训练部分，数据集会被分为训练集与测试集。其中训练集占 80%，用于对模型进行训练；测试集占 20%，用于对模型的分类准确性进行评估。所有的图像都是采用测试集中的样本点绘制得到的。

Part 2

1、生成模型

生成模型使用的是朴素贝叶斯模型。

step 1 求出 A、B、C 三个类别的先验概率 P(A)、P(B)、P(C)。

记训练集中三个类别的样本数量分别为$\mathrm{n_a}$，$\mathrm{n_b}$，$\mathrm{n_c}$，则：

$$\mathrm{P(A)} = \frac{\mathrm{n_a}}{\mathrm{n_a} + \mathrm{n_b} + \mathrm{n_c}} \quad \mathrm{P(B)} = \frac{\mathrm{n_b}}{\mathrm{n_a} + \mathrm{n_b} + \mathrm{n_c}} \quad \mathrm{P(C)} = \frac{\mathrm{n_c}}{\mathrm{n_a} + \mathrm{n_b} + \mathrm{n_c}}$$

step 2 求出三个类别的似然函数 P(x|A)、P(x|B)、P(x|C)。

二维正态分布的最大似然估计为:

$$\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} \qquad \Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu)(x^{(i)} - \mu)^T$$

关于 A、B、C 的最大似然函数分别为:

$$P(x|A) = \frac{1}{2\pi * |\Sigma_A|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(x - \mu_A)^T {\Sigma_A}^{-1}(x - \mu_A)\}$$

$$P(x|B) = \frac{1}{2\pi * |\Sigma_B|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(x - \mu_B)^T {\Sigma_B}^{-1}(x - \mu_B)\}$$

$$P(x|C) = \frac{1}{2\pi * |\Sigma_C|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(x - \mu_C)^T {\Sigma_C}^{-1}(x - \mu_C)\}$$

step 3 分类

由贝叶斯公式可知，后验概率正比于先验概率*似然函数，即:

$$P(A|x) \propto P(x|A) * P(A)$$

$$P(B|x) \propto P(x|B) * P(B)$$

$$P(C|x) \propto P(x|C) * P(C)$$

对于每一个测试集中的样本 x，将 x 代入三种类别的似然函数中，进一步求得$P(x|A) * P(A)$、$P(x|B) * P(B)$、$P(x|C) * P(C)$的值，三个值中的最大值对应的类别就是该生成模型给出的分类。

在 Part1 使用的数据集中，该模型的分类准确率为 98%。

2、判别模型

判别模型使用的是逻辑斯蒂回归。

step 1 定义模型和超参数

预测函数使用最简单的线性函数:

$$y = \sigma(wx + b)$$

其中 w 是一个 3*2 的参数矩阵，x 是一个样本点，b 是一个 3*1 的参数矩阵表示偏置，σ是 sigmoid 函数，用于将线性函数的输出映射到(0,1)中。y 为函数的输出，也是一个 3*1 的矩阵，三个输出的数分别表示样本被分到 A、B、C 三类的概率。

由于模型和问题都比较简单，模型在学习过程中很容易达到收敛状态, 可以设置较小的学习率和较少的 epoch 数, 同时没有加入 batch 的设定, 而是在每个 epoch 都让模型利用训练集中所有的数据进行训练和参数更新。在经过一系列尝试后，设置学习率为 0.01，epoch 为 500，此时模型可以得到比较好的效果。

step 2 选择损失函数

选择损失函数时发现，最小均方误差(MSE)不适合此类问题，所以选用交叉熵作为损失函数，具体如下:

$$\text{Loss Function} = -\sum_{i=1}^{n}\sum_{j=1}^{3}[t_j \log(y_j) + (1 - t_j)\log(1 - y_j)]$$

其中 n 为样本数量，y_j表示样本预测为类别 j 的概率，t_j表示样本是否属于类别 j，1 表示属于，0 表示不属于。

step 3 训练模型

训练模型采用普通的梯度下降法，即:

$$w \leftarrow w - \alpha \frac{\partial L}{\partial w} \qquad b \leftarrow b - \alpha \frac{\partial L}{\partial b}$$

