University of Chinese Academy of Sciences

模式识别与机器学习

081203M04004H

Chap 6 课程作业解答

2022 年 11 月 3 号

Professor: 黄庆明

学生: 周胤昌
学号: 202228018670052
学院: 网络空间安全学院
所属专业: 网络空间安全
方向: 安全协议理论与技术

Problem 1

给定训练数据集 $\boldsymbol{X}=\begin{pmatrix}1 & 2 & 5 & 4\\ 2 & 5 & 1 & 2\end{pmatrix}, \boldsymbol{y}=\begin{pmatrix}19 & 26 & 19 & 20\end{pmatrix}^{\mathrm{T}}$, 令 $\alpha=0.001, \boldsymbol{w}_0=\begin{pmatrix}1 & 1\end{pmatrix}^{\mathrm{T}}$. 编程实现 SGD 和 GD 算法, 求解 $\boldsymbol{w}$.

Solution: 最优化问题 (即代价函数) 为

$$\min_{\boldsymbol{w}} J(\boldsymbol{w})=\frac{1}{2N}\sum_{i=1}^{N}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}^{(i)}-y^{(i)}\right)^2$$

于是批梯度下降 (BGD) 和梯度迭代更新规则分别为 (具体的 python 训练代码见如下)

$$\frac{\partial J(\boldsymbol{w})}{\partial w_j}=\frac{1}{N}\sum_{i=1}^{N}x_j^{(i)}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}^{(i)}-y^{(i)}\right),\quad w_j\leftarrow w_j-\frac{\alpha}{N}\sum_{i=1}^{N}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}^{(i)}-y^{(i)}\right)x_j^{(i)}, \alpha>0$$

```python
# 批量梯度下降 BGD
# 拟合函数为: y = theta * x
# 代价函数为: J = 1 / (2 * m) * ((theta * x) - y) * ((theta * x) - y).T;
# 梯度迭代为: theta = theta - alpha / m * (x * (theta * x - y).T);
import time
import numpy as np
# 1、多元线性回归的 BGD 程序
def bgd_multi():
    # 训练集, 每个样本有 2 个分量
    x = np.array([(1, 2), (2, 5), (5, 1), (4, 2)])
    y = np.array([19, 26, 19, 20])
    # 初始化
    m, dim = x.shape
    theta = np.ones(dim)  # 参数
    alpha = 0.001  # 学习率
    threshold = 0.0001  # 停止迭代的错误阈值
    iterations = 1500  # 迭代次数
    error = 0  # 初始错误为 0
    # 迭代开始
    for i in range(iterations):
        error = 1 / (2 * m) * np.dot((np.dot(x, theta) - y).T, (np.dot(x, theta) - y))
        # 迭代停止
        if abs(error) <= threshold:
            break
        theta -= alpha / m * (np.dot(x.T, (np.dot(x, theta) - y)))
    print('BGD 的迭代次数为%d,' % (i + 1), 'theta:', theta, ', error: %f' % error)
if __name__ == '__main__':
    start = time.time()
    bgd_multi()
    end = time.time()
    print('运行时间为: ', (end - start) * 1000, 'ms')
```

上述代码的输出结果为 $\boldsymbol{w}=\begin{pmatrix}2.868 & 4.565\end{pmatrix}^{\mathrm{T}}$, 循环迭代次数为 1500, 循环终止时的线性拟合误差为 $\epsilon=6.995$, BGD 算法运行时间为 13.84ms.

而对于 SGD 算法, 最优化问题 (即代价函数) 仍为

$$\min_{\boldsymbol{w}} J(\boldsymbol{w})=\frac{1}{2N}\sum_{i=1}^{N}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}^{(i)}-y^{(i)}\right)^2$$

于是随机梯度下降 (SGD) 和梯度迭代更新规则分别为 (具体的 python 训练代码见如下)

$$\frac{\partial J(\boldsymbol{w})}{\partial w_j} = \frac{1}{N}\sum_{i=1}^{N} x_j^{(i)}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}^{(i)} - y^{(i)}\right), \quad w_j \leftarrow w_j - \alpha\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}^{(i)} - y^{(i)}\right)x_j^{(i)}, \alpha > 0$$

```python
# 随机梯度下降 SGD
import time
import numpy as np
# 2、多元线性回归的 SGD 程序
def sgd():
    # 训练集，每个样本有 2 个分量
    x = np.array([(1, 2), (2, 5), (5, 1), (4, 2)])
    y = np.array([19, 26, 19, 20])
    # 初始化
    m, dim = x.shape
    theta = np.ones(dim)  # 参数
    alpha = 0.001  # 学习率
    threshold = 0.0001  # 停止迭代的错误阈值
    iterations = 1500  # 迭代次数
    error = 0  # 初始错误为 0
    # 迭代开始
    for i in range(iterations):
        error = 1 / (2 * m) * np.dot((np.dot(x, theta) - y).T, (np.dot(x,
        ↪ theta) - y))
        # 迭代停止
        if abs(error) <= threshold:
            break
        j = np.random.randint(0, m)
        theta -= alpha * (x[j] * (np.dot(x[j], theta) - y[j]))
    print('SGD 的迭代次数为%d,' % (i + 1), 'theta:', theta, ', error: %f' %
    ↪ error)
if __name__ == '__main__':
    start = time.time()
    sgd()
    end = time.time()
    print('运行时间为: ', (end - start) * 1000, 'ms')
```

