实验报告

实验描述

PART I

生成一个数据集，包含三个分别被标注为0，1，2的高斯分布

PART II

分别用生成模型、判别模型对给定的训练集进行训练，并对给定的测试集进行分类

PART III

对分类模型进行测试

实验思路

首先，使用numpy库的库函数random.multivariate_normal进行数据的生成。

显然，有$P(c_1|x)=\frac{P(x,c_1)}{\sum P(x,c_i)}$。令$P(x,c_i)=exp(a_i)$，$a_i=w_i^T*x+w_{i0}$，分别通过生成模型和判别模型训练，得到系数向量$w_i$和偏置向量$w_{i0}$。对于每个一测试数据$x$，通过$softmax$函数处理求得的$a_i$，并最终通过对得到的新的$a_i$用$argmax$处理得到概率最大的类别，该类别被认为时$x$所属于的类别。

模型实现

生成模型

根据高斯分布，令每一类的协方差矩阵相同，有

$P(x,c_i)=P(c_i)N(x|\mu_i,\Sigma)=P(c_i)-\frac{1}{2\pi*|\Sigma|}exp(-\frac{1}{2}(\mu_i-x)^T*\Sigma^{-1}*(x-\mu_i))$

通过最大似然估计，得到

$\mu_i=\frac{\sum(t_x==i)}{N}$，$\Sigma=\sum\frac{1}{N}(x-\mu_{t_x})(x-\mu_{t_x})^T$

之后，把μ_i和Σ 代入到a_i中，得到

$w_i=-\sum\mu_i^T\Sigma^{-1}$，$w_0=-\frac{1}{2}\mu^T\Sigma^{-1}\mu+ln(prior_i)$。

判别模型

通过设置mini_batch的方法，设置epoch次迭代，每次将所有数据分成若干个大小为batch的组，对于每组的训练数据用梯度下降进行训练。

对于一个数据x，令t是一个one-hot的向量表示其注释。通过当前w_i和w_i0利用$softmax$算出的结果为y，那么就设损失函数为$t-y$。用这个损失函数对w_i和w_i0求偏导，得到$\nabla w_i=-\sum(y_i-t_i)*x$，$\nabla w_{i0}=-\sum(y_i-t_i)$。对于每个$batch$内的数据先求出$y$，再让$w$，$w_0$分别减去其梯度进行更新。

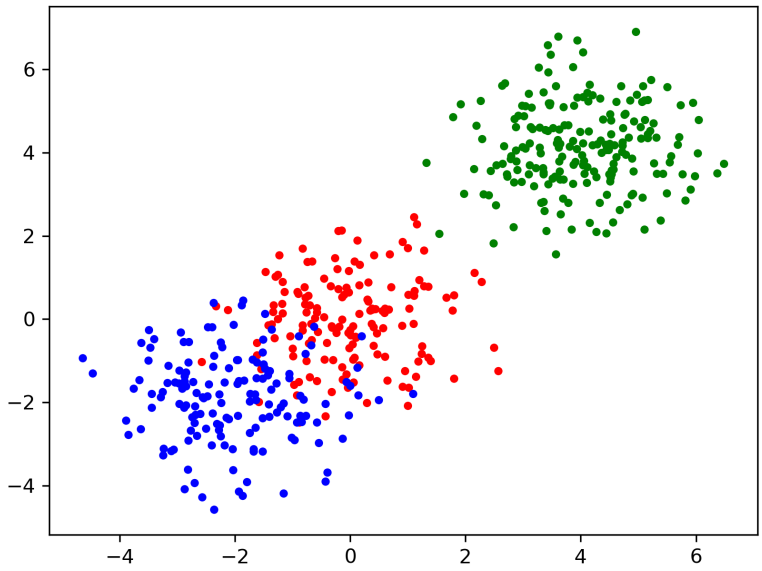
结果展示

PART I

默认生成的高斯分布参数设置：

$\mu_A = (0,0)$，$\mu_B = (-2,-2)$，$\mu_c = (3,2)$，$\Sigma = ((1,0),(0,1))$，$prior = (0.3,0.3,0.4)$，$n = 500$