其中α为学习率。将损失函数代入上式后得到:

$$w \leftarrow w - \alpha * (t - y) * x \qquad b \leftarrow b - \alpha * (t - y)$$

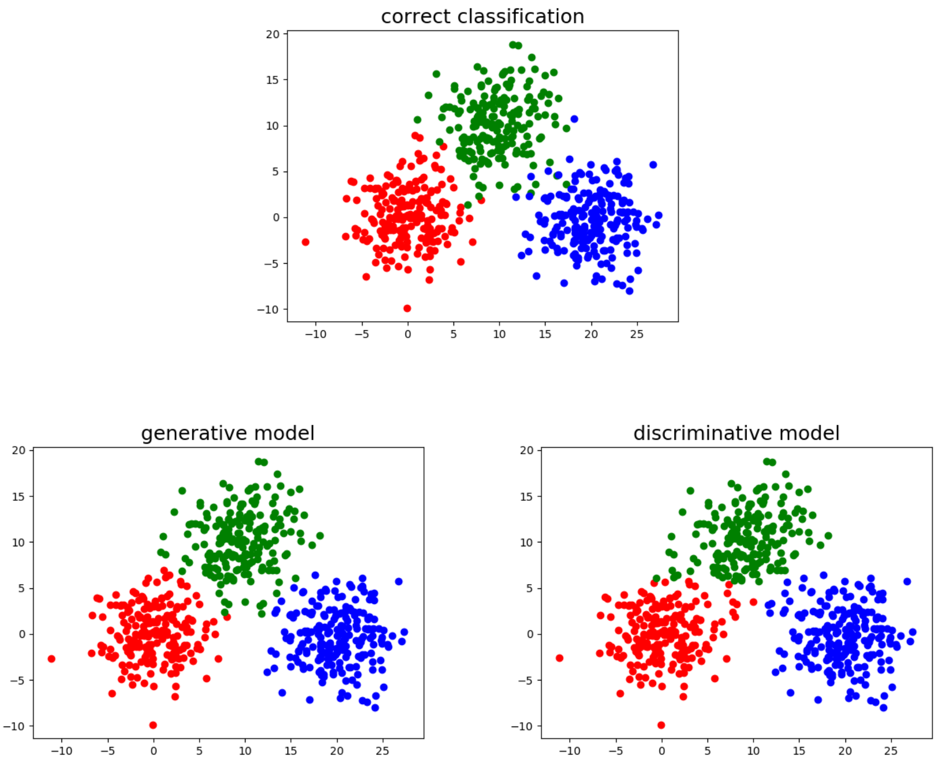
step 4 分类

将测试集的样本 x 代入训练完成后的预测函数，得到输出 $y = [y_1, y_2, y_3]^T$，其中数值最大的y_i对应的类别即为该判别模型给出的分类。

在 Part1 使用的数据集中，该模型的分类准确率为 95.83%。

3、模型比较

两种模型在大数据集上的分类效果如下图:



在分类准确率方面，生成模型和判别模型在大数据集上的准确率分别为 98%和 95.83%，两个模型都有较高的分类准确率。通过观察上面的图片，也可以得到同样的结论。

在运行时间方面，生成模型的运行时间与样本数量成正比，而判别模型的运行时间同时与样本数量和 epoch 数量成正比。在这个分类问题中，判别模型的运行时间要远多于生成模型。

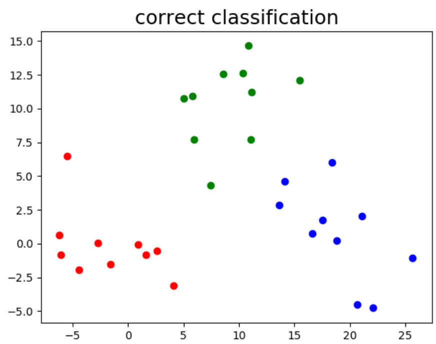
在参数数量方面，由于此问题的样本均服从二维高斯分布，两类模型需要学习的参数数量相差不大。但当高斯分布的维数增加，或者问题更加复杂时，生成模型需要学习的参数更多。