上述代码的输出结果为 $\boldsymbol{w} = \begin{pmatrix}2.880 & 4.613\end{pmatrix}^{\mathrm{T}}$, 循环迭代次数为 1500, 循环终止时的线性拟合误差为 $\epsilon = 7.01$, SGD 算法的运行时间为 13.51ms. 可以看出 SGD 算法的速度是略快于 BGD 算法的.

Problem 2

利用下面表格1中的训练数据训练朴素贝叶斯分类器. 给定测试样本 $\boldsymbol{x} = (2, S)^{\mathrm{T}}$ 和 $\boldsymbol{x} = (1, N)^{\mathrm{T}}$, 请预测他们的标签.

表 1: Problem 2 的训练数据

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3
x_2	S	M	M	S	S	S	M	M	L	L	L	M	M	L	L
y	-1	-1	1	1	-1	-1	-1	1	1	1	1	1	1	1	-1

由于测试样本中出现了训练样本中未出现的特征分量, 所以需要对朴素贝叶斯分类器参数的极大似然估计做 *Laplace* 平滑 (其中平滑系数 λ 选取为 1):

$$p(y=1) = \frac{\sum_{i=1}^{N} I_{\{y^{(i)}=1\}} + 1}{N + K} = \frac{9+1}{15+2} = \frac{10}{17} \Rightarrow p(y=-1) = 1 - \frac{10}{17} = \frac{7}{17}$$

第 1 个特征分量的似然函数为

$$p(x_1 = 1|y=1) = \frac{\sum_{i=1}^{N} I_{\{x_1^{(i)}=1 \wedge y^{(i)}=1\}} + 1}{\sum_{i=1}^{N} I_{\{y^{(i)}=1\}} + S_1} = \frac{2+1}{9+3} = \frac{1}{4},$$

$$p(x_1 = 2|y=1) = \frac{\sum_{i=1}^{N} I_{\{x_1^{(i)}=2 \wedge y^{(i)}=1\}} + 1}{\sum_{i=1}^{N} I_{\{y^{(i)}=1\}} + S_1} = \frac{3+1}{9+3} = \frac{1}{3},$$

$$p(x_1 = 3|y=1) = \frac{\sum_{i=1}^{N} I_{\{x_1^{(i)}=3 \wedge y^{(i)}=1\}} + 1}{\sum_{i=1}^{N} I_{\{y^{(i)}=1\}} + S_1} = \frac{4+1}{9+3} = \frac{5}{12}$$

第 2 个特征分量的似然函数为:

$$p(x_2 = S|y=1) = \frac{\sum_{i=1}^{N} I_{\{x_2^{(i)}=S \wedge y^{(i)}=1\}} + 1}{\sum_{i=1}^{N} I_{\{y^{(i)}=1\}} + S_2} = \frac{1+1}{9+4} = \frac{2}{13},$$

$$p(x_2 = M|y=1) = \frac{\sum_{i=1}^{N} I_{\{x_2^{(i)}=M \wedge y^{(i)}=1\}} + 1}{\sum_{i=1}^{N} I_{\{y^{(i)}=1\}} + S_2} = \frac{4+1}{9+4} = \frac{5}{13},$$

$$p\left(x_2=L|y=1\right)=\frac{\sum_{i=1}^{N} I_{\left\{x_2^{(i)}=L\wedge y^{(i)}=1\right\}}+1}{\sum_{i=1}^{N} I_{\left\{y^{(i)}=1\right\}}+S_2}=\frac{4+1}{9+4}=\frac{5}{13},$$

$$p\left(x_2=N|y=1\right)=\frac{\sum_{i=1}^{N} I_{\left\{x_2^{(i)}=N\wedge y^{(i)}=1\right\}}+1}{\sum_{i=1}^{N} I_{\left\{y^{(i)}=1\right\}}+S_2}=\frac{0+1}{9+4}=\frac{1}{13}$$

