Pro. 多元统计分析
指导教师: 李野
仓库地址: https://github.com/DGMEFG/anime_cls

基于 GochiUsa_Faces 数据集分类问题的解决方案

沈运之
052110814
1729990469@qq.com

黄开奕
182111510
1476060622@qq.com

徐行
082120109
1928161381@qq.com

December 22, 2023

Keywords: 图像分类 降维 判别 多因变变量线性回归 假设检验

1. 介绍

1.1. 概要

统计学习中，分类问题应该算得上是一个相当经典的模型，大多数方法都可以参与这一问题的解决，基于此，用分类问题来应用多元统计分析所学到的知识再合适不过。

分类问题中，图像分类占据了很大程度的一部分，然后，现实中的图片分类问题要经过传感器获取，以及 Jpeg 压缩一系列退化的过程，其一般受噪声影响较为严重，所以我们选择了产生于互联网上的图片，即动漫人物的图片构建我们的分类问题 (其实单纯是因为兴趣)。

该图片数据集主要由两个文件夹构成，ANIME 文件夹用于训练，DANBOORU 文件夹用于测试，其中包含 9 个类别，分别是 Blue Mountain，Chino，Chiya，Cocoa，Maya，Megumi，Mocha，Rize，Sharo 对应数字 0-8；ANIME 包含 59579 张图片，DANBOORU 包含 9141 张图片，初始文件夹里包含 (通道数为 3) 从 26×26，到 987×987 尺寸不一的图片，为了便于处理，已经经过 python 脚本统一处理为 32×32。原数据集来源于 Kaggle:`https://www.kaggle.com/datasets/rignak/gochiusa-faces`。

1.2. 解决方案

首先我们小组成员自行充当分类器，分类效果非常好，因此这个学习问题是理论上可以实现。下面我将阐述这份实验提供的解决方案：

Note:

- 首先观察图片数据的特征是否近似满足正态分布，以及初步构建对于数据认识。
- 然后基于先验，选择合适的方法进行降维，并将降至二维进行可视化。
- 对于不同的降维结果，使用基于模型的多因变量的线性回归，SVM，以及 model-free 的基于决策树的分类器进行测试，挑选出最好的结果。

- 基于以上结果进行分析。

1.3. 符号约定

为了便于叙述，这里规定 N 为数据集样本数，M 为每个样本的特征，这里定义每个样本的特征为图片张量向量化的结果，X 为 $N \times M$ 的数据矩阵，Y 为 $N \times 1$ 的标签向量，其中 $y_i \in Z$ and $y_i \in [0, 8]$，约定每一个样本为 $X_i^\top = \begin{bmatrix} x_{i1} & \cdots & x_{iM} \end{bmatrix}$，对应标签为 y_i，$Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_N \end{bmatrix}^\top$ 从而有：

$$X = \begin{bmatrix} X_1^\top \\ X_2^\top \\ \vdots \\ X_N^\top \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1M} \\ x_{21} & x_{22} & \cdots & x_{2M} \\ \vdots & & & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NM} \end{bmatrix}$$

2. 数据属性

在对数据进行进一步分析，为了尽可能防止出现数值问题 (0-255 以内的数字多次线性组合可能会是很大的值)，首先先将数据通过标准化处理调整为均值为 0，方差为 1，设 $\bar{x}_i$ 为数据矩阵 X 第 i 列的样本均值 (也就是随机变量 X_i 的 N 次取样)，σ_i 为其标准差，于是其内的数据 x 的标准化后的值 $\tilde{x}$ 为:

$$\tilde{x} = \frac{x - \bar{x}_i}{\sigma_i}$$

2.1. 类别情况

首先观察最直观的数据属性，将每个类别在训练集和测试集上的规模画出 (见 Figure 1)，训练集内最少的两个类别为 Mocha 与 Blue Mountain 分别有 1241 个，1607 个，而数量最多的类别 Chino 有 12941 个，倍数达到十倍，该数据集为长尾数据集，原数据集作者说，大部分角色具有明显的特征，因此我们仍然选择这两个类别作为我们分类任务的一环 (本质上还是因为这个学习问题不太难)。

