一、 数据集设置

使用标签 0 来代表类 A，标签 1 来代表类 B，标签 2 来代表类 C

初始设置：A、B 和 C 是三个 2 维高斯分布。三者的 x1 和 x2 都是相互独立的，其协方差矩阵为([1,0],[0,1])。A 的均值为[3,3]，B 的均值为[0,0]，C 的则是[-3,3]。

采样设置：三个类分别采样 800 个，作为训练集，共 2400 个。再分别采样 200 个，作为测试集，共 600 个。比例是 4:1。总共是数据 3000 个

提交设置：为了提交不大于 20kB 的.data 文件，重新进行了采样，三个类分别采样 80 个，作为训练集，共 240 个。再分别采样 20 个，作为测试集，共 60 个。比例仍是 4:1。总共是数据 600 个

二、 分类

1、判别式模型

设$\boldsymbol{y} = f(x) = \boldsymbol{Wx} + \boldsymbol{b}$

这里，$\boldsymbol{W}$是 3×2 的矩阵，$\boldsymbol{x}$是 2×1 的向量，属于变量，$\boldsymbol{b}$是 3×1 的向量，属于偏置。

输出$\boldsymbol{y}$也是一个 3×1 的向量。

此处的 3 代表有 3 类不同的高斯分布，2 代表这是一个二元正态分布，$\boldsymbol{x}$是两维的。

使用 SoftMax 函数来归一化处理，所以有：

$$\boldsymbol{p} = \boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x}) = \boldsymbol{softmax}(\boldsymbol{y}) = \boldsymbol{softmax}(\boldsymbol{Wx} + \boldsymbol{b})$$

这里，$\boldsymbol{p}$是一个 3×1 的向量，3 个值分别代表分到 3 个类中的概率。

接下来进行训练---

使用**交叉熵损失函数**，**随机梯度法，单步处理**，于是有：

$$\boldsymbol{L} = -\sum_{\boldsymbol{i}=\boldsymbol{0}}^{\boldsymbol{2}} \boldsymbol{t_i} \boldsymbol{lnp}(\boldsymbol{y_i}|\boldsymbol{x})$$

其中，$\boldsymbol{t_i}$是向量$\boldsymbol{t}$中的元素，代表样本是否被分在第$\boldsymbol{i}$类，如果是则为 1，否则为 0。

另外，$\boldsymbol{p}(\boldsymbol{y_i}|\boldsymbol{x})$则是上面向量$\boldsymbol{p}$的元素，代表当前分到第$\boldsymbol{i}$类的概率。并且，有：

$$\boldsymbol{p}(\boldsymbol{y_i}|\boldsymbol{x}) = \boldsymbol{softmax}(\boldsymbol{y_i}) = \frac{\boldsymbol{e}^{\boldsymbol{y_i}}}{\sum_{\boldsymbol{j}=\boldsymbol{0}}^{\boldsymbol{2}} \boldsymbol{e}^{\boldsymbol{y_j}}}$$

然后要计算两个参数的梯度$\boldsymbol{G_W}$和$\boldsymbol{G_b}$，于是有：

$$\boldsymbol{G_W = \frac{\partial L}{\partial W} \qquad G_b = \frac{\partial L}{\partial b}}$$

但是，$\boldsymbol{L}$的表达式里不带$\boldsymbol{W}$和$\boldsymbol{b}$，所以要间接去求，于是有：

$$\boldsymbol{G_W = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial W} \qquad G_b = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial b}}$$

接下来求$\boldsymbol{\frac{\partial L}{\partial y}}$，这明显是一个向量，其三个元素分别为$\boldsymbol{\frac{\partial L}{\partial y_0}}$，$\boldsymbol{\frac{\partial L}{\partial y_1}}$和$\boldsymbol{\frac{\partial L}{\partial y_2}}$

所以以$\boldsymbol{\frac{\partial L}{\partial y_0}}$为例，来进行求值：

$$\boldsymbol{\frac{\partial L}{\partial y_0} = \frac{\partial - \sum_{i=0}^{2} t_i lnp(y_i|x)}{\partial y_0} = \frac{\partial - ln\frac{e^{y_k}}{\sum_{j=0}^{2} e^{y_j}}}{\partial y_0}}$$

需要说明的是，这里的$\boldsymbol{k}$是训练集的值，代表样本所属的类，现在不知道是多少。

化简上式得：

$$\boldsymbol{\frac{\partial - ln\frac{e^{y_k}}{\sum_{j=0}^{2} e^{y_j}}}{\partial y_0} = \frac{\partial ln\frac{\sum_{j=0}^{2} e^{y_j}}{e^{y_k}}}{\partial y_0} = \frac{\partial ln\sum_{j=0}^{2} e^{y_j} - lne^{y_k}}{\partial y_0} = \frac{\partial [ln\sum_{j=0}^{2} e^{y_j} - y_k]}{\partial y_0}}$$