预测表达式为

$$p\left(y=1|\boldsymbol{x}\right)=\frac{p\left(\boldsymbol{x}|y=1\right)p\left(y=1\right)}{p\left(\boldsymbol{x}\right)}=\frac{\prod_{j=1}^{D} p\left(x_j|y=1\right)p\left(y=1\right)}{\prod_{j=1}^{D} p\left(x_j|y=1\right)p\left(y=1\right)+\prod_{j=1}^{D} p\left(x_j|y=-1\right)p\left(y=-1\right)}$$

于是对于样本 $\boldsymbol{x}^{(1)}=(2,S)^{\mathrm{T}}$, 预测概率为

$$p\left(y=1|\boldsymbol{x}^{(1)}\right)=\frac{\frac{4}{12}\times\frac{2}{13}\times\frac{10}{17}}{\frac{4}{12}\times\frac{2}{13}\times\frac{10}{17}+\frac{2+1}{6+3}\times\frac{3+1}{6+4}\times\frac{7}{17}}=0.35461$$

$$p\left(y=-1|\boldsymbol{x}^{(1)}\right)=\frac{\frac{2+1}{6+3}\times\frac{3+1}{6+4}\times\frac{7}{17}}{\frac{4}{12}\times\frac{2}{13}\times\frac{10}{17}+\frac{2+1}{6+3}\times\frac{3+1}{6+4}\times\frac{7}{17}}=0.64539$$

所以样本 $\boldsymbol{x}^{(1)}=(2,S)^{\mathrm{T}}$ 的标签预测为 $y=-1$.

而对于样本 $\boldsymbol{x}^{(2)}=(1,N)^{\mathrm{T}}$, 预测概率为

$$p\left(y=1|\boldsymbol{x}^{(2)}\right)=\frac{\frac{2+1}{9+3}\times\frac{0+1}{9+4}\times\frac{10}{17}}{\frac{2+1}{9+3}\times\frac{0+1}{9+4}\times\frac{10}{17}+\frac{3+1}{6+3}\times\frac{0+1}{6+4}\times\frac{7}{17}}=0.382003$$

$$p\left(y=-1|\boldsymbol{x}^{(2)}\right)=\frac{\frac{3+1}{6+3}\times\frac{0+1}{6+4}\times\frac{7}{17}}{\frac{2+1}{9+3}\times\frac{0+1}{9+4}\times\frac{10}{17}+\frac{3+1}{6+3}\times\frac{0+1}{6+4}\times\frac{7}{17}}=0.617997$$

所以样本 $\boldsymbol{x}^{(2)}=(1,N)^{\mathrm{T}}$ 的标签预测为 $y=-1$. 上述算法可以用 Python 代码实现, 不妨将 S,M,L,N 分别记为 $1,2,3,4$, 具体代码如下:

```python
import numpy as np
from sklearn.naive_bayes import MultinomialNB
X = np.array([[1, 1], [1, 2], [1, 2], [1, 1], [1, 1], [2, 1], [2, 2], [2, 2],
              [2, 3], [2, 3], [3, 3], [3, 2], [3, 2], [3, 3], [3, 3]])
# S,M,L,N 分别用数字 1,2,3,4 替代
Y = np.array([-1, -1, 1, 1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, -1])
clf = MultinomialNB(alpha = 1.0, fit_prior = True, class_prior = None)
clf.fit(X, Y)
clf.predict_proba([[2, 1], [1, 4]]) # 输出 (2,S) 和 (1,N) 划分到各个类别的概率值
```

经过上述代码可得: 对于样本 $\boldsymbol{x}^{(1)}=(2,S)^{\mathrm{T}}$, 预测概率为 $p\left(y=1|\boldsymbol{x}^{(1)}\right)=0.406$, $p\left(y=-1|\boldsymbol{x}^{(1)}\right)=0.594$, 因此标签为 $y=-1$; 对于样本 $\boldsymbol{x}^{(2)}=(1,N)^{\mathrm{T}}$, 预测概率为 $p\left(y=1|\boldsymbol{x}^{(2)}\right)=0.384$, $p\left(y=-1|\boldsymbol{x}^{(2)}\right)=0.616$, 故标签为 $y=-1$. 且经过 MultinomialNB 后所习得的 (平滑) 先验概率的对数为 $-0.511,-0.916$.

