

# Assignment 1 实验报告

---

## Assignment 1 实验报告

Part 1 生成三个高斯分布数据

Part2 生成模型进行分类

Generative model

Discriminative Model

Generative model 和 Discriminative Model的区别

Part3 调整数据分布后的实验结果

高维数据和低维数据

数据之间的重叠程度

代码运行方式

## Part 1 生成三个高斯分布数据

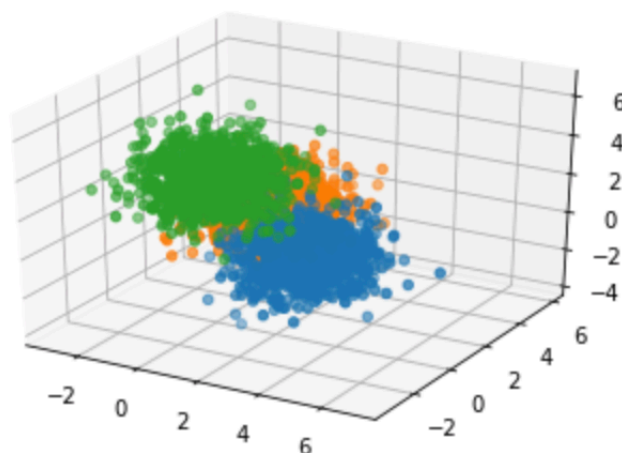
---

通过GenerateData函数生成三个三维高斯分布

$$N(\mu_1, \Sigma_1), N(\mu_2, \Sigma_2), N(\mu_3, \Sigma_3)$$
$$\mu_1 = (0, 0, 3), \mu_2 = (0, 3, 0), \mu_3 = (3, 0, 0), \Sigma_1 = \Sigma_2 = \Sigma_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

每个高斯分布取样100个点，随机sample70%数据用于训练，30%数据用于测试

生成的数据分布如下：



## Part2 生成模型进行分类

---

### Generative model

在训练集上通过train函数求得：

$$P \text{ 为先验概率, } P = (0.34857143, 0.33142857, 0.32)$$

$$\hat{\mu}_1 = (3.019, -0.007, -0.024), \hat{\mu}_2 = (-0.008, 2.968, 0.064), \hat{\mu}_3 = (-0.009, 0.0171, 2.974),$$

$$\hat{\Sigma}_1 = \begin{bmatrix} 0.927 & -0.009 & -0.12 \\ -0.009 & 0.990 & 0.105 \\ -0.012 & 0.105 & 0.994 \end{bmatrix}, \hat{\Sigma}_2 = \begin{bmatrix} 0.998 & -0.011 & -0.106 \\ -0.011 & 1.118 & -0.046 \\ -0.011 & -0.046 & 0.961 \end{bmatrix}, \hat{\Sigma}_3 = \begin{bmatrix} 1.016 & 0.032 & -0.005 \\ 0.032 & 0.943 & -0.011 \\ -0.05 & -0.01 & 0.968 \end{bmatrix}$$

在测试集上通过argmax求得节点类别

$$label(x) = \arg \max (P(x|\mu_i, \Sigma_i))$$

最终得到Acc=0.931

## Discriminative Model

$$P(y|x) = P(x|y) \frac{P(y)}{P(x)}$$

$$\hat{y} = \arg \max_y P(y|x) = \arg \max_y P(y)P(x|y)$$

$$P(t|p, \mu, \Sigma) = \prod_{n=1}^N P_c N(x_n | \mu_c, \Sigma_c), c = class(x_n)$$

设  $N_1, N_2, N_3$  为  $class1, class2, class3$  的样本数量，

$$\text{则由极大似然估计可知 } \mu_i = \frac{1}{N_i} \sum_{t_n \in C_i} x_n, \Sigma_i = \frac{1}{N_i} \sum_{t_n \in C_i} (x_n - \mu_i)(x_n - \mu_i)^T.$$

$$P(c_k|x) = \frac{p(x|c_k)P(c_k)}{\sum_j P(x|c_j)P(c_j)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)}, a_k = \log P(x|c_k)P(c_k)$$