所以，原式要进行分类讨论：

若$\boldsymbol{k} = 0$，则$\boldsymbol{y_k = y_0}$，那求偏导的结果就是

$$\frac{\boldsymbol{y_0}}{\boldsymbol{\sum_{j=0}^{2} e^{y_j}}} - 1$$

若$\boldsymbol{k} \neq 0$，则$\boldsymbol{y_k} \neq \boldsymbol{y_0}$，那求偏导的结果就是

$$\frac{\boldsymbol{y_0}}{\boldsymbol{\sum_{j=0}^{2} e^{y_j}}}$$

推广到一般的情况，即$\boldsymbol{\frac{\partial L}{\partial y_i}}$的情况，则有：

$$\boldsymbol{\frac{\partial L}{\partial y_i}} = \frac{\boldsymbol{y_i}}{\boldsymbol{\sum_{j=0}^{2} e^{y_j}}} - \{1 \text{ 或 } 0\}(\text{当 k = i 时为 1，否则为 0})$$

为了表示方便，设向量$\boldsymbol{K}$为除了第 K 位为 1，其余位为 0 的 1×3 向量。

所以原式又可以表示为：

$$\boldsymbol{\frac{\partial L}{\partial y}} = \frac{\boldsymbol{y}}{\boldsymbol{\sum_{j=0}^{2} e^{y_j}}} - K = \boldsymbol{p - K}$$

其中，$\boldsymbol{p}$是前面所述的概率向量，此处正好形式相同。

然后求$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{W}}$和$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{b}}$，很显然有：

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{W}} = \boldsymbol{x} \quad \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{b}} = 1$$

所以，综上所述：

$$\boldsymbol{G_W} = \frac{\partial \boldsymbol{L}}{\partial \boldsymbol{W}} = (\boldsymbol{p} - \boldsymbol{K}) \cdot \boldsymbol{x}$$

$$\boldsymbol{G_b} = \frac{\partial \boldsymbol{L}}{\partial \boldsymbol{b}} = \boldsymbol{p} - \boldsymbol{K}$$

最后，设定学习率α，每当一个新的样本进入，就修改一次参数$\boldsymbol{W}$和$\boldsymbol{b}$

接下来是学习结果---

设定学习率为α为 0.008，学习步数 10000 次。

最后进行测试，600 个测试对象中，有 592 个命中。命中率为 98.67%。

2、生成式模型

目标是求 P(A|x)，P(B|x)和 P(C|x)，并找到他们的最大者，作为决策

而根据贝叶斯公式：

$$\mathrm{P(A|x)} = \frac{\mathrm{P(x|A) \cdot P(A)}}{\mathrm{P(x|A) \cdot P(A) + P(x|B) \cdot P(B) + P(x|C) \cdot P(C)}}$$

而 P(B|x)和 P(C|x)同理。

因此只需要求出分子的两项，进行比较就可以了。

第一步，求 P(A)，P(B)和 P(C)：

在采样的时候，已经按照三个类 1:1:1 的方式去采样，因此三者均为 1/3

第二步，求 P(x|A)，P(x|B)和 P(x|C)。

根据公式有：

$$\mathrm{P(x|A)} = \frac{\mathbf{1}}{\mathbf{2}\boldsymbol{\pi} \cdot |\boldsymbol{\Sigma_A}|^{\mathbf{1/2}}} \cdot \boldsymbol{exp}\{-\frac{\mathbf{1}}{\mathbf{2}}(\boldsymbol{x} - \boldsymbol{\mu_A})^{\boldsymbol{T}} \boldsymbol{\Sigma_A}^{-\mathbf{1}}(\boldsymbol{x} - \boldsymbol{\mu_A})\}$$

$$\mathrm{P(x|B)} = \frac{\mathbf{1}}{\mathbf{2}\boldsymbol{\pi} \cdot |\boldsymbol{\Sigma_B}|^{\mathbf{1/2}}} \cdot \boldsymbol{exp}\{-\frac{\mathbf{1}}{\mathbf{2}}(\boldsymbol{x} - \boldsymbol{\mu_B})^{\boldsymbol{T}} \boldsymbol{\Sigma_B}^{-\mathbf{1}}(\boldsymbol{x} - \boldsymbol{\mu_B})\}$$

$$\mathrm{P(x|C)} = \frac{\mathbf{1}}{\mathbf{2}\boldsymbol{\pi} \cdot |\boldsymbol{\Sigma_C}|^{\mathbf{1/2}}} \cdot \boldsymbol{exp}\{-\frac{\mathbf{1}}{\mathbf{2}}(\boldsymbol{x} - \boldsymbol{\mu_C})^{\boldsymbol{T}} \boldsymbol{\Sigma_C}^{-\mathbf{1}}(\boldsymbol{x} - \boldsymbol{\mu_C})\}$$

其中，μ和Σ的值可以查公式得到：

$$\boldsymbol{\mu} = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$

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

得到每一个样本后，将它们代入到上面的几个式子中即可求出上述值 。

由于这里的 P(A)，P(B)和 P(C)都为 1/3，甚至不用加上它们，直接比较 P(x|A)，P(x|B)和 P(x|C)都可以。不过在一般情况下，仍然需要比较它们的乘积。