上述算法也可以用 Matlab 代码实现, 具体代码见下所示:

```
n=input('请输入训练集的个数:n\n');
k=2; A=zeros(n,3);% 存储训练集, 用来学习
A(:,1)=input('请输入所有样本的第一个特征 (用列向量表示): \n');
A(:,2)=input('请输入所有样本的第二个特征 (用列向量表示): \n');
A(:,3)=input('请输入所有样本所属的类 (用列向量表示): \n');
x1=input('请输入需要预测的样本的第一个特征: 1 或 2 或 3 \n');
x2=input('请输入需要预测的样本的第二个特征: S 或 M 或 L 或 N\n','s');
% 计算其属于 1 类的概率
F1=zeros(1,3);
for i=1:n
    if A(i,3)==1
        F1(1,1)=F1(1,1)+1;
        if A(i,1)==x1
            F1(1,2)=F1(1,2)+1;
        end
        if A(i,2)==x2
            F1(1,3)=F1(1,3)+1;
        end
    end
end
C1=(F1(1,1)+1)/(n+k)*(F1(1,2)+1)/(F1(1,1)+3)*(F1(1,3)+1)/(F1(1,1)+4);
% 属于 2 类的概率
F2=zeros(1,3);
for i=1:n
    if A(i,3)==-1
        F2(1,1)=F2(1,1)+1;
        if A(i,1)==x1
            F2(1,2)=F2(1,2)+1;
        end
        if A(i,2)==x2
            F2(1,3)=F2(1,3)+1;
        end
    end
end
C2=(F2(1,1)+1)/(n+k)*(F2(1,2)+1)/(F2(1,1)+3)*(F2(1,3)+1)/(F2(1,1)+4);
if C1>C2
    disp('该样本的类为 1');
else if C1==C2
        disp('该样本的类为-1 或 1')
    else
        disp('该样本的类为-1')
    end
end
```

在命令行窗口的输出过程为:

```
>> MultinomialNB
请输入训练集的个数:n
15
请输入所有样本的第一个特征 (用列向量表示):
[1,1,1,1,1,2,2,2,2,2,3,3,3,3,3]
请输入所有样本的第二个特征(用列向量表示):
['S','M','M','S','S','S','M','M','L','L','L','M','M','L','L']
请输入所有样本所属的类(用列向量表示):
[-1,-1,1,1,-1,-1,-1,1,1,1,1,1,1,1,-1]
请输入需要预测的样本的第一个特征: 1 或 2 或 3
1
请输入需要预测的样本的第二个特征: S 或 M 或 L 或 N
N
该样本的类为-1
```

```
>> MultinomialNB
请输入训练集的个数:n
15
请输入所有样本的第一个特征 (用列向量表示):
[1,1,1,1,1,2,2,2,2,2,3,3,3,3,3]
请输入所有样本的第二个特征(用列向量表示):
['S','M','M','S','S','S','M','M','L','L','L','M','M','L','L']
请输入所有样本所属的类(用列向量表示):
[-1,-1,1,1,-1,-1,-1,1,1,1,1,1,1,1,-1]
请输入需要预测的样本的第一个特征: 1 或 2 或 3
2
请输入需要预测的样本的第二个特征: S 或 M 或 L 或 N
S
该样本的类为-1
```

通过 Matlab 右侧的变量值栏目可以计算出: 测试样本 $\boldsymbol{x}^{(1)} = (2, S)^{\mathrm{T}}$ 的预测概率为 $p\left(y = 1|\boldsymbol{x}^{(1)}\right) = 0.3546$, $p\left(y = -1|\boldsymbol{x}^{(1)}\right) = 0.6454$, 因此标签为 $y = -1$. $\boldsymbol{x}^{(2)} = (1, N)^{\mathrm{T}}$ 的预测概率为 $p\left(y = 1|\boldsymbol{x}^{(2)}\right) = 0.3820$, $p\left(y = -1|\boldsymbol{x}^{(1)}\right) = 0.6180$, 因此标签为 $y = -1$.

Problem 3

生成式判别模型: 高斯贝叶斯分类器和逻辑回归

Part A

考虑一类特定的高斯朴素贝叶斯分类器, 其中

- Y 是服从伯努利分布的布尔变量, 其中参数 $\pi = P(Y = 1), P(Y = 0) = 1 - \pi$;
- $X = [x_1, \cdots, x_D]^{\mathrm{T}}$, 其中特征分量 x_i 是连续的随机变量. 对于每个 x_i, $P(x_i|Y = k)$ 是高斯分布 $\mathcal{N}(\mu_{ik}, \sigma_i)$. 注意到 σ_i 是高斯分布的标准差 (且不依赖于 k);
- 给定 Y 时, $\forall i \neq j, x_i$ 和 x_j 都是条件独立的 (因此称之为朴素分类器).

问题: 请证明判别式分类器 (如逻辑回归) 与上述特定类别的高斯朴素贝叶斯分类器之间的关系正是逻辑回归所使用的形式.