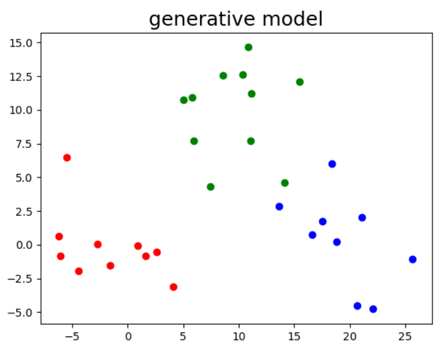
生成模型通常需要预设参数的分布，判别模型不能用来生成数据。由于这一问题的特殊性，两个模型的这些缺点并没有体现出来。

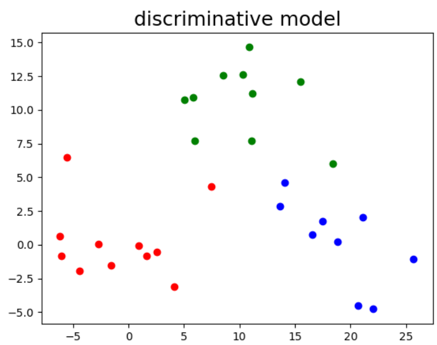
Part 3

1、调整数据规模

减少样本数量，从每一类中抽样 50 个样本点。分类效果如下图：

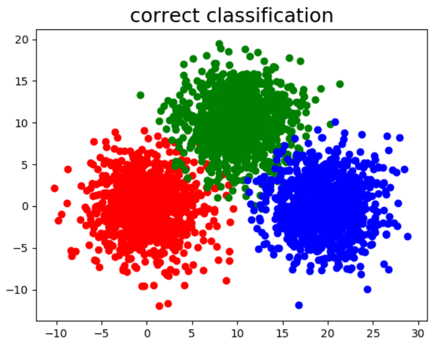


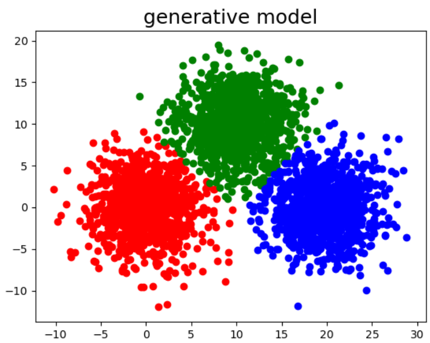


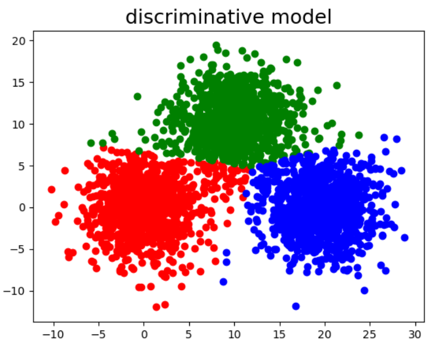


生成模型准确率 96.67%，判别模型准确率 93.33%。因为设计的高斯分布重叠较小且各类样本点数量均等，所以小样本不会对分类准确率产生太大影响。

增加样本数量，从每一类中抽样 5000 个样本点。分类效果如下图：



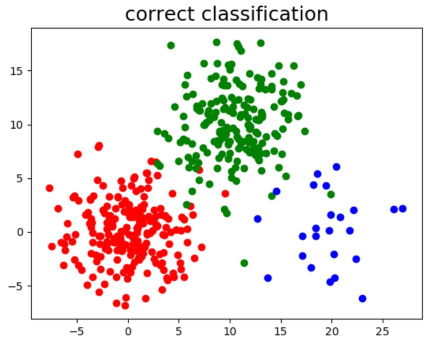


