



## 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 多因变量线性回归 假设检验 SVM 决策树 KNN

包含 9141 张图片，初始文件夹里包含 (通道数为 3) 从  $26 \times 26$ , 到  $987 \times 987$  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  $32 \times 32$ 。原数据集来源于 Kaggle:<https://www.kaggle.com/datasets/rignak/gochiusa-faces>。

## 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

### 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  $X_i^\top = [x_{i1} \ \dots \ x_{iM}]$ , 对应标签为  $y_i$ ,  $Y = [y_1 \ y_2 \ \dots \ y_N]^\top$  从而有:

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

## 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),  $\sigma_i$  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), 训练集中最少的两个类别为 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)。

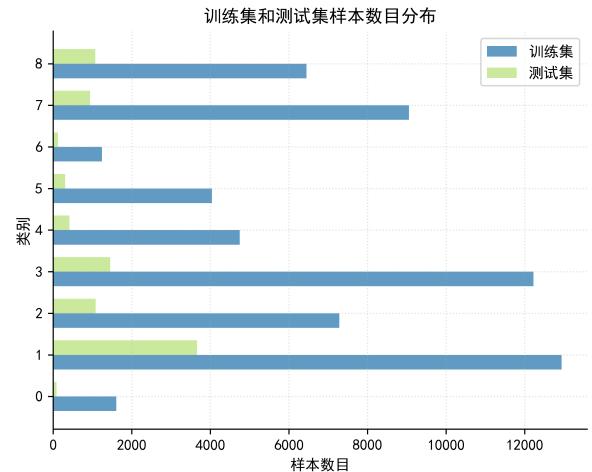


Figure 1: 各类别训练集与测试集的分类情况, 横轴为类别对应的样本数目, 且采用对数刻度

### 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  $S$ ,  $V^{1/2} = \text{diag}(\sqrt{S_{11}}, \sqrt{S_{22}}, \dots, \sqrt{S_{MM}})$ , 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 的矩阵优化, 统计总计  $3072 \times 3072$  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)

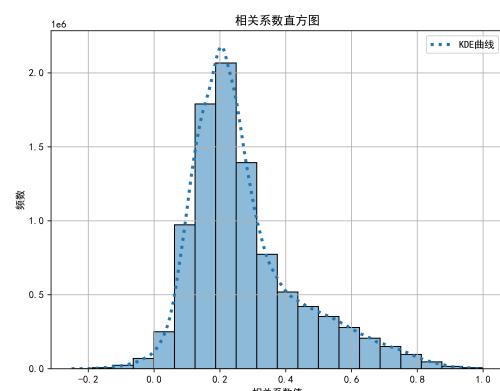


Figure 2: 频率直方图

计算得到有多达 280930 对特征具有  $\geq 0.7$  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 ( $9 \times 1$  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  $\begin{bmatrix} y_{11} & \cdots & y_{1p} \end{bmatrix}^\top$  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  $E^\top = [\epsilon_{(1)} \ \cdots \ \epsilon_{(N)}]$ ,  $\beta : (k+1) \times p$ ,  $\beta$  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 [2] 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:

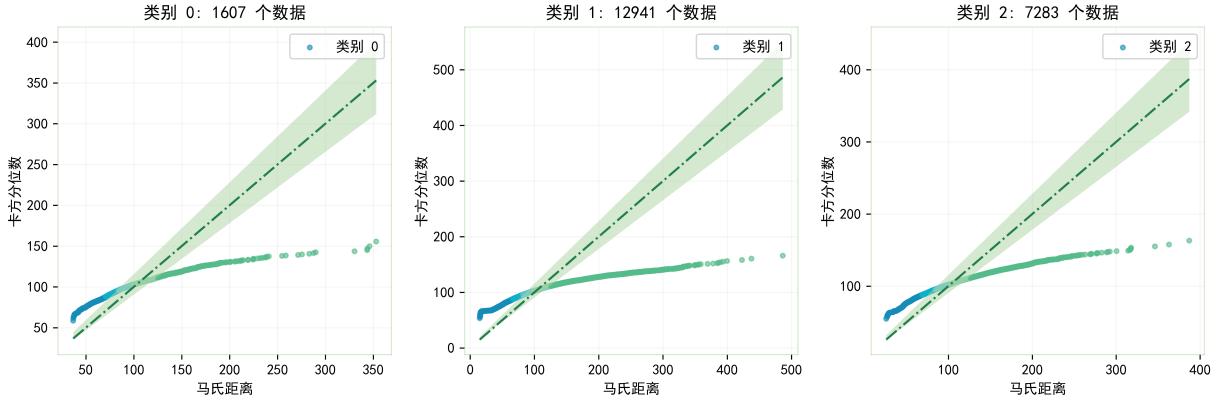
#### Lema 1

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

其中  $\mu$  使用  $\bar{X}$  估计,  $\Sigma$  使用  $S$  估计, 因此, 可以考虑绘制  $\chi^2$  统计量的 Q-Q 图, 设  $D_{(t)}^2$  为排序后第  $t$  个样本的马氏距离, 以及  $\chi_t^2$  为  $\chi^2(M)$  对应的分位数, 通过观察  $(D_{(t)}^2, \chi_t^2) (t=1, \dots, N)$  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  $\sigma$  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



**Figure 3:** 取前 3 个类别的分布结果作展示，可以看到 (马氏距离, 卡方分布分位数) 的点对并没有落在斜率为 1 的过远点的直线上，因此有理由拒绝原假设，即个总体不服从正态分布。

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  $(X - \mu)^T \Sigma^{-1} (X - \mu)$  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  $G_i (i = 0, \dots, 8)$  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训练集中各个类别出现的频率， $S_t$  指的是第  $t$  个总体的样本协方差阵，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 KNN(邻居数 = 5)，SVM(传统学习方法中分类问题必不可少的登场角色)[4] 和决策树 (model-free 的分类器的代表) 等传统学习模型，目标是通过对多种分类器进行测试，检验不同降维方法在各个模型上的具体效果，以进一步丰富我们的解决方案。通过这样的综合考虑，我们可以更全面地了解不同模型在应对特定问题时的优劣势，并为我们的解决方案提供更深入、可靠的评估。这种多模型测试的方法有助于我们更全面地把握问题的复杂性，同时为最终选择最合适的模型和降维策略提供了有力的支持。

## 3.3. 指标介绍

由于是多分类问题，并且数据集具有长尾性质，这就要求需要有多样的分类器评价指标来支撑实验的有效性，实验采用了以下指标

- Accuracy: 最常见的分类器性能评判指标
- Micro-F1(等同于 Accuracy): 对于数据集中每个类单独计算  $TP_i, FP_i, FN_i$ ，同时定义  $pre_{mi}$  以及  $rec_{mi}$ ，从而得到 Micro-F1:

$$\begin{aligned} pre_{mi} &= \frac{\sum_{i=1}^p TP_i}{\sum_{i=1}^p TP_i + \sum_{i=1}^p FP_i} \\ rec_{mi} &= \frac{\sum_{i=1}^p TP_i}{\sum_{i=1}^p TP_i + \sum_{i=1}^p FN_i} \\ F1_{mi} &= \frac{2 \times pre_{mi} \times rec_{mi}}{pre_{mi} + rec_{mi}} \end{aligned}$$

该指标会受到数量较多的类别影响较大。

- Macro-F1: 不同于 Micro-F1, 定义  $\text{pre}_{ma}$  为各个类别精度 (precision) 的算术平均值,  $\text{rec}_{ma}$  为各个类别召回率 (recall) 的算术平均值, 得到 Macro-F1 为:

$$\text{F1}_{ma} = \frac{2 \times \text{pre}_{ma} \times \text{rec}_{ma}}{\text{pre}_{ma} + \text{rec}_{ma}}$$

