Report of GaussMixModel

H24116049 莊秉宸

● Understanding the Program

Before diving into the analysis of different random seeds, I spent some time to explore what the simulation does. The **Gaussian_poolOrNot.c** program widely uses the GNU Scientific Library (GSL) package to generate data from the distribution and sample parameters form the prior probability distribution. The relevant functions are stored in **GSLfun.c** program.

The program has four main algorithms implemented, and I have added some comments to help me understand the code:

- **data_prob_1component_bySampling()**

  This algorithm calculated the mean probability of dataset sampled from the Gaussian distribution under parameters from the prior probability distribution. In each iteration (totally iterates for '**sampleRepeatNum'** times), it first selects parameters $(\mu_{iter}, \sigma_{iter})$ from the prior Gaussian and gamma distribution, then calculates the joint probability of the dataset under the $N(\mu_{iter}, \sigma_{iter})$ (Note: Since data points are iid, the joint probability is the product of each data point). Finally, it returns the average probability across iterations, which represents the overall likelihood.

- **data_prob_2component_bySampling()**

  This algorithm extends the sampling method from single-component to two-component mixture model. The parameter 'mixConf' is selected by Jefferey's prior beta distribution $Beta(0.5, 0.5)$, while 'Gauss1', 'Gauss2' are selected using the same method as in data_prob_1component_bySampling(). Since the probability of each data point can be calculated in formula:

  $$(1 - MixConf) * Gauss1(data_i) + MixConf * Gauss2(data_i)$$

  We can obtain the average joint probability of the datasets.

- **data_prob_1component_bySumming()**

  In this function, parameters (μ, σ) are selected from the precomputed distribution quantiles (μ from '**cdfInv_Gauss**' and σ from '**cdfInv_gamma**', traversing all the combinations of them). Similarly to the calculation of the joint probability of points generated by parameters in sampling method, the function returns the average likelihood.

- **data_prob_2component_bySumming()**
  This algorithm calculates the average probability of observing a dataset under a two-component mixture model by traversing over all possible parameter combinations. Similar to data_prob_1component_bySumming(), it selects parameters from precomputed quantiles and evaluates the joint probability of the dataset for each combination.

After constructing the functions, the code applies them to datasets generated from the prior distribution for both the one-component and mixture model. Then it compares **'prob_data1_bySampling'** and **'prob_data2_bySampling'** values (similar comparison for Summing method) to determine whether the datasets originate from an one-component prior distribution or a mixture model distribution. Lastly, the code states the performance of two method (Sampling and Summing):

```
By sampling: Model1 data, correct selection 8/10
             Model2 data, correct selection 10/10
By summing:  Model1 data, correct selection 10/10
             Model2 data, correct selection 9/10
```

(the outcome of the default parameters)

To interpret the result, the Summing method is more accuracy identifying the cases than Sampling method (10/10 versus 8/10), while the latter performed well in identifying the mixture model. To further compare the advantages of the methods, I began conducting some simulation with different random seeds.

- **Simulation playground**

In the simulation process, I use **change the seeds** to observe the selections of two methods under different datasets:

```
By sampling: Model1 data, correct selection 9/10
             Model2 data, correct selection 7/10
By summing:  Model1 data, correct selection 8/10
             Model2 data, correct selection 7/10
```

(the outcome with GSL_RNG_SEED = gsl_rng_mt19937_1999)

In this case, the sampling method preforms well in identifying dataset from one-component model, while two method have both 70% correctly identifying the two-component model.

```
By sampling: Model1 data, correct selection 9/10
             Model2 data, correct selection 5/10
By summing:  Model1 data, correct selection 7/10
             Model2 data, correct selection 6/10
```

(outcomes with GSL_RNG_SEED = gsl_rng_mt19937_1998)

In this case, the sampling method performs better on Model1 than the summing method, while the latter performs slightly better on the Model2 data though the difference is not significantly.

To summarize the advantages and disadvantages of two method:

➢ Sampling

Since this method estimates the probability through extensively random sampling of the parameter space, it is more **suitable and scalable for higher-dimensional space**, but because of the stochastic sampling process, the result of two sampling process may not be the same.

➢ Summing

The Summing method can yield more deterministic and consistently result since it uses the precomputed quantiles and has higher accuracy in low dimension space, provided that the number of precomputed quantiles is sufficient. But, this method is not suitable for high dimension space.

● Visualization of generated dataset

In this visualization process, I plot 4 distributions of generated datasets with 40 data points each.

