# MNIST上的无监督学习实验

> **课程：** 人工智能与机器学习基础 Lab2
>
> **作者：** TA：杨睿卿
>
> **日期：** 2025年10月



# 0. 实验简介

本实验旨在帮助学生深入理解无监督学习的核心概念和实际应用。实验聚焦于两类重要技术：

1. **降维（Dimension Reduction）**：使用PCA,t-SNE以及AutoEncoder两种方法将高维图像数据投影到低维空间
2. **聚类（Clustering）**：使用高斯混合模型（GMM）对降维后的数据进行聚类分析

通过对MNIST手写数字数据集的实验，我们将直观地观察到：

- 不同降维方法的效果和特点
- EM算法在GMM中的应用
- 聚类结果与真实标签的对比分析

# 0.1 数据集介绍

**MNIST** 是机器学习领域最经典的图像分类数据集之一：

- **数据量**：60,000个样本
- **图像格式**：28×28像素的灰度图像
- **类别**：10个类别（数字0-9）
- **特点**：数据规模适中，适合教学实验

在本实验中，每个28×28的灰度图像会被展平为784维的向量，然后通过降维方法压缩到更低维的特征空间（如100维），最后在低维空间中应用GMM进行聚类。

# 0.2 文件组织结构

```
├── code/                    # 实验核心代码
│   ├── arguments.py         # 命令行参数定义
│   ├── dataloader.py        # 数据加载器（MNIST）
│   ├── submission.py        # 核心模型实现（PCA、GMM）
│   ├── autoencoder.py       # 自编码器模型
│   ├── train.py             # 主训练脚本
│   ├── visualization.py     # 可视化工具
│   └── util.py              # 工具函数
├── requirements.txt         # 依赖包列表
└── MNIST.md                 # 实验流程讲解文件
```

`submission.py`是你最终需要提交的，助教会检查其中的代码正确性。如果并非必须，请尽量尽量不要修改其他文件，如果需要修改，请写一个README文件进行说明。在评测中，助教会使用你的模型和权重，在测试数据上进行批量化脚本测试

# 1. 实验流程

# 1.1 环境准备

首先配置虚拟环境和安装必要的依赖包：

```
cd [$你文件所在目录]
pip install -r requirements.txt
```

## 1.2 数据预处理（10%）

我们在dataloader.py中调用在 `submission.py` 中 `data_preprocess()` 函数将28×28的图像转换为两种格式：

- `image1D`：展平为784维向量，用于PCA降维
- `image2D`：保持2D形状 (1, 28, 28)，用于AutoEncoder

你也可以尝试其他数据处理操作来优化后续的数据处理部分。

# 2. 核心任务实现

## 2.1 PCA降维实现（10%）

在 `submission.py` 中已提供PCA框架，需完成 `fit` 方法以实现主成分分析。

**任务要求**

完成 `PCA` 类的 `fit` 方法，该方法需要：

- 保留 `d` 个主成分（保存在 `self.components_` 中）
- 将数据集的均值向量保存到 `self.mean_` 中

## 2.2 GMM聚类实现 (20%)

在 `submission.py` 中已提供GMM框架，需完成 `_estep` 和 `_mstep` 方法以实现EM算法。

**任务要求**

使用EM算法求解GMM参数，需要求解：

- `means_`: 每个高斯分布的均值向量
- `covariances_`: 每个高斯分布的协方差矩阵
- `weights_`: 混合系数（先验概率）

记 $N$ 表示样本数量，$K$ 表示聚簇数量。

#### 2.2.1 E-step实现 (10%)

完成 `_estep` 方法，计算每个样本属于每个聚簇的后验概率。

**输入**:

- 数据矩阵 $\boldsymbol{X}$（形状 $(N, D)$）
- 当前模型参数：`weights_`, `means_`, `covariances_`

**输出**:

- 后验概率矩阵 $\gamma$（形状 $(N, K)$），其中 $\gamma_{ik}$ 表示样本 $i$ 属于聚类 $k$ 的概率
- 对数似然下界的值

#### 2.2.2 M-step实现 (10%)

完成 `_mstep` 方法，使用最大似然估计更新模型参数。

**输入**:

- 数据矩阵 $\boldsymbol{X}$（形状 $(N, D)$）
- 后验概率矩阵 $\gamma$（形状 $(N, K)$）

**输出**:

- 更新后的 `weights_`, `means_`, `covariances_`

# 3. 实验训练与测试

## 3.1 运行训练

使用以下命令运行训练：

```
python train.py
```

我们可以在 `arguments.py`中调整对应的参数。

**参数说明：**

- `--dr_method`: 降维方法（`pca` 或 `autoencoder`）
- `--pca_components`: PCA主成分数量
- `--gmm_components`: GMM聚类数量（通常设为10）

**输出：**

- 模型保存到 `results/[$ 时间戳]/` 目录
- `pca.npz`: PCA模型参数
- `gmm/`: GMM模型参数目录

# 3.2 结果可视化

生成可视化图像，评估聚类效果：

```
python visualization.py --results_path ./results/[$时间戳] --plot_mode comparison --
sample_size 10000
```

这里对参数进行一些说明。

**命令行参数：**

- `--results_path`（必需）：指向你的实验结果目录
- `--plot_mode`（可选）：绘图模式
  - `comparison`（默认）：生成左右对比图，分别用真实标签和聚类标签着色
  - `clustering`：生成文本标签图，数字表示真实标签，颜色表示聚类
- `--sample_size`（可选）：可视化样本数量
  - 默认：60000（使用全部数据）
  - 建议：10000-20000（减少内存占用和绘图时间）

**Comparison模式（对比图）：**

每张图包含左右两个子图：

- **左图（True Label）**：使用真实标签着色
  - 不同数字用不同颜色表示（0-9共10种颜色）
  - 理想情况下同一数字应该聚集在一起
- **右图（Cluster Label）**：使用聚类结果着色
  - 不同聚类用不同颜色表示
  - 好的聚类结果应该与左图有类似的聚集模式

**Clustering模式（文本标签图）：**

- **数字**：显示每个样本的真实标签（0-9）
- **颜色**：表示该样本所属的聚类
- **优点**：可以直接看到每个点的真实标签和聚类归属
- **缺点**：样本多时数字可能重叠

# 4. 思考题(15%)

- 为什么需要先降维再聚类？(5pt)
- 从训练速度、降维效率、灵活性、数据分布保持程度、可视化效果等方面比较 PCA、t-SNE、AutoEncoder 三种降维方法(5pt)
- KMeans 和 GMM 都是聚类算法，请介绍一下他们的不同点和相同点(5pt)