Solution: 先利用贝叶斯公式:

$$\begin{aligned} P\left(Y = 1|X\right) &= \frac{P\left(X|Y = 1\right) \cdot P\left(Y = 1\right)}{P\left(X|Y = 0\right) \cdot P\left(Y = 0\right) + P\left(X|Y = 1\right) \cdot P\left(Y = 1\right)} \\ &= \frac{1}{1 + \dfrac{P\left(X|Y = 0\right) \cdot P\left(Y = 0\right)}{P\left(X|Y = 1\right) \cdot P\left(Y = 1\right)}} \\ &= \frac{1}{1 + \exp\left(\ln \dfrac{P\left(X|Y = 0\right) \cdot P\left(Y = 0\right)}{P\left(X|Y = 1\right) \cdot P\left(Y = 1\right)}\right)} \end{aligned}$$

再代入 $P(Y = 1) = \pi, P(Y = 0) = 1 - \pi$, 同时将 ln 中的乘法变为加法:

$$P\left(Y = 1|X\right) = \frac{1}{1 + \exp\left(\ln \dfrac{P\left(X|Y = 0\right)}{P\left(X|Y = 1\right)} + \ln \dfrac{P\left(Y = 0\right)}{P\left(Y = 1\right)}\right)} = \frac{1}{1 + \exp\left(\ln \dfrac{P\left(X|Y = 0\right)}{P\left(X|Y = 1\right)} + \ln \dfrac{1 - \pi}{\pi}\right)}$$

因为 X 是 D 维的, 同时每一个特征都是相互条件独立的, 因此 $P\left(X|Y\right) = \prod_{i=1}^{D} P\left(x_i|Y\right)$, 于是有

$$P\left(Y = 1|X\right) = \frac{1}{1 + \exp\left(\ln \dfrac{\prod_{i=1}^{D} P\left(x_i|Y = 0\right)}{\prod_{i=1}^{D} P\left(x_i|Y = 1\right)} + \ln \dfrac{1 - \pi}{\pi}\right)}$$

由于 $\forall x_i$, $P(x_i|Y = k)$ 都服从高斯分布 $\mathcal{N}(\mu_{ik}, \sigma_i)$, 即 $P\left(x_i|Y = k\right) = \dfrac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\dfrac{(x_i - \mu_{ik})^2}{2\sigma_i^2}\right)$. 我

们先看 $\ln\dfrac{\prod_{i=1}^{D} P\left(x_i|Y=0\right)}{\prod_{i=1}^{D} P\left(x_i|Y=1\right)}$:

$$\begin{aligned}
\ln\frac{\prod_{i=1}^{D} P\left(x_i|Y=0\right)}{\prod_{i=1}^{D} P\left(x_i|Y=1\right)} &= \ln\prod_{i=1}^{D}\frac{1}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{(x_i-\mu_{i0})^2}{2\sigma_i^2}\right) - \ln\prod_{i=1}^{D}\frac{1}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{(x_i-\mu_{i1})^2}{2\sigma_i^2}\right) \\
&= \sum_{i=1}^{D}\left(\ln\frac{1}{\sqrt{2\pi}\sigma_i} - \frac{(x_i-\mu_{i0})^2}{2\sigma_i^2}\right) - \sum_{i=1}^{D}\left(\ln\frac{1}{\sqrt{2\pi}\sigma_i} - \frac{(x_i-\mu_{i1})^2}{2\sigma_i^2}\right) \\
&= \sum_{i=1}^{D}\left(\frac{(x_i-\mu_{i1})^2}{2\sigma_i^2} - \frac{(x_i-\mu_{i0})^2}{2\sigma_i^2}\right) = \sum_{i=1}^{D}\left(\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}x_i + \frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2}\right)
\end{aligned}$$

再将其代入到 $P(Y=1|X)$ 的表达式中:

$$\begin{aligned}
P\left(Y=1|X\right) &= \frac{1}{1+\exp\left\{\sum_{i=1}^{D}\left(\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}x_i + \frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2}\right) + \ln\frac{1-\pi}{\pi}\right\}} \\
&= \frac{1}{1+\exp\left\{-\left[\sum_{i=1}^{D}\underbrace{\frac{\mu_{i1}-\mu_{i0}}{\sigma_i^2}}_{\text{即}w_i}x_i + \underbrace{\left(\sum_{i=1}^{D}\frac{\mu_{i0}^2-\mu_{i1}^2}{2\sigma_i^2} + \ln\frac{\pi}{1-\pi}\right)}_{\text{即}b}\right]\right\}} \\
&= \frac{1}{1+\exp\left[-\left(\sum_{i=1}^{D} w_i x_i + b\right)\right]}
\end{aligned}$$

同理可得:

$$P\left(Y=0|X=1\right) = 1 - P\left(Y=1|X\right) = \frac{\exp\left[-\left(\sum_{i=1}^{D} w_i x_i + b\right)\right]}{1+\exp\left[-\left(\sum_{i=1}^{D} w_i x_i + b\right)\right]}$$

也就是说, 上述特定类别的高斯朴素贝叶斯分类器其实正好就是逻辑回归的形式! 逻辑回归的参数 w 通常采用 SGD 算法去学习, 而而高斯朴素贝叶斯分类器根据假设直接给出了这些参数, 这也说明了逻辑回归是一种更强的模型, 因为它的假设很弱, 而朴素贝叶斯的假设非常的强.