- Geometric-acc: 所有类别 accuracy 的算术平均值
- Harmonic-acc: 所有类别 accuracy 的调和平均值, 由于调和平均对于较小的值有很大的惩罚, 因此该项指标可以适于检测长尾学习里分类器的性能
- Lowest Recall: 所有类别召回率的最小值

### 3.4. PCA 降维

最简单的无监督降维是 PCA 方法 [3], 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:  $\text{ratio} = \frac{\sum_{i=1}^k \lambda_k}{\sum_{i=1}^M \lambda_M}$

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 4 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。

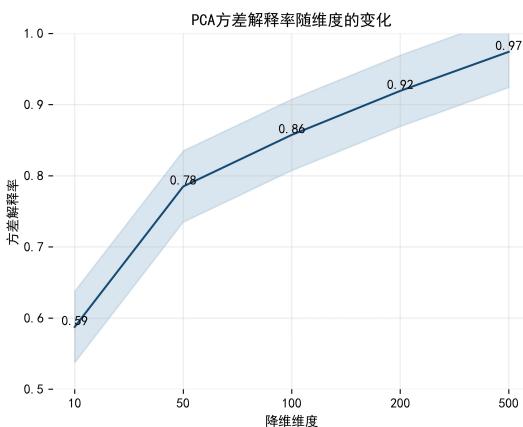


Figure 4: PCA 解释方差比

### 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性的降维, 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。

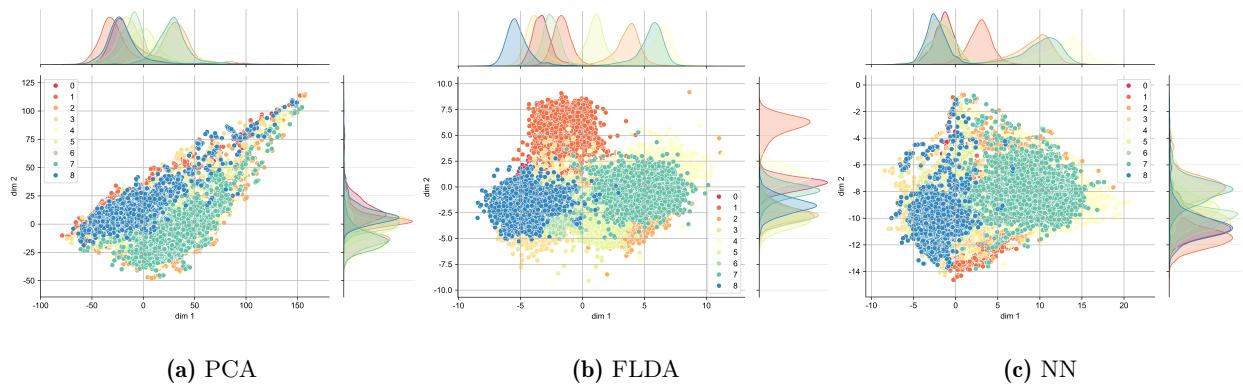
### 3.6. 神经网络降维

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
激活函数	Relu



**Figure 5:** 降维效果展示，其中 NN 的第一层降维至 2 维

### 3.7. 初步降维结果分析

首先为了很好地直观展示三者降维的效果，我们将原始数据降至 2 维，绘制它们的散点图及 KDE 曲 [5](参考 Figure5)。

对于 PCA，能够看出类 7 和类 8 峰值区域，具有明显的差异，意味着 PCA 能够很好的分开两个类，事实上类 7(Rize) 与类 8(Sharo) 在训练集有 9052, 6445 条数据，Rize 是紫发紫瞳女生，Sharo 是金发碧眼女生，从颜色上就可以有很好地区分；类 8(Sharo) 与类 0(Blue mountain) 的头发颜色都属于偏黄的类型，大概率前 2 个主成分只提取了整体特征，一些眼睛颜色以及更细节的特征（脸形应该不算在内，二次元人物脸形都是类似的），同样深紫色的 Rize(类 7)，棕色的 Chiya(类 2)，深蓝色的 Maya(类 4)，颜色差异性也不是很大，因此在 PCA 的 KDE 曲线中这 3 个类别的峰值区域几乎重合。可以得出结论，PCA 最先感知到的是图像的整体特征。