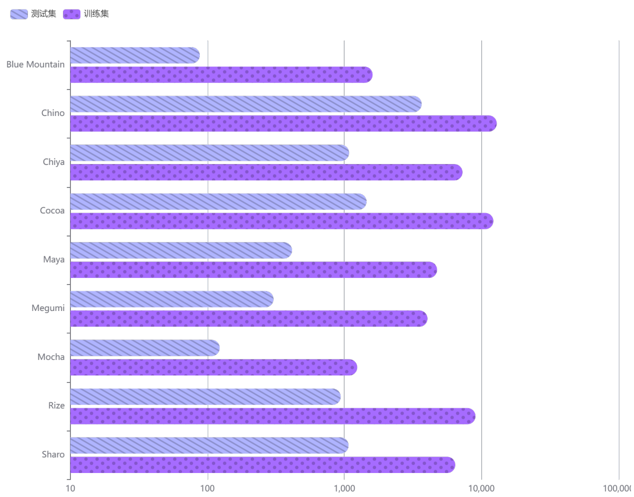

Figure 1: 各类别训练集与测试集的分类情况，横轴为类别对应的样本数目，且采用对数刻度

2.2. 特征相关性

由于下面要使用线性回归模型，需要先保证数据特征不存在强相关，否则严重的多重共线性将导致线性模型 $C^\top C$ 不满秩，使得线性回归将不存在唯一解，这可能会影响答案的准确性。注意到样本的特征数为 $M = 3072$，设样本协方差阵为 S，$V^{1/2} = \text{diag}(\sqrt{S_{11}}, \sqrt{S_{22}}, \cdots, \sqrt{S_{MM}})$，相关系数矩阵 R 由以下公式给出:

$$R = (V^{1/2})^{-1} S (V^{1/2})^{-1} b$$

实际计算复杂度为 $N \times M^2$，实际运行却很快，这可能得归功于 numpy 的矩乘优化，统计总计 3072×3072 个相关系数，绘制其频率 (已经划分好分段区间) 直方图 (参考 Figure 2)

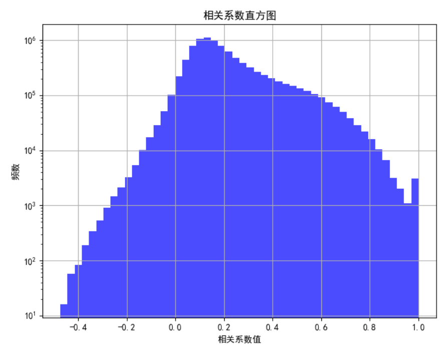

Figure 2: 频率直方图，已经将纵坐标处理为 log 尺度

计算得到有多达 136496 对特征具有 ≥ 0.7 的相关

性 (情理之中)，之后的进一步工作可以尝试使用降维方法减少多重共线性，那时再做进一步分析。

3. 降维

现在，我们已经对数据 (训练集) 有了先验认知，发现数据集存在以下几个问题：

Observación 1

1. 样本特征维度过大，这无论是对于存储与速度都提出了极大的要求。
2. 特征之间存在不可忽视的相关性。

使用这样的数据来训练传统模型是不可行的，由于本身该分类问题属于不太难的学习问题，只是用少量特征来学习大概率可行，接下来考虑使用降维方法对数据进行简化。

3.1. 分类器介绍

3.1.1. 多因变量线性回归

基于 softmax 回归的思想，回归问题可以通过拟合在每个类别出现的概率 (9×1 向量) 再使用 softmax 函数转化为一个分类问题，因此这里可以将标签向量 Y 的所有分量 y_i 处理成独热编码的形式。

设有 k(降维后特征的个数) 个自变量，$p(p=9)$ 个因变量，降维后数据矩阵 X^* 为 $N\times k$ 的矩阵，其中每一行代表一个样本的特征，这里新引入因变量矩阵 Y^*，其具有如下形式:

$$Y^* = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1p} \\ y_{21} & y_{22} & \cdots & y_{2p} \\ \vdots & & & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{Np} \end{bmatrix}$$

其中 $\begin{bmatrix} y_{i1} & \cdots & y_{ip} \end{bmatrix}^\top$ 是初始标签向量 Y 的分量 y_i 产生的独热编码。

因此学习问题转化为模型:

$$\begin{cases} Y^* = (\mathbf{1}_N|X^*)\beta + E = C\beta + E \\ \epsilon_{(i)} \sim N_p(0,\Sigma), \epsilon_i \perp \epsilon_j (i\neq j) \end{cases}$$

其中 $E^\top = \begin{bmatrix} \epsilon_{(1)} & \cdots & \epsilon_{(N)} \end{bmatrix}$，$\beta : (k+1)\times p$，$\beta$ 的最小二乘估计为:

Teorema 1

$$\hat{\beta} = (C^\top C)^{-1} C^\top Y$$

最终的模型就是线性回归模型输出向量中分量最大值对应的下标为预测类别。

3.2. 判别分析

判别分析也可以解决新样品的归属问题，因此可以使用判别分析解决分类问题，判别分析中我们考虑广义平方距离判别法以及贝叶斯判别法，判别分析首先假设样本属于正态分布，这需要我们检验数据是否近似满足 9 元正态分布。