Part B

一般的高斯朴素贝叶斯分类器和逻辑回归: 将“$P(x_i|Y=k)$ 的标准差 σ_i 不依赖于 k”的假设删除掉. 即 $\forall x_i, P(x_i|Y=k)$ 是高斯分布 $\mathcal{N}(\mu_{ik}, \sigma_{ik})$, 其中 $i = 1, \cdots, D$ 且 $k = 0, 1$.

问题: 更一般的高斯朴素贝叶斯分类器中 $P(Y|X)$ 的表达式是否仍然形如逻辑回归的形式. 写出 $P(Y|X)$ 的新格式来证明你的答案.

Solution: 其实就是去掉高斯分布中的方差与其所属类别无关的假设, 即 $P(x_i|Y=k)$ 服从高斯分布 $\mathcal{N}(\mu_{ik}, \sigma_{ik})$. 前面的推导与上面一致, 即

$$P(Y=1|X) = \frac{1}{1+\exp\left(\ln\dfrac{\prod_{i=1}^{D} P(x_i|Y=0)}{\prod_{i=1}^{D} P(x_i|Y=1)} + \ln\dfrac{1-\pi}{\pi}\right)}$$

由于高斯分布假设更改, 即 $P(x_i|Y=k) = \dfrac{1}{\sqrt{2\pi}\sigma_{ik}}\exp\left(-\dfrac{(x_i-\mu_{ik})^2}{2\sigma_{ik}^2}\right)$, 将其代入 $\ln\dfrac{\prod_{i=1}^{D} P(x_i|Y=0)}{\prod_{i=1}^{D} P(x_i|Y=1)}$ 得到:

$$\begin{aligned}
\ln\frac{\prod_{i=1}^{D} P(x_i|Y=0)}{\prod_{i=1}^{D} P(x_i|Y=1)} &= \ln\prod_{i=1}^{D}\frac{1}{\sqrt{2\pi}\sigma_{i0}}\exp\left(-\frac{(x_i-\mu_{i0})^2}{2\sigma_{i0}^2}\right) - \ln\prod_{i=1}^{D}\frac{1}{\sqrt{2\pi}\sigma_{i1}}\exp\left(-\frac{(x_i-\mu_{i1})^2}{2\sigma_{i1}^2}\right) \\
&= \sum_{i=1}^{D}\left[\ln\frac{1}{\sqrt{2\pi}\sigma_{i0}} - \frac{(x_i-\mu_{i0})^2}{2\sigma_{i0}^2}\right] - \sum_{i=1}^{D}\left[\ln\frac{1}{\sqrt{2\pi}\sigma_{i1}} - \frac{(x_i-\mu_{i1})^2}{2\sigma_{i1}^2}\right] \\
&= \sum_{i=1}^{D}\left[\left(\frac{1}{2\sigma_{i1}^2} - \frac{1}{2\sigma_{i0}^2}\right)x_i^2 + \left(\frac{\mu_{i0}}{\sigma_{i0}^2} - \frac{\mu_{i1}}{\sigma_{i1}^2}\right)x_i + \left(\frac{\mu_{i1}^2}{2\sigma_{i1}^2} - \frac{\mu_{i0}^2}{2\sigma_{i0}^2}\right) + \ln\frac{\sigma_{i1}}{\sigma_{i0}}\right]
\end{aligned}$$

将上式结果代入 $P(Y=1|X)$ 得到:

$$P(Y=1|X) = \frac{1}{1+\exp\left(\sum_{i=1}^{D}\left[\underbrace{\left(\frac{1}{2\sigma_{i1}^2} - \frac{1}{2\sigma_{i0}^2}\right)}_{\text{二次项非零}}x_i^2 + \left(\frac{\mu_{i0}}{\sigma_{i0}^2} - \frac{\mu_{i1}}{\sigma_{i1}^2}\right)x_i + \left(\frac{\mu_{i1}^2}{2\sigma_{i1}^2} - \frac{\mu_{i0}^2}{2\sigma_{i0}^2}\right) + \ln\frac{\sigma_{i1}}{\sigma_{i0}}\right] + \ln\frac{1-\pi}{\pi}\right)}$$

显然上述形式中的二次项系数非零, 即去掉了高斯分布中方差与类别无关的假设后朴素贝叶斯变成了非线性的方式. 但是逻辑回归中没有二次项, **所以这种假设条件下的高斯朴素贝叶斯分类器无法直接用 Logistic 回归表示**.

