Gaussian Mixture Model

1.The data set you have chosen and its main characteristics.

The datasets with ids 53 and 109 were selected on the uci website.The dataset with id 53 is Iris, with a small number of features, only four, and all of them are continuous and size-differentiated feature data.The dataset with id 109 is a wine dataset with a high number of features, 13 dimensions.

ps:There is no missing data in the above three datasets.

2.The design of the different modules of your program. In describing each module, you should relate your design to the theories you have learnt in the class.

- **common.py** holds the hyperparameters for model training and model testing, including training epoch, iteration count, gradient descent parameters, etc.
- The **config** folder contains settings for model hyperparameters for different datasets
- Under the **process** folder **ucidata.py** is the modules that process the data and improve the quality of the data. After reading the data into memory through the data import method provided by uci , the data is then processed in the way mentioned in question 2, and the data is categorized into raw data, training set data, test set data, metadata, etc., and put into a folder with the name **id**.
- The **layer** folder defines classes for various network layers, which are fullyconnet layer,Relu layer,softmax layer.
- **test.py** is used to test the accuracy of Gaussian mixture models.
- **train.py** is the model training module.

3.The prediction results and your interpretation of the results.

3.1 model training

The code is set up with a training set to test set ratio of **4:1**.When training the model, the epoch amount is the same for the same dataset.

$$
\begin{aligned}
\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) &= \arg\max_{\boldsymbol{\theta}} \prod_i^N P(\boldsymbol{y}^{(i)}|\boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} \sum_i^N \log P(\boldsymbol{y}^{(i)}|\boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} \sum_i^N \log[\sum_k^K P(\boldsymbol{y}^{(i)}, z_k|\boldsymbol{\theta})] \\
&= \arg\max_{\boldsymbol{\theta}} \sum_i^N \log[\sum_k^K P(z_k|\boldsymbol{y}^{(i)}, \boldsymbol{\theta}^t) \frac{P(\boldsymbol{y}^{(i)}, z_k|\boldsymbol{\theta})}{P(z_k|\boldsymbol{y}^{(i)}, \boldsymbol{\theta}^t)}] \\
&\Rightarrow \text{Jensen 不等式} \\
&\arg\max_{\boldsymbol{\theta}} \sum_i^N [\sum_k^K P(z_k|\boldsymbol{y}^{(i)}, \boldsymbol{\theta}^t) \log \frac{P(\boldsymbol{y}^{(i)}, z_k|\boldsymbol{\theta})}{P(z_k|\boldsymbol{y}^{(i)}, \boldsymbol{\theta}^t)}] \\
&= \arg\max_{\boldsymbol{\theta}} \sum_i^N [\sum_k^K P(z_k|\boldsymbol{y}^{(i)}, \boldsymbol{\theta}^t) \log P(\boldsymbol{y}^{(i)}, z_k|\boldsymbol{\theta})] \\
&= \arg\max_{\boldsymbol{\theta}} \sum_i^N \mathbb{E}_{z|\boldsymbol{Y}, \boldsymbol{\theta}^t \sim P} \log(P(\boldsymbol{y}^{(i)}, z_k|\boldsymbol{\theta}))
\end{aligned}
$$

In fact, through the derivation of the Gaussian mixture model formula above, we find that its essence is still maximum likelihood estimation, but it maximizes a lower bound derived from Jensen's inequality.Due to the fact that log is a strongly concave function, the second-order derivative, i.e. the Hessian matrix, cannot be a zero matrix, and therefore cannot be judged using the equality condition of the Jensen inequality.So $\max_{\boldsymbol{\theta}} \sum_i^N \mathbb{E}_{z|\boldsymbol{Y}, \boldsymbol{\theta}^t \sim P} \log(P(\boldsymbol{y}^{(i)}, z_k|\boldsymbol{\theta})) < \max_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$is inevitable.Since the Gaussian mixture model is an iterative algorithm that can quickly converge to the optimal solution, we only need to ensure that the maximum value of such a lower bound is as large as possible. This indicates that the quality of the Gaussian mixture model largely depends on the selection of initial values.

So, the experiment used two methods, random and k-means, to initialize the parameters$\alpha, \boldsymbol{u}, \boldsymbol{\Sigma}$.

- Using a random method to initialize parameters, record the parameter values for every 10 epochs, and it was found that convergence usually occurs after 50 epochs. And we found that the convergence values are different each time (which precisely verifies the description of the formula), and there are also significant differences in clustering performance, which will be discussed in the model testing in 4.2.
- Using the k-means method to initialize parameters (i.e. using the center of the k-means clustering result as the initial parameter for model iteration), it was found that convergence can be achieved in 10 epochs and there is a good clustering effect.

Due to the strict derivation process based on mathematical formulas and the use of for loops, training on datasets with large amounts of data can easily lead to excessively long processing times in **train.py**.The optimized version of matrix multiplication is provided in **better_train.py**, which enables training to be completed quickly regardless of the size of the dataset.Please refer to **ReadMe** for specific usage instructions

3.2 model test

3.2.1

Since the sample contains labels, we can calculate the most likely normal distribution parameter $\theta$for each category. Then, we compare the two different initialization methods, the distribution generated by the existing Gaussian mixture model in sklearn, with the true value, and calculate the KL divergence to observe the differences in the distribution. The table shows the KL diversity results for different datasets using different parameter initialization methods. As we are more concerned about whether each category is correct, we calculated the KL divergence for the same cluster under different methods and took their average results. It can be seen that the method of initializing parameters using kmeans is better.

method\dataset	random	k_means	sklearn
iris	77.25	3.833	3.833
wine	80.64	40.56	29.22

3.2.2

Due to the fact that the samples contain labels, the accuracy of the Gaussian mixture model can be evaluated using the existing information of the samples, namely labels. The table below shows the accuracy of our method and the sklearn method on different datasets. (Recall rate, F1score, etc. are similar and will not be listed one by one.)

dataset\method	ours	sklearn
iris	96.7%	97.5%
wine	38.7%	50%

For the dataset Wine, both algorithms have relatively low accuracy because the inter class difference is very small and cannot be clearly classified through clustering.If according to the theory of high-dimensional statistics, in fact, the higher the dimension, even the feature vectors of the same category will be almost orthogonal. Therefore, regardless of whether they belong to the same category or not, it is almost impossible to distinguish what category they belong to through unsupervised learning such as clustering

4.Limitations and possible improvements of the program.

- Although the indicators proposed in this article can already effectively measure the quality of the model, many experiments do not have labels, and more evaluation indicators for clustering under unsupervised learning should be used.
- Since the model training is simple and there are not a large number of hyperparameters, there is no division of the validation set.