可以使用 *softmax* 回归的方式求解，使用交叉熵作为损失函数

$\hat{y}^{(n)} = \text{softmax}(w^T x^{(n)})$  为样本  $x^{(n)}$  在每个类别下的后验概率。

$$L(w) = -\frac{1}{N} \sum_n \sum_c y_v^{(n)} \log \hat{y}_c^{(n)}$$

$$= -\frac{1}{N} \sum_n (y^{(n)})^T \log \hat{y}^{(n)}.$$

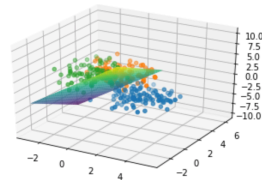
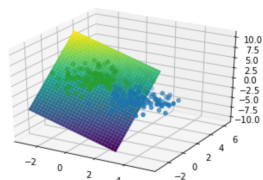
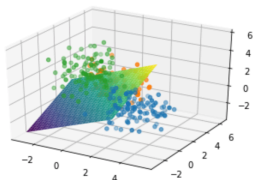
使用梯度下降，设置  $LearningRate = 0.01 \Delta w = LearningRate \times \frac{\partial L}{\partial w}$

$$w = w + \Delta w$$

$$\text{最终 } w = \begin{bmatrix} 1.21892269 & -0.6235152 & -0.65033862 & -0.02508469 \\ -0.59180617 & 1.26088011 & -0.58163856 & 0.05631258 \\ -0.62711652 & -0.6373649 & 1.23197718 & -0.03122789 \end{bmatrix}$$

可以看出较好地分辨三个类别。

通过w可以得到每一个class的投影平面，通过plot函数进行可视化，可以看出投影平面之间的区别



在测试集上通过test函数进行分类,  $\text{label}(x)=\arg \max (\text{softmax}(Wt*x))$

最终得到 $\text{Acc}=0.929$

## Generative model 和 Discriminative Model的区别

在这个实验中, 由于数据维度并不高, 并且数据之间的交叉并不明显.

生成模型准确率为93.1%, 判别模型为92.9%, 二者相差不大, 只有少数outlier会分布在的其他数据中, 造成预测错误。

生成模型对数据的分布进行预测, 是通过计算每一个class的分布后, 使用先验概率乘似然的方法得到后验概率进行分类。同时也可以生成数据。

判别式模型直接学习分类需要的线性函数。

生成式模型优点:

生成式模型有解析解, 可以直接计算出最优的参数分布。

生成式模型缺点:

生成模型预设了数据的分布类型, 进行生成式模型的构建时需要先验知识, 了解其分布, 所以如果数据的真实分布情况不是高斯分布, 或者不是简单的分布时, 生成模型的能力就会受限。

生成式模型需要的参数数量远大于判别式求解。

判别式模型优点:

参数数量少, 易于通过梯度下降或其他迭代方法求解

判别式模型缺点:

难以设置learning rate等超参数, 需要调参以达到模型的最好效果

具体的生成式模型和判别式模型的区别在part3的实验中进行分析

## Part3 调整数据分布后的实验结果

### 高维数据和低维数据

- 猜想1: 高维数据需要更多的训练样本才能得到比较好的结果

- 实验1: 构造三个50维高斯分布,  $\mu_1=(3,0,0,0,\dots,0,0,0,0,0), \mu_2=(0,3,0,0,0,\dots,0,0,0,0), \mu_3=(0,0,3,0,0,0,\dots,0,0,0), \Sigma_1 = \Sigma_2 = \Sigma_3 = \text{np.eye}(50)$ , 每个高斯分布取样100个点, 随机sample70%数据用于训练, 30%数据用于测试