Part C

高斯贝叶斯分类器和逻辑回归: 现在考虑下述假设中的高斯贝叶斯分类器 (不带有“朴素”):

- Y 是服从伯努利分布的布尔变量, 其中参数 $\pi = P(Y = 1), P(Y = 0) = 1 - \pi$;
- $X = [x_1, x_2]^{\mathrm{T}}$, 即我们只考虑样本的特征维数是 2 的情况, 并且每个特征分量都是一个连续的随机变量. 给定 y 时, x_1, x_2 不再是条件独立的了. 我们假设 $P(x_1, x_2|Y = k)$ 是双变量高斯分布 $\mathcal{N}(\mu_{1k}, \mu_{2k}, \sigma_1, \sigma_2, \rho)$, 其中 μ_{1k}, μ_{2k} 分别是 x_1, x_2 的均值, σ_1, σ_2 分别是 x_1, x_2 的标准差, 并且 ρ 是 x_1, x_2 的相关系数. 因此双变量高斯分布的密度函数为

$$P\left(x_1, x_2|Y = k\right) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}}\exp\left\{\frac{1}{1-\rho^2}\left[-\frac{\left(x_1-\mu_{1k}\right)^2}{2\sigma_1^2} + \frac{\rho}{\sigma_1\sigma_2}\left(x_1-\mu_{1k}\right)\left(x_2-\mu_{2k}\right) - \frac{\left(x_2-\mu_{2k}\right)^2}{2\sigma_2^2}\right]\right\}$$

问题: 不朴素的高斯贝叶斯分类器中 $P(Y|X)$ 的表达式是否仍然形如逻辑回归的形式. 写出 $P(Y|X)$ 的新格式来证明你的答案.

Solution: 由于 $X = [x_1, x_2]^{\mathrm{T}}$, 直接使用 **Part A** 中推导的公式

$$P\left(Y = 1|X\right) = \frac{1}{1 + \exp\left(\ln\frac{P\left(X|Y=0\right)}{P\left(X|Y=1\right)} + \ln\frac{1-\pi}{\pi}\right)}$$

将题中的二元正态分布的密度函数表达式代入 $P(Y = 1|X)$, 此时我们不妨先看 $\ln P\left(X|Y = k\right)$:

$$\ln P\left(X|Y = k\right) = \ln\frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} + \frac{1}{1-\rho^2}\left[-\frac{\left(x_1-\mu_{1k}\right)^2}{2\sigma_1^2} + \frac{\rho}{\sigma_1\sigma_2}\left(x_1-\mu_{1k}\right)\left(x_2-\mu_{2k}\right) - \frac{\left(x_2-\mu_{2k}\right)^2}{2\sigma_2^2}\right]$$

于是经过**精心整理**可得:

$$\begin{aligned}
\ln\frac{P\left(X|Y=0\right)}{P\left(X|Y=1\right)} &= \ln P\left(X|Y=0\right) - \ln P\left(X|Y=1\right) \\
&= \frac{1}{1-\rho^2}\left\{\left[\frac{\mu_{10}-\mu_{11}}{\sigma_1^2} + \frac{\rho\left(\mu_{21}-\mu_{20}\right)}{\sigma_1\sigma_2}\right]x_1 + \left[\frac{\mu_{20}-\mu_{21}}{\sigma_2^2} + \frac{\rho\left(\mu_{11}-\mu_{10}\right)}{\sigma_1\sigma_2}\right]x_2\right. \\
&\quad \left. + \left[\frac{\mu_{11}^2-\mu_{10}^2}{2\sigma_1^2} + \frac{\mu_{21}^2-\mu_{20}^2}{2\sigma_2^2} + \frac{\rho\left(\mu_{10}\mu_{20}-\mu_{11}\mu_{21}\right)}{\sigma_1\sigma_2}\right]\right\}
\end{aligned}$$

将上述结果代入到 $P(Y = 1|X)$, 可将 $P(Y = 1|X)$ 表示为逻辑回归的形式:

$$P\left(Y = 1|X\right) = \frac{1}{1 + \exp\left[-\left(\sum_{i=1}^{2} w_i x_i + b\right)\right]}$$