一个简单的方式是，当原假设 $H_0 : X \sim N_M(\mu,\Sigma)$ 成立时，数据总体 X 具有如下性质：

Lema 1

$$D^2 = (X-\mu)^\top \Sigma^{-1}(X-\mu) \sim \chi^2(M)$$

其中 μ 使用 $\bar{X}$ 估计，Σ 使用 S 估计，因此，可以考虑绘制 χ^2 统计量的 Q-Q 图，设 $D^2_{(t)}$ 为排序后第 t 个样本的马氏距离，以及 χ^2_t 为 $\chi^2(M)$ 对应的分位数，通过观察 $(D^2_{(t)}, \chi^2_t)(t=1,\cdots,N)$ 散点图是否近似分布在斜率为 1 的直线上来检验其正态性

然而，我们运气并不好，首先数据本身特征达到 3072 个，其次之前的相关系数矩阵已经告诉我们有 136496 个特征对具有强相关性，因此原数据的协方差阵极大概率是半正定，而并非正定，从而 σ 无法取逆。

考虑一种退而求其次的方式，我们使用主成分分析降维之后再检验其正态性，如果降维后仍然不是正

态分布，完全可以拒绝原假设，此时也就意味着无法使用贝叶斯判别法 (要求得到后验分布)，但是仍然可以测试一下广义平方距离判别法的性能。

在正式讲述 PCA 模块之前，我们先在此先展示一下这些样本降维后关于卡方分布的 Q-Q 图 (参考 Figure-3)，很显然，这说明我们预计的 $(X-\mu)^\top\Sigma^{-1}(X-\mu)$ 与 $\chi^2(M)$ 有着不小的差距，因此可以正式将贝叶斯判别法从我们的考虑方案去除。

3.2.1. 广义平方距离判别

在这个具体问题下，由于总体并非正态分布，无法检验样本总体之间的均值是否具有显著性差异，但是可以作为一种基线方法给出 acc 的下界。

具体来说，广义平方距离判别法通过计算新样品 X 到 9 个总体 $G_i(i=0,\cdots,8)$ 的广义平方距离：

$$D^2(X,G_t)=d_t^2(X)+\ln|S_t|-2\ln|q_t|$$

其中 q_t 对应于训练集的先验即，事实上当各个总体服从正态分布时，广义平方距离判别与贝叶斯判别一致，但是广义平方距离判别法适用范围更广。

3.2.2. 其它分类器

我们还考虑了 SVM，决策树传统学习模型，旨在通过多种分类器的测试，检验各种降维方法对于不同模型的具体效果，同时丰富我们的解决方案。

3.3. 指标介绍

3.4. PCA 降维

最简单的无监督降维是 PCA 方法，PCA 将原始特征降维至几个正交的主成分，避免了多重共线性的隐忧，同时极大地保留了原始特征的信息。同时得到方差解释比例：

Lema 2

$$\text{ratio}=\frac{\sum_{i=1}\lambda_k}{\sum_{i=1}\lambda_M}$$

为了搜索到一个合适的主成分个数，我们取了一个离散集合 $\{10,50,100,200,500\}$，绘制了 Figure 3 来展示这种变化的趋势，由 $50,100,200,500$ 自变量以 2 倍变化，因变量等差变化，可知实际上因变量实际上大致以 $\log_2$ 的趋势变化，在计算机计算速度 (尤其是 SVM 运行有点慢了) 与保留主成分个数的权衡后，我们选择使用 100 个主成分继续实验。

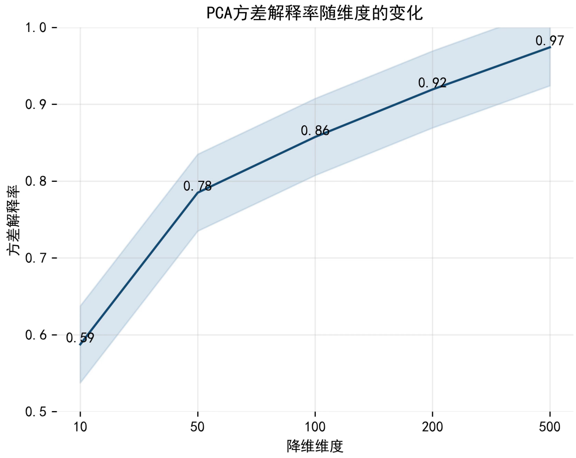

Figure 3: PCA 解释方差比

3.5. FLDA 降维

有监督降维引入了对于标签的考虑，其大致工作流程是计算类内散度矩阵 A 与类间散度矩阵 B，使用 $A^{-1}B$ 的特征向量作为投影向量对原数据进行降维，其中由于矩阵 B 秩最多为 C - 1 (C 为类别数)。不同于 PCA 简单地仅仅使用训练集的特征，FLDA 还使用了训练集的标签作为先验，有助于模型更全面的考虑这个学习问题。

