MNIST

> 机器学习概论lab3
>
> **Author:**@Rosykunai
>
> **Date:**2024 年 11 月

这个文档是实验正文，有关实验环境配置和其它要求请查看仓库`README.md`文件

0. Intro

本次实验我们从最常见的无监督学习——聚类入手,不同于二维空间中的简单聚类,我们的实验目标是对高维空间中的数据(图像)做聚类。我们使用高斯混合模型来进行建模,你将使用课上学习的EM算法求解这个模型的参数。为了提高计算效率,在训练模型之前对数据做降维是很有必要的,PCA会帮助你完成这一步。我们还提供了tSNE和一个AutoEncoder作为另外的降维方法,你将看到它们的不同之处。最后，我们会尝试把你的聚类建模作为一种生成模型来生成类似的样本。

0.1 数据集

MNIST 数据集包含 70,000 张 28x28 的黑白手写数字图像，这些图像来自于两个 NIST 数据库。其中，训练集包含 60,000 张图像，验证集包含 10,000 张图像。每个数字对应一个类别，总共有 10 个类别，每个类别有 7,000 张图像（其中 6,000 张用于训练，1,000 张用于测试）。数据集中有一半的图像由美国人口普查局的员工书写，另一半由高中生书写（这种划分在训练集和测试集中是均匀分布的）。

0.2 文件组织

下面是对各个文件的简要介绍，更详细的内容请参考注释，

- `utils.py`: 工具函数包
- `load_resources.py`: 加载必要的实验资源
- `model.py`: pytorch模型定义在这里
- `train.py`: 训练脚本
- `grade.py`: 评分脚本
- `visualization.py`: 可视化降维后的数据
- `submission.py`:你需要修改并提交的文件

你只需要修改`submission.py`中的内容，不得修改其它文件中的内容(它们不会被提交).

0.3 加载实验资源

运行`load_resources.py`加载必要的实验资源:

```
python load_resources.py
```

这个脚本将会完成:

- 下载`mnist`数据集,预处理为`mnist_encoded`并保存在本地
- 下载AutoEncoder和ClassUNet模型,缓存到`C:\Users\{username}\.cache\huggingface\hub`目录下
- 测试pytorch设备

1. GMM聚类(20%)

在`submission.py`中助教已经实现了一个高斯混合模型的框架,以下面的参数初始化模型:

- `n_components`: 聚簇数量
- `data_dim`: 输入数据的维度

需要求解的参数:

- `means`: 每个高斯的均值向量
- `covs`: 每个高斯的协方差矩阵
- `pi`: 隐变量,混合参数

在`fit`方法中,我们使用`sklearn`库的`KMeans`聚类结果作为初始化`GMM`的初始化,然后运行EM算法求解参数。

记N表示样本数量, K表示聚簇数量

1.1 E-step(10%)

完成`GMM`类的`_e_step`方法,该方法接受数据矩阵$\boldsymbol{X}$作为输入,根据当前的参数计算每个样本属于每个聚簇的概率,返回γ(shape为$[N, K]$)(10 points)

1.2 M-step(10%)

完成`GMM`类的`_m_step`方法,该方法接受数据矩阵$\boldsymbol{X}$和上面输出的γ作为参数,使用最大似然估计更新模型的参数`pi`、`means`和`covs`(10 points)

提示:

- `_gaussian`方法接受数据矩阵$\boldsymbol{X}$,一个高斯的均值向量`mean`,协方差矩阵的逆`inv_cov`和协方差矩阵的行列式`det`作为输入,返回每个样本对给定聚簇的概率密度.
- 尽可能使用`numpy`库中的矩阵操作完成上面的步骤,使用`for`循环会导致计算效率过低
- 在更新`covs`时可以加上一个$10^{-6}\boldsymbol{I}$防止后续计算出现异常

2. PCA降维(8%)

在`submission.py`中助教已经实现了一个PCA的框架,以下面的参数初始化模型:

- `dim`: 降维后的维度

需要求解的参数:

- `components`: 保留的主成分
- `mean`: 数据的均值向量

记N为样本数量, D 为数据初始维度,d为降维后的维度

2.1 计算主成分(8%)

完成`PCA`类的`fit`方法,该方法接受数据矩阵$\boldsymbol{X}$作为输入,你需要保留d个主成分,并把数据集的均值向量保存到`self.mean`中。(8 points)

提示:为了防止计算结果出现复数,请使用`np.linalg.eigh`计算特征值和特征向量

3. Train

训练脚本已经为你写好,下面是各命令行参数的含义:

- `embedding_dim`: 使用PCA降维后的数据维度
- `use_pca`: 传递以启用PCA,否则使用AutoEncoder降维
- `n_components`: 聚簇数量,请保持为数据集的label数
- `max_iter`: EM算法的最大迭代次数
- `results_path`: 结果保存地址
- `seed`: 随机种子

启动训练脚本:

```
python train.py --use_pca
```

默认情况下,你的结果应该被保存到`../results/{datetime}`目录下, 你可以调试上面的参数,在保持一定计算效率的前提下,尽可能提高聚类的质量。

如果你想测试AutoEncoder的性能,不传入`use_pca`即可:

```
python train.py
```

4. 比较不同的降维方法

运行`visualization.py`脚本,这个脚本分别使用PCA,tSNE和AutoEncoder三种降维方法把数据集降到2维,并保存可视化图片。

命令行参数:

- `results_path`: 训练时的保存路径
- `cluster_label`: 传入以启用聚簇标签染色,否则以真实标签染色

可视化真实标签:

```
python visualization.py --results_path "../results/{datetime}"
```

可视化聚簇标签:

```
python visualization.py --results_path "../results/{datetime}" --
cluster_label
```

`tSNE`计算较慢,请耐心等待

5. 作为生成模型的GMM(7%)

高斯混合模型可以视为一种建模真实数据分布的一种方式,既然我们已经有了真实的数据分布,那么从中采样即可实现"生成"功能。

5.1 从GMM中采样(7%)

实现`sample_from_gmm`函数。该函数接受高斯混合模型`GMM`,`PCA`模型(用于还原维度),聚簇标签`label`, 样本保存地址`path`作为参数,从`label`对应的高斯分布中采样一个样本,然后用`PCA`类型的`inverse_transform`方法复原回原始数据维度。最后,你需要把这个样本处理为像素值在$[0, 255]$范围内,shape为[H, W]的图片,然后使用`Pillow`库保存。(7 points)

提示:确保传入给`Image.fromarray()`的numpy数组的元素类型是`np.uint8`

6. 测试(5%)

只有使用PCA降维的结果会计入Performance分数!!!

在测试之前,确保你已经通过`visualization.py`分别生成了真实标签和聚簇标签下染色的降维图片,

对比真实标签下的降维图片和聚簇标签下的降维图片,找到真实标签"6"对应的聚簇标签。(5 points)

把对应的聚簇标签作为参数`sample_index`传递给评分脚本,如果你想测试用AutoEncoder降维的性能,无需传递`sample_index`参数。

运行`grade.py`获取你的Performance分数

```
python grade.py --sample_index k --results_path "../results/{datetime}"
```

评分脚本在计算Performance得分的同时还会生成两张图片"6",其中一张由GMM生成,另一张由DDPM生成(原理见书面作业第四题)。

*生成图片并不会受随机种子影响,如果你觉得效果不好,可以多试几次。

*DDPM生成图片需要1000步,请耐心等待

如果你的操作正确,你应当能得到如下结果,其中模型参数被保存为`safetensors`格式Why?:

- `results/{datetime}`
  - `gmm`
    - config.json
    - gmm.safetensors
  - `pca`
    - config.json
    - pca.safetensors
  - config.yaml
  - cluster_ae.png
  - cluster_pca.png
  - cluster_tsne.png
  - ddpm_sample.png
  - gmm_sample.png
  - true_ae.png
  - true_pca.png
  - true_tsne.png

*如果未使用PCA,保存的结果不会包含`pca`文件夹

7. 回答问题(20 %)

- 从**训练速度**,**降维效率**,**灵活性**(eg.是否适用于各种类型的数据),**对数据分布的保持程度**,**可视化效果**这几个方面比较PCA,tSNE,AutoEncoder这三种降维方法的优劣(你可以列一个表格)(10 points)
- 从**生成效率**,**生成质量**,**灵活性**(eg.是否适用于各种类型的数据),**是否可控**(eg.生成指定类别的样本)这几个方面比较GMM和DDPM的优劣,(DDPM的原理参考书面作业第四题)(10 points)

H2 8. 反馈 (1%)

引言 你可以写下任何反馈，包括但不限于以下几个方面：课堂、作业、助教工作等等。

必填 你在本次作业花费的时间大概是？(1 points)

选填 你可以随心吐槽课程不足的方面，或者给出合理的建议。若没有想法，直接忽略本小题即可。