对于 FLDA，尽管将 3072 维降至 2 维，FLDA 以其极具针对性的目标优化，使得其无论在第一个降维特征，还是第二个降维特征下，每个类别的峰值区域有了较大的差异，或许使用一个不经 learning 的普通学习器也能发挥很好的分类性能。这也暗示了，FLDA 具有很强的针对性压缩功能（但是这里无法看出泛化性能，因此还需更深入的工作），大概率可以估计 FLDA + KNN 实际生活中应用非常广泛。

神经网络降维有点大失所望，也许是梯度下降只能搜到局部最优，也许是其层与层之间还有着一些联系，导致各类别 KDE 密度曲线的峰值区域大部分重合。有可能是全连接层输出维度较低，使得其难以

构建准确的像素特征，我们似乎很难解释它究竟干了什么。但是，我们仍可以将其纳入接下来任务考虑的一环，毕竟我们并不需要数据降维至 2 维，100 维即可满足大多数情况下的需求，因此接下来，我们会采用隐层 100 个神经元的神经网络构建分类器。

### 3.8. 降维方法与分类器

## 4. 比较分析

接下来，我们将具体观察三种降维度方法对于分类模型的效果对比，首先观察 Table 1。

## 4.1. 降维方法

FLDA 降维是使用的所有降维方法中最好的，适用于方案中所有分类器，不仅如此，缓解了长尾数据集带来的影响，这可以从 Least Recall 以及 Harmonic-acc 看得很明显，如果一旦某个类别的分类准确度较低，那么 Harmonic-acc 将大幅减少：

### Note:

$$\frac{2 \times 0.5 \times 0.5}{0.5 + 0.5} = 0.5 \quad \frac{2 \times 0.1 \times 0.9}{0.1 + 0.9} = 0.18$$

即 Harmonic-acc 对较小的 acc 给予更大的惩罚。

然后 PCA 整体表现一般，无论在哪个分类器上都落后于同为线性降维的 FLDA，但是 PC 维度上选择

Method	Accuracy	Macro-F1	Least Recall	Geometric-acc	Harmonic-acc
决策树 + PCA	0.640	0.432	0.102	0.445	0.305
KNN + PCA	0.760	0.560	0.136	0.551	0.362
SVM + PCA	0.875	0.753	0.352	0.742	0.660
QDA + PCA	0.859	0.676	0.024	0.674	0.164
多因变量线性回归 + PCA	0.788	0.548	0.000	0.537	0.000
决策树 +FLDA	0.791	0.645	0.364	0.673	0.613
KNN + FLDA	0.880	0.747	0.420	0.758	0.700
SVM + FLDA	0.877	0.750	0.409	0.756	0.696
QDA + FLDA	0.866	0.736	0.409	0.775	<b>0.730</b>
多因变量线性回归 + FLDA	0.823	0.637	0.081	0.611	0.295
决策树 +NN	0.827	0.690	<b>0.455</b>	0.719	0.684
KNN+ NN	0.871	0.756	0.409	0.766	0.711
SVM + NN	<b>0.888</b>	<b>0.778</b>	0.341	<b>0.780</b>	0.699
QDA + NN	0.678	0.525	0.267	0.584	0.468
多因变量线性回归 + NN	0.824	0.603	0.000	0.595	0.000

Table 1: 效果比对图

具有灵活性，因此可以尝试对于不同降维维度做进一步分析。

神经网络特征层降维对于各种分类器的性能有所提升，与 PCA 有类似之处，性能好于 PCA，与 FLDA 效果差不多。可见这种奇怪的降维方式在维度足够的情况下，的确能加入对于标签的考虑（基于损失函数的梯度优化）。另一种神经网络降维方式时自编码器（Auto-Encoder），这次实验最大的遗憾或许就在此，在实验的最后阶段，我们才得知 AutoEncoder(AE)[1] 的存在，想要用 AE 来丰富降维工具，但是事实上 AE 的架构某种视角下其实是 PCA 的非线性版本，也许更能丰富我们对于无监督降维的理解。