结果: 生成式模型结果 $\text{Acc}=0.544$ , 判别式模型 $\text{Acc}=0.977$

分析:

1.这个数据集的前三维分布和part1中的数据分布完全一样, 但是生成式得到的结果相差很多, 说明在这个实验中数据的噪声对生成式的影响远大于对判别式模型的影响。

2.对生成式的训练集进行预测, 发现 $\text{Acc}=1$ , 远高于在测试集上的结果, 说明生成模型过拟合了训练数据, 导致在测试集中的结果变差。分析认为根本原因是因为生成式的模型参数多, 而判别式需要的模型参数相对较少, 在训练样本数量相对于需要的参数数量比较小的时候, 更多的参数更容易造成过拟合。

- 实验2: 在实验1的基础上增加取样点的个数, 每个高斯分布取样1000个点

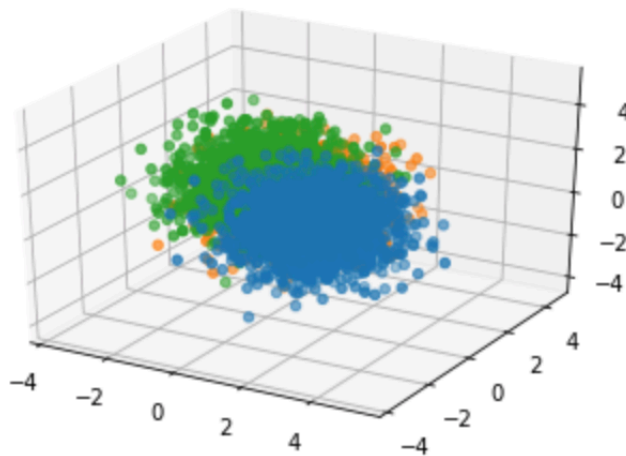
结果：生成式模型结果 $Acc=0.954$ ,判别式模型 $Acc=0.972$

分析：在训练集足够大时，生成式模型的效果也有了提升

- 结论：猜想正确
- 猜想2：高维数据训练生成式模型需要时间高于判别式训练模型需要的时间
  - 实验：构造三个50维高斯分布，每个高斯分布取样1000个点，随机sample70%数据用于训练，30%数据用于测试
  - 结果：生成式模型训练时间=0.088s，判别式模型训练时间=0.226s
  - 分析：生成式模型复杂度为  $3*N*D(\text{求}\mu)+3*N*D*D(\text{求}\Sigma)$  判别式模型训练时间= $N((D+1)*3*3*(D+1)(\text{求softmax结果})+D*3(\text{进行梯度下降}))$ ，所以虽然生成式需要的参数数量多，但是在这种模型下，判别式的计算复杂度更大。但是两者之间的比例似乎过大，可能是因为numpy内置函数对矩阵乘法的优化或者一些外部干扰原因，因为计算量并不足够大。
  - 结论：猜想错误

## 数据之间的重叠程度

- 猜想：高斯分布之间重叠度越高分类结果越差
  - 实验：构造三个3维高斯分布， $\mu_1=(1.5,0,0),\mu_2=(0,1.5,0),\mu_3=(0,0,1.5),\Sigma_1 = \Sigma_2 = \Sigma_3 = \text{np.eye}(3)$ ，每个高斯分布取样100/1000/2000个点，随机sample70%数据用于训练，30%数据用于测试
  - 结果：生成式模型结果 $Acc=0.755/0.772/0.767$ ,判别式模型 $Acc=0.711/0.767/0.765$
  - 分析：取样100个点时数据效果低于取样1000个点，因为训练数据不足，模型没能完全学习到数据的性质，取样1000个点和取样2000个点比，模型效果不好是因为数据之间的重叠和part1中数据相比较，有较多的不同类别的点互相重合，将其可视化后如下图



- 结论：猜想正确

## 代码运行方式

```
python source.py
```