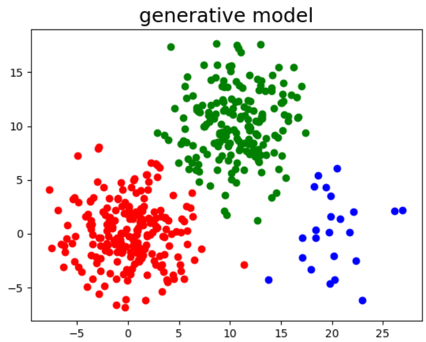


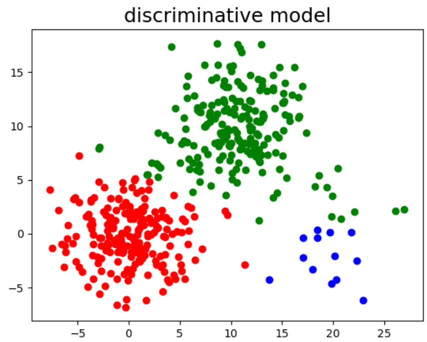
生成模型准确率 98.13%，判别模型准确率 96.1%。观察图像可以发现，两类模型的分类结果都出现了比较明显的边界，且判别模型有明显的线性决策边界。

2、调整数据分布

显著降低某一类的样本数量，三个类别的抽样数量分别为 1000、1000、100，分类效果如下图：

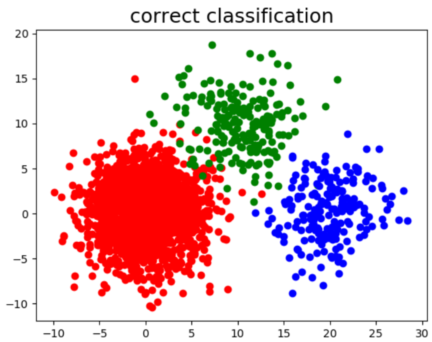


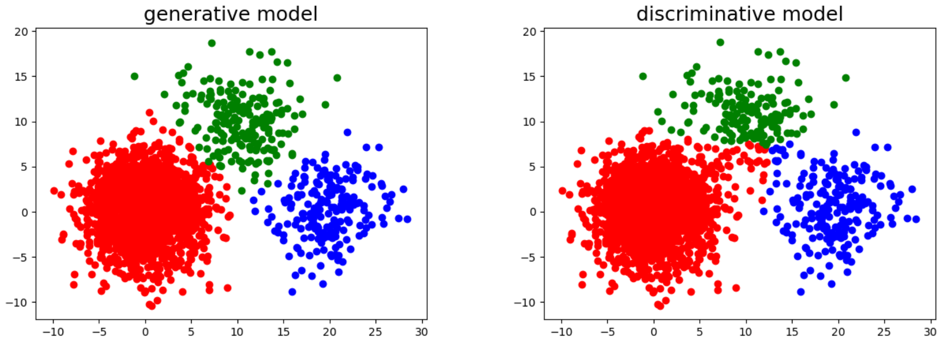




生成模型准确率 97.62%，判别模型准确率 94.29%。观察图像可以发现，判别模型在对蓝色样本进行分类时错误率很高，猜测原因是样本数量少导致对这类样本的学习次数少，模型没有达到收敛。将 epoch 从 500 修改为 5000 后，分类准确率提升至 96%，且对蓝色样本的分类准确率明显提升。