虽然将数据维度从 3072 维降至 $\min(N,C-1)$ 维，似乎丢失了大量信息，但是其对于目标任务有针对的降维，使得其往往呈现比 pca 更好的效果，同时原始数据已经偏离正态分布，判别分析基于正态性假设而建立，FLDA 表现最佳时应是各类别近似是正态总体，通过这样看似大胆的降维也许更有助于发现一些有趣的事情。

3.6. 神经网络降维

一般来说神经网络是一个 encoder-decoder 架构，encoder 层扮演着特征提取的角色，因此在进行前两个

实验中，以及之前上课有所思考，大概感觉其实这个 encoder 层完全可以认为是一个降维层。

如果只是用一层全连接层来作特征提取，那么本质上也是一个线性降维，但是，注意，神经网络权重参数的调整还受到来自标签所导致的损失函数的影响，所以比起像 PCA ，其更像 FLDA ，而且由于其不拘泥于数据是否存在闭式解，以及激活函数带来的非线性性质，其效果应该比 FLDA 更好。

为了证实我们的想法，我们用 torch 实现了两层神经网络，第一层隐藏层有 100 个神经元，这与 PCA 一致，第二层将 100 个神经元映射到 9 个神经元输出 logits，除此之外，更加详细的训练配置参数如下所示：

Name	Info
批次大小	256
优化器	Adam
学习率	0.001
损失函数	多元交叉熵

Definición 1 (Nombre de la definición)
Sea f ...

Definición 2
Definición sin nombre ...

Ejemplo para hacer referencia a una definición (teorema, corolario, etc), en la definición 2.

Notación 1
Notación sin nombre ...

Proof. Prueba de teorema □

Teorema 2
Teorema sin nombre ...

En el teorema ??

Ejemplo 1 (Nombre del ejemplo)
Sea f ...

Ejemplo 2
Ejemplo sin nombre ...

Corolario 1 Sea f ...

Corolario 2 Corolario sin nombre ...

Lema 3 (Nombre del lema) Sea f ...

Lema 4 Lema si nombre ...

Note: (Nombre de la nota) Sea f ...

Note: Nota sin nombre ...

Vocabulario 1 (Nombre del vocabulario) Sea f ...

Vocabulario 2 Vocabulario sin nombre ...

Algoritmo 1 (Nombre del algoritmo)
Algoritmo con nombre ...

(Nombre de la caja)
Sea f ...

Scaja sin nombre ...

Note cómo en la Figura 4 ...

En la Figura 5, en la subfigura 5b se observa que ...

Figure 4: Título de la figura. Decir si es elaboración propia o poner referencia.

(a) Subfigura 1

(b) Subfigura 2

Figure 5: Título para la figura en general. Decir si es elaboración propia o poner referencia.

Table 1: Título de la Tabla. Decir si es elaboración propia o poner referencia.

name	*foo*			
Models	A	B	C	D
Model X	X1	X2	X3	X4
Model Y	Y1	Y2	Y3	Y4

Puede observar en la Tabla 1 ...

Ecuación numerada:

$$x = \frac{b \pm \sqrt{b^2 - 4ac}}{2a} \tag{1}$$

En la fórmula 1 ...

Ecuación no numerada:

$$x = \frac{b \pm \sqrt{b^2 - 4ac}}{2a}$$

Ecuación alineada numerada:

$$\begin{aligned} x &= a^2 - b^2 \qquad (2) \\ &= (a-b)(a+b) \qquad (3) \end{aligned}$$

En las expresiones 2 y 3 ...

Ecuación alineada no numerada:

$$\begin{aligned} x &= a^2 - b^2 \\ &= (a-b)(a+b) \end{aligned}$$

Ecuación centrada

$$a^2 = b^2 + c^2$$

Ejemplo de código Java:

```
/**
* This is a doc comment.
*/
package
  com.ociweb.jnb.lombok;

import java.util.Date;
import lombok.Data;
import
  lombok.EqualsAndHashCode;
import lombok.NonNull;

@Data
@EqualsAndHashCode(exclude={"address","
public class Person {
        enum Gender { Male,
           Female }

        // another comment

        @NonNull private
           String firstName;
        @NonNull private
           String lastName;
        @NonNull private
           final Gender
           gender;
        @NonNull private
           final Date
           dateOfBirth;
```

```
            private String ssn;
            private String
               address;
            private String city;
            private String state;
            private String zip;
    }
```

Este es código en la misma línea `import java.util.Date;`, el símbolo | es sólo un delimitador y se puede cambiar por algún otro que no se utilice en el código.

Esta es una cita de la bibliografía: [1]

La bibliografía se prefiere según APA con utilizando biblatex con Biber, también aceptamos el formato IEEE.

4. Bibliography

[1] Cita H

A. Apéndice

Apéndice