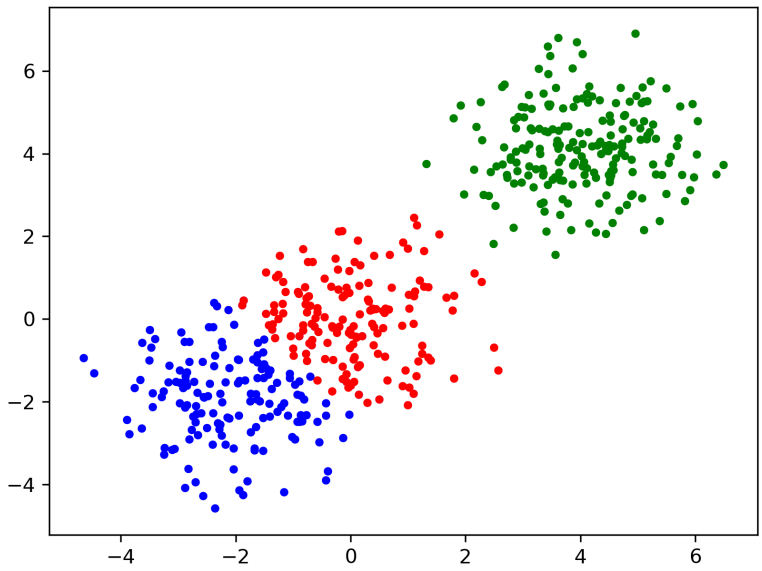
三类点的分布图：



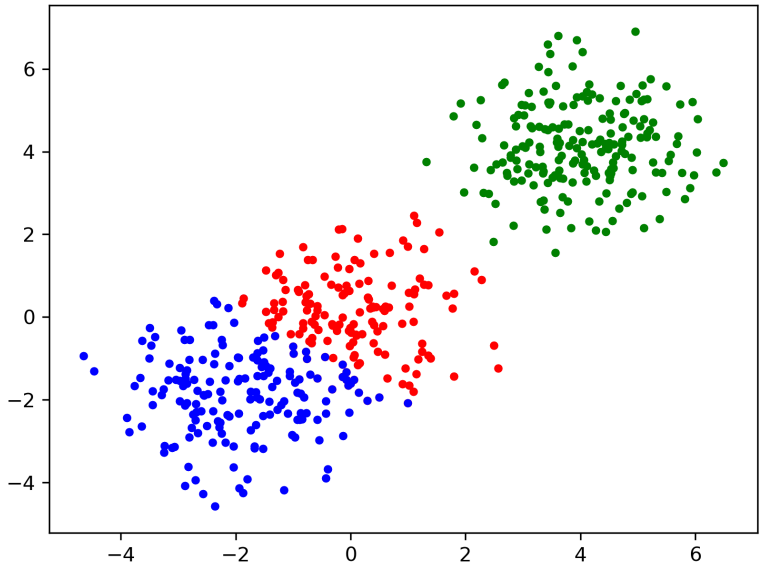
PART II

根据PART I所示的数据，在学习率0.7下的分类结果：

生成模型：0.93



判别模型：0.922



生成的高斯分布数据的准确度的影响：

测试一：

数据生成的高斯分布参数设置：

$\mu_A = (0,0)$，$\mu_B = (-2,-2)$，$\mu_c = (3,2)$，$\Sigma = ((1,0),(0,1))$，$prior = (0.3,0.3,0.4)$，$n = 500$

学习率0.7

判别模型的参数设置：

$batch = 32$，$epoch = 50$，

得到生成模型、判别模型的五次测试正确率如下：

0.9268，0.9208

测试二：

数据生成的高斯分布参数设置：

$\mu_A = (0,0)$，$\mu_B = (-1,-1)$，$\mu_c = (1.5,1)$，$\Sigma = ((1,0),(0,1))$，$prior = (0.3,0.3,0.4)$，$n = 500$

学习率0.7

判别模型的参数设置：

$batch = 32$，$epoch = 50$，

得到生成模型、判别模型的五次测试正确率如下：

0.7276，0.6852

测试三：

数据生成的高斯分布参数设置：

$\mu_A = (0,0)$，$\mu_B = (-2,-2)$，$\mu_c = (3,2)$，$\Sigma = ((2,0),(0,2))$，$prior = (0.3,0.3,0.4)$，$n = 500$

学习率0.7

判别模型的参数设置：

$batch = 32$，$epoch = 50$，

得到生成模型、判别模型的五次测试正确率如下：

0.8212，0.778

当生成的数据点的均值较为接近，或者协方差矩阵较大，同一种点的分布较为分散时，在分类时会造成产生在某个区域中同时存在多个类型的点。在此情况下，判别模型和生成模型的正确率都出现了较大幅度的下降。由于在最后是使用argmax函数对某个点进行分类的，当出现某个点在每个种类下都有较大概率的时候，只能选择最大的那个作为分类结果。若这几个概率的值较为接近，很容易出现分类出错的现象。在分类的结果图中，三类点的分布区域都有明显的界

限，这与实际上的高斯分布出现不同。而这个问题也是按照argmax函数对概率进行分类无法避免的

另外，在测试中，发现生成模型在不同种类的点分布区域相互混合较为严重的测试（测试二、三）中，在多次测量时极少数情况下会出现正确率大幅下降地现象（对同一组数据测试从0.8左右跌至0.6），猜测可能与测试数据的选取顺序有关。但是，在实际实现过程中，在每个$epoch$内都会让数据进行打乱操作，应该不会因为数据的选取而产生这么大的误差。