## 4.2. 分类器

综合 Figure 7 与 Table 1，下文 QDA 即广义平法距离法，我们可以有一下分析：

**决策树 KNN QDA** 可以看到，当 PCA 主成分个数在 {10, 50, 100, 200, 500} 内变动，这些不需要参数进行学习的方法几乎不受影响，而且变化没有规律性。相比之下，FLDA 提取到了更有针对性的特征，这些非参数模型的效果都上升了 10 个百分点；KNN 在神经网络降维里也表现得很优秀。

KNN, QDA 在该问题展现了非常好的性能，最终的方案里将考虑这两个分类器。

**多因变量线性回归** 由于测试集和训练集类别 0 和类别 6 出现次数都较少（可以从其混淆矩阵观察得出），加之维数较低时 PCA 降维无法提供有效的特征，其分类性能较差，随着选择主成分数量增多，性能有所提升。

同时，注意到多因变量线性回归在神经网络降维上表现最优，事实上神经网络线性层的工作就是将 100 维度的数据映射至 9 维，这本质上就是一个多因变量线性回归任务，线性层的工作与现在的模型的工

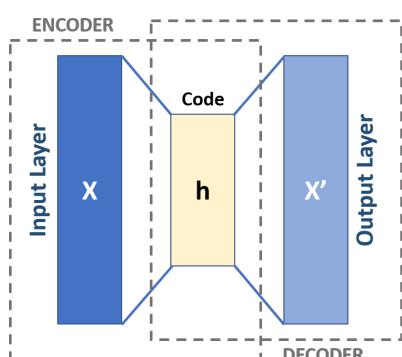
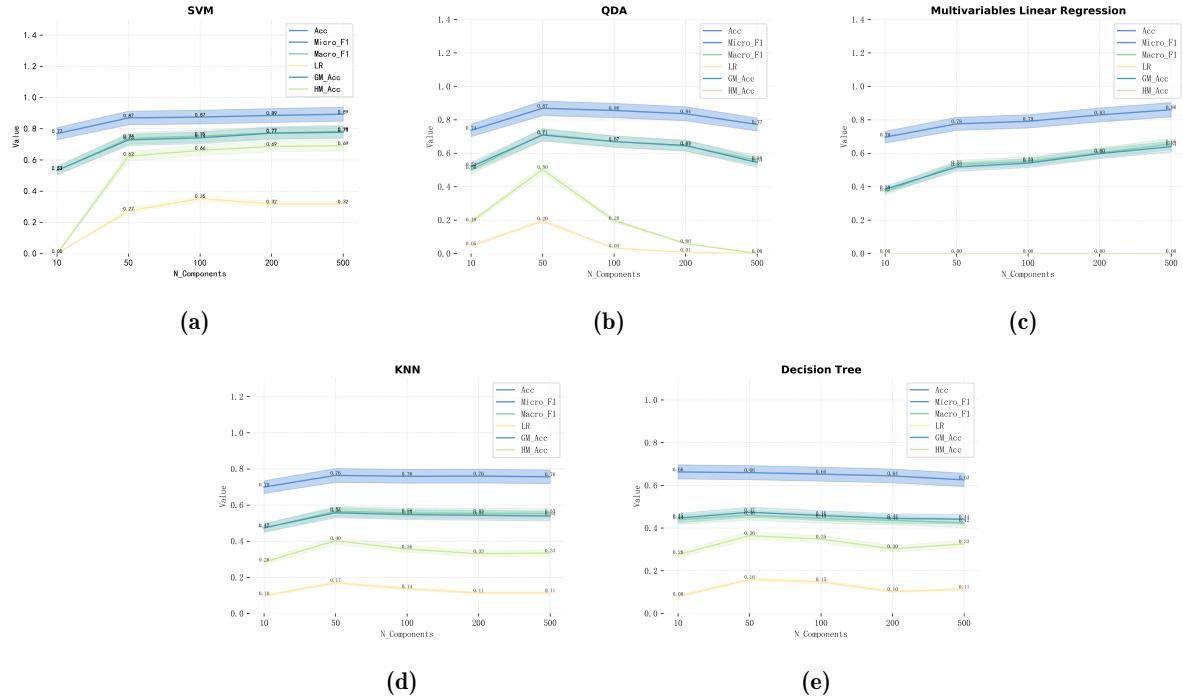
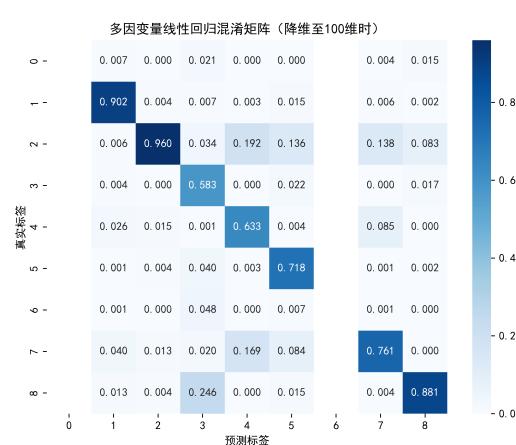