其中

$$\begin{aligned}
w_1 &= \frac{-1}{1-\rho^2}\left[\frac{\mu_{10}-\mu_{11}}{\sigma_1^2} + \frac{\rho\left(\mu_{21}-\mu_{20}\right)}{\sigma_1\sigma_2}\right] \\
w_2 &= \frac{-1}{1-\rho^2}\left[\frac{\mu_{20}-\mu_{21}}{\sigma_2^2} + \frac{\rho\left(\mu_{11}-\mu_{10}\right)}{\sigma_1\sigma_2}\right] \\
b &= \frac{-1}{1-\rho^2}\left[\frac{\mu_{11}^2-\mu_{10}^2}{2\sigma_1^2} + \frac{\mu_{21}^2-\mu_{20}^2}{2\sigma_2^2} + \frac{\rho\left(\mu_{10}\mu_{20}-\mu_{11}\mu_{21}\right)}{\sigma_1\sigma_2}\right] + \ln\frac{1-\pi}{\pi}
\end{aligned}$$

因此可以知道**去掉“naive”的高斯贝叶斯分类器依然是线性的, 可以用 Logistic 回归表示**.

通过上述推导, 分别去掉了两个不同的假设来揭开高斯朴素贝叶斯分类器的面纱, 最后简单总结为:

- **传统的高斯朴素贝叶斯等价于通过假设直接从训练数据中算出参数的逻辑回归;**
- **去掉高斯分布中方差与类别无关的假设后, 高斯朴素贝叶斯将变成非线性分类器, 其中多了二次项特征;**
- **移除特征之间的条件独立性假设后, 高斯贝叶斯依然是线性的, 但是其联合概率密度计算会比较复杂.**

Problem 4

Logistic 回归 (LR) 的 MLE 参数估计: 如果有参数的正则项 (用以抑制过拟合), 那么该如何估计参数? 其中

$$l(\boldsymbol{w}) = \log L(\boldsymbol{w}) - \lambda \|\boldsymbol{w}\|_2^2 = \sum_{i=1}^{N} y^{(i)} \log f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right) + \left(1 - y^{(i)}\right) \log\left(1 - f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right)\right) - \lambda \|\boldsymbol{w}\|_2^2$$

Solution: 易知

$$l(\boldsymbol{w}) = \sum_{i=1}^{N} y^{(i)} \log f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right) + \left(1 - y^{(i)}\right) \log\left(1 - f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right)\right) - \lambda \boldsymbol{w}^{\mathrm{T}} \boldsymbol{w}$$

故可以如下求得梯度:

$$\frac{\partial l(\boldsymbol{w})}{\partial w_j} = \sum_{i=1}^{N} \left[y^{(i)} \frac{1}{f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right)} \cdot \frac{\partial f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right)}{\partial w_j} + \left(1 - y^{(i)}\right) \frac{1}{1 - f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right)} \cdot \frac{-\partial\left(f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right)\right)}{\partial w_j} \right] - 2\lambda w_j$$

又因为

$$\frac{\partial f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right)}{\partial w_j} = \underbrace{\frac{\partial g\left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}^{(i)}\right)}{\partial w_j} = g\left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}^{(i)}\right)\left(1 - g\left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}^{(i)}\right)\right) \boldsymbol{x}_j^{(i)}}_{g(z)=\frac{1}{1+\exp(-z)} \Rightarrow g'(z)=g(z)(1-g(z))} = f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right)\left(1 - f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right)\right) \boldsymbol{x}_j^{(i)}$$

于是得到梯度表达式:

$$\begin{aligned}
\frac{\partial l(\boldsymbol{w})}{\partial w_j} &= \sum_{i=1}^{N} \left\{ y^{(i)} \cdot \left[\left(1 - f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right)\right) \boldsymbol{x}_j^{(i)} \right] + \left(1 - y^{(i)}\right)(-1) \cdot \left(f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right) \boldsymbol{x}_j^{(i)} \right) \right\} - 2\lambda w_j \\
&= \sum_{i=1}^{N} \left[y^{(i)} \cdot \left(1 - f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right)\right) \cdot \boldsymbol{x}_j^{(i)} + \left(y^{(i)} - 1\right) \cdot f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right) \cdot \boldsymbol{x}_j^{(i)} \right] - 2\lambda w_j \\
&= \sum_{i=1}^{N} \left[y^{(i)} - f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right) \right] \boldsymbol{x}_j^{(i)} - 2\lambda w_j
\end{aligned}$$

为了得到 MLE 的参数估计, 我们需要使用 SGD 算法来得到 $\boldsymbol{w}$, 因此 SGD 的梯度更新规则如下:

$$w_j \leftarrow w_j + \alpha \left\{ \left[y^{(i)} - f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right) \right] \boldsymbol{x}_j^{(i)} - 2\lambda w_j \right\} = (1 - 2\lambda\alpha) w_j + \alpha \left[y^{(i)} - f\left(\boldsymbol{x}^{(i)}, \boldsymbol{w}\right) \right] \boldsymbol{x}_j^{(i)}$$

其中 α 为步长; λ 为超参数, 需要手动调整来观察过拟合的抑制情况.