最后根据贝叶斯公式，将二者的乘积进行比较。

接下来是学习结果--

最后进行测试，600 个测试对象中，有 591 个命中。命中率为 98.5%。

三、 进一步的讨论

在完成了基本的分类后，可以进行一些探索。

1、两元不独立（协方差不为 0）的情况

之前的采样中，均使用了 x1 和 x2 相互独立的二维高斯分布。其协方差为 0。

可以通过调整协方差矩阵，来让二者不独立，来观察其分类情况。

采样设置：A、B 和 C 类的协方差矩阵变为([1,0.8],[0.8,1])。三类的采样个数以及测试集、训练集的比例不变。

接下来是学习结果--

判别法：600 个测试对象中，有 576 个命中。命中率为 96%。

生成法：600 个测试对象中，有 575 个命中。命中率为 95.83%。

可以看出，在不独立的情况下，分类变难了，正确率有一定程度的下降。可能是因为协方差增大，让两个不同高斯分布之间的重叠增加了。

2、更多重叠部分的数据

之前的采样中，重叠部分不多。三者的均值分别为[3,3]，[0,0]和[-3,3]，相差不大。

可以通过缩小均值之间的距离，或者放大方差来增加三者的重叠。

采样设置：A、B 和 C 类的协方差矩阵变为([2,0],[0,2])。三类的采样个数以及测试集、训练集的比例不变。

接下来是学习结果---

判别法：600 个测试对象中，有 547 个命中。命中率为 91.67%。

生成法：600 个测试对象中，有 546 个命中。命中率为 91%。

接下来进一步增加三者的重叠部分。

采样设置：A、B 和 C 类的协方差矩阵变为([3,0],[0,3])。三类的采样个数以及测试集、训练集的比例不变。

接下来是学习结果---

判别法：600 个测试对象中，有 514 个命中。命中率为 85.67%。

生成法：600 个测试对象中，有 513 个命中。命中率为 85.5%。

可以看出，随着重叠部分的增加，分类的准确性也在下降，两种方法的下降程度近似。这是因为重叠的越多，产生误判的可能性也就越大。

3、分布形状不同的情况

之前的采样中，三者的分布虽然均值不同，但协方差矩阵相同，形状相同。

可以通过调整协方差矩阵，来让三者形状不同，来观察其分类情况。

采样设置：A 的协方差矩阵变为([1,0.8],[0.8,1])。B 的协方差矩阵变为([2,0],[0,2])。C 的协方差矩阵变为([1.2,0.7],[0.7,1.2])。三类的采样个数以及测试集、训练集的比例不变。

接下来是学习结果---

判别法：600 个测试对象中，有 561 个命中。命中率为 93.5%。

生成法：600 个测试对象中，有 547 个命中。命中率为 91.67%。

可以看出，在形状不同的情况下，分类变难了。由于形状不同，在类与类之间很难产生一个比较规则的界面，二者之间的区分界面可能是二次的曲线，这增加了区分的难度。而且，生成法的正确率下降要高于判别法，可能生成法在对于这种情况的耐受力并不如判别法好。

4、修改判别式方法的学习率

之前的采样中，使用了学习率α=0.008。

可以通过调整学习率，来观察其分类情况。

采样设置：将学习率调整为 0.02 和 0.004，步长保持 10000。

接下来是学习结果---

α=0.02：600 个测试对象中，有 583 个命中。命中率为 96.83%。

α=0.004：600 个测试对象中，有 591 个命中。命中率为 98.5%。

可以看出，学习率并不是越大越好。0.02 的学习率反而比不上 0.008 的学习率。可能是因为学习率过大，使其难以收敛。而 0.004 的学习率相比于原来的没有明显改变，说明 10000 步以内，两个学习率都能达到收敛，因此效果相差不大。

5、判别式方法的不同执行方式的讨论

和其他同学进行了讨论。我使用的梯度是每一个数据都进行调整的。而也有同学使用的是将整组数据的梯度做了平均再进行调整的。

讨论中发现两者各有优点。单步调整的，能在较小的数据量下收敛，因为其每一个数据都会进行一次调整。求平均梯度的方法则可以很好地避免某些“偏差点”对整体的影响，尤其是在数据较少的情况。

四、 比较两个模型

通过上面的分析，以及实验中的体验，发现：

1、 判别式模型的正确率普遍优于生成式模型，不过在初始设置的分布之下，几乎没有区别。而在更为复杂的数据下，差距变大。

2、 生成式模型的学习较为简单。而判别式模型的学习较为复杂，在二维正态分布的情况下，耗时也稍微长一些。不过随着维数的增加，生成式模型参数会变得更多，可能会显著影响性能。而对判别式模型影响不大。

3、 判别式模型是直接从输入得到输出，更为直接；而生成式模型则可以获取各类的特征，比较输入和各类特征的相似程度，选取其中最大的做出决策。虽然没那么直接，但是可以用于一些分类之外的功能。