Figure 6: AutoEncoder



**Figure 7:** 对 PCA 的 5 个具有代表的主成分个数值设置实验，上面 5 张图代表了解决方案中的 5 个分类器，每张图，当主成分个数变化时，分类器的各项性能变化的折线图。



**Figure 8:** 主成分个数 =100 时多因变量线性回归的混淆矩阵，注意类别 0, 6 出现缺失

作相契合，所以使用多因变量的线性回归充当 NN 的分类器效果较好，这与之前的预想相吻合。

**SVM** 作为传统学习分类模型的最强者，当主成分个数达到 50，Accuracy 就已经达到 86.9%；主成分个数为 100 时，达到了 87.5%，已然是该问题最好的分类器之一，这符合预期，并提供了一个较好的上界。

## 5. 总结

经过以上工作的筛选，剩下值得考虑的 5 个方案，分别是 KNN + FLDA, SVM + FLDA, QDA + FLDA, KNN + NN, SVM + NN，这其中 NN 方案旨在启发思考，不过由于我们的模型文件似乎只有 1mb 左右（见 api/Trained/base.pt），实际部署也可以考虑，但是优先度肯定没有 FLDA 降维方法高。KNN + FLDA 与 QDA(广义平法距离判别法) + FLDA 都是部署起来较简单的方法，因此它们时最后的该分类问题的解决方案。

本次实验，还尝试利于神经网络 Decoder-Encoder 架构的 Decoder 降维，效果不错，不过可以考虑之后我们小组学习研究一下自编码器降维。

在进行本实验前，我们采用过 ResNet18 进行过图像分类，在测试集上达到了 92% 的 Accuracy，最后是简单的模型以及 SVM(理应是) 胜出，违反预期的是多因变量的线性回归，本以为它类似神经网络的线性层，可以和普通的两层隐层网络的效果一致（测试集 Accuracy 达到 90%），也许是标准 softmax 才能实现这种效果。

## 6. 参考

- [1] Umberto Michelucci, An Introduction to Autoencoders, CoRR, vol. abs/2201.03898, 2022, <https://arxiv.org/abs/2201.03898>.
- [2] 高惠璇. 应用多元统计分析. 北京大学出版社, 2005.
- [3] Svante Wold, Kim Esbensen, Paul Geladi. Principal component analysis. Chemometrics and Intelligent Laboratory Systems, 2(1-3): 37-52, 1987. Elsevier.
- [4] Marti A. Hearst, Susan T. Dumais, Edgar Osuna, John Platt, Bernhard Scholkopf. Support vector machines. IEEE Intelligent Systems and Their Applications, 13(4): 18-28, 1998. IEEE.
- [5] Khosrow Dehnad, Density estimation for statistics and data analysis, Taylor & Francis, 1987.