测试四：

数据生成高斯分布参数设置：

$\mu_A = (0,0)$，$\mu_B = (-2,-2)$，$\mu_c = (3,2)$，$\Sigma = ((2,0),(0,2))$，$prior = (0.3,0.3,0.4)$，$n = 500$

学习率1.0

判别模型参数设置：

$batch = 32$，$epoch = 50$,

得到判别模型五次测试平均正确率：0.7936

测试五：

数据生成高斯分布参数设置：

$\mu_A = (0,0)$，$\mu_B = (-2,-2)$，$\mu_c = (3,2)$，$\Sigma = ((2,0),(0,2))$，$prior = (0.3,0.3,0.4)$，$n = 500$

学习率1.0

判别模型参数设置：

$batch = 2$，$epoch = 50$,

得到判别模型十次测试平均正确率：0.7546

以上两组测试都为对同一测试集、训练集进行的测试。在测试中发现，当batch过大时会造成w一次的变化过大，即逆梯度方向的该变量过大，导致exp运算溢出；当batch调小时，会导致正确率出现突然下降的现象频率增加，而且训练出来的正确率普遍较低，估计是因为每次只对某少量数据进行梯度下降，导致对其余数据产生了较大的误差，每次只增加了某一个数据点方向上的梯度而没有考虑整体的梯度，而且噪点的影响会因此变大。

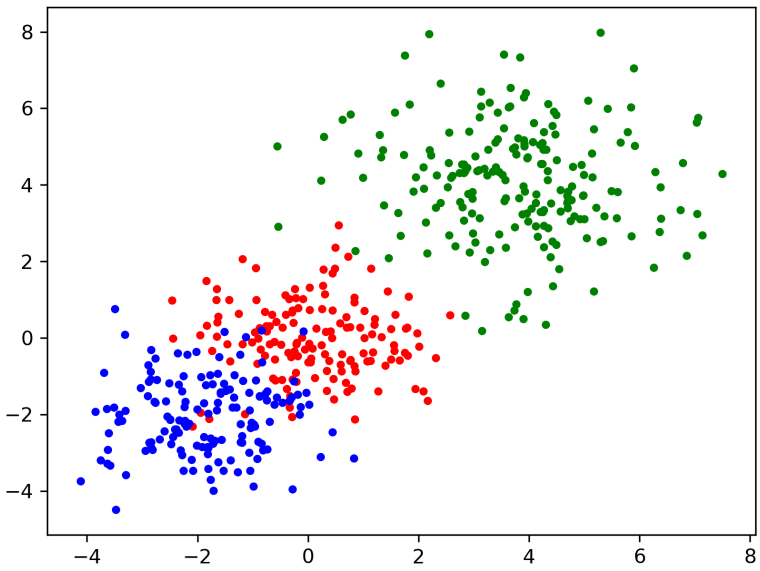
另外，在测试协方差矩阵不同的情况下两个模型的分类情况：

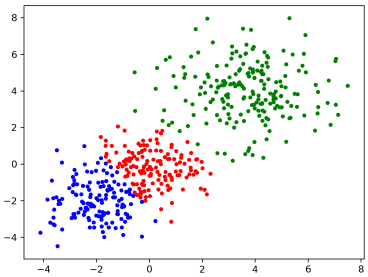
数据生成的高斯分布参数设置：

$\mu_A = (0,0)$，$\mu_B = (-2,-2)$，$\mu_c = (4,4)$，$prior = (0.3,0.3,0.4)$，$n = 500$

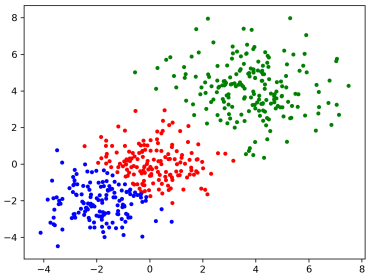
$\Sigma_A = ((1,0),(0,1))$，$\Sigma_B = ((1,0),(0,1))$，$\Sigma_C = ((2,0),(0,2))$

学习率设置为1.0





生成模型,ac=0.948



判别模型,ac=0.928

对于不同高斯分布协方差矩阵的数据，由于这是最大似然估计后的偏导数计算较为复杂，而且在网上并没有相关的参考资料，也就没有进行编写。在测试中发现，当数据中各种点的分布区域重叠并不严重时，相同协方差矩阵下的模型较为准确的分类。由于每种点分布概率最大值的区域（μ_i）和相同协方差矩阵下的情况相比并没有较大的变化，在两种点分布区域的边界上的点才会出现分类错误，而这类点出现的个数并不会太多（高斯分布的概率并不大），因此猜测上述模型在不同的协方差矩阵情况下误差的大部分来源还是点集出现区域的重叠与argmax函数分类结果相违背。