显著提升某一类的样本数量，三个类别的抽样数量分别为 10000、1000、1000，分类效果如下图：

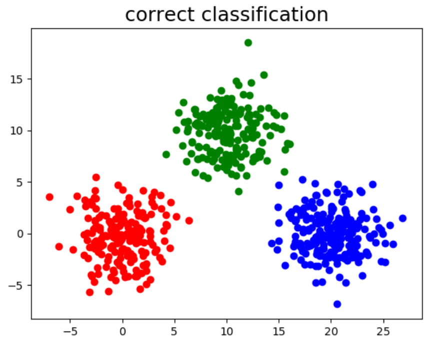


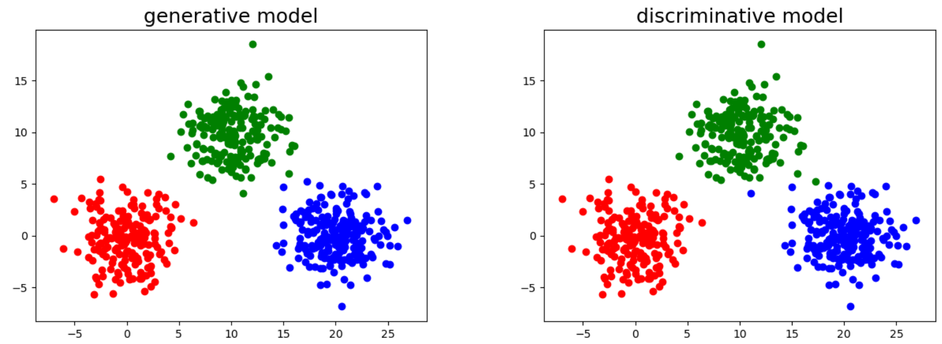


生成模型准确率 99.04%，判别模型准确率 97.71%。观察图像可以发现，由于红色样本点较多，判别模型会将部分绿色和蓝色的点分类为红色导致准确率下降。

3、调整重叠

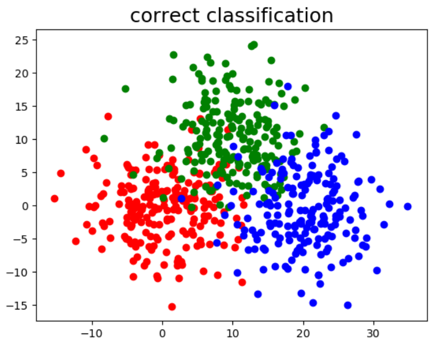
减少高斯分布之间的重叠部分，将三个高斯分布的协方差矩阵均修改为$\begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$，分类效果如下图：

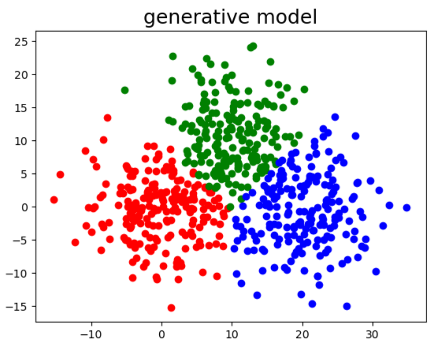


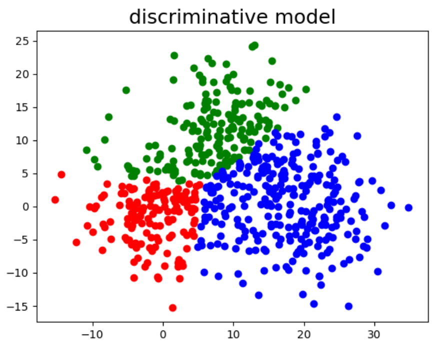


生成模型准确率 100%，判别模型准确率 99.67%。当各个高斯分布之间几乎没有重叠部分时，两个模型的准确率都非常高。如果继续减少重叠部分，两个模型的准确率都会达到 100%。

增加高斯分布之间的重叠部分，将三个高斯分布的协方差矩阵均修改为$\begin{bmatrix}25 & 0\\ 0 & 25\end{bmatrix}$，分类效果如下图：







生成模型准确率 88.67%，判别模型准确率 76.83%。当各个高斯分布存在大面积的重叠时，两个模型的准确率都显著降低。如果继续增加重叠部分，两个模型的准确率还会继续下降。

4、总结

在不同的数据集上调用生成模型和判别模型进行分类时，生成模型的准确率都会略高于判别模型。猜想这是因为，生成模型的分类准

确率与预先设定的概率分布有关，而在这个问题中已知数据集是通过高斯分布生成的，求得的似然函数与产生数据集的高斯分布的概率密度函数非常接近，因此获得了较高的分类准确率。判别模型在这个问题中，只能给出线性的决策边界，这也在一定程度上影响了判别模型的表现。

生成模型和判别模型的分类准确率都会受到数据的影响，调整各个高斯分布的重叠对分类准确率的影响最大。