For the first dataset (GSL_RNG_SEED = gsl_rng_mt19937, (mu,sigma)=(0.54,3.54)) we visualize the distribution:

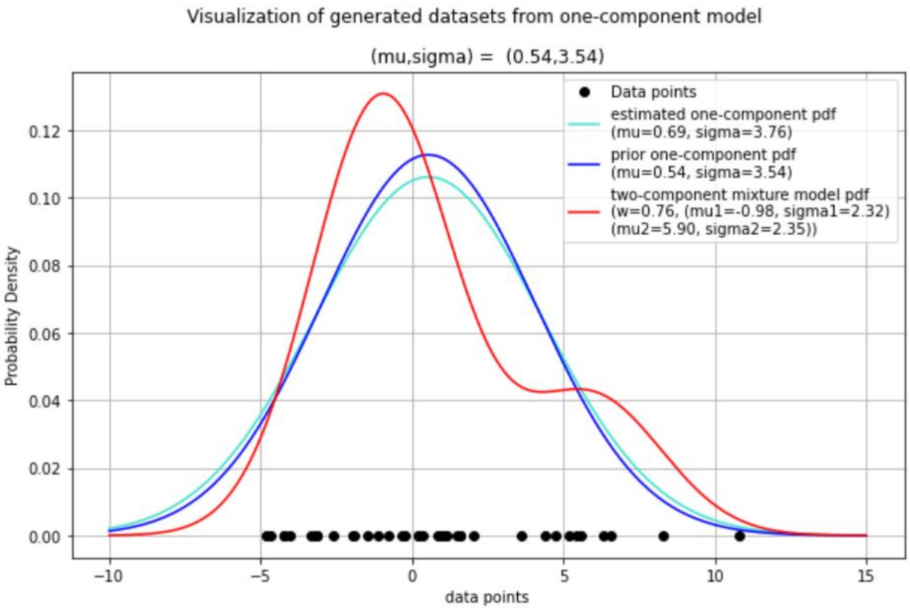

Figure1: Visualization of datasets (mu,sigma)=(0.54,3.54)

The blue line represent the prior gaussian distribution with default parameters (mu,sigma) = (0.54, 3.54), the turquoise-color line shows the probability density function estimated by sample mean and sigma (mu, sigma)=(0.69, 3.76). In addition, we also construct a two-component gaussian mixture model (GMM) using Expectation Maximization. While the one-component model closely fits the distribution line, the mixture model captures more complexity despite showing slight lack-of-fit.

For the dataset (GSL_RNG_SEED = gsl_rng_mt19937, (mu,sigma)=(4.19,3.16)):

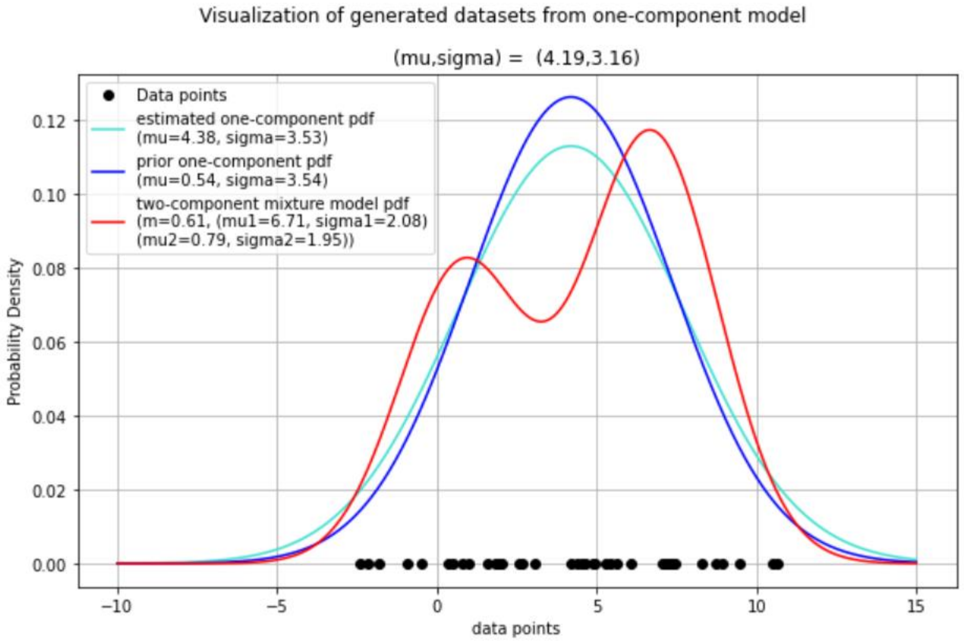

Figure2: Visualization of datasets (mu,sigma) = (4.19,3.16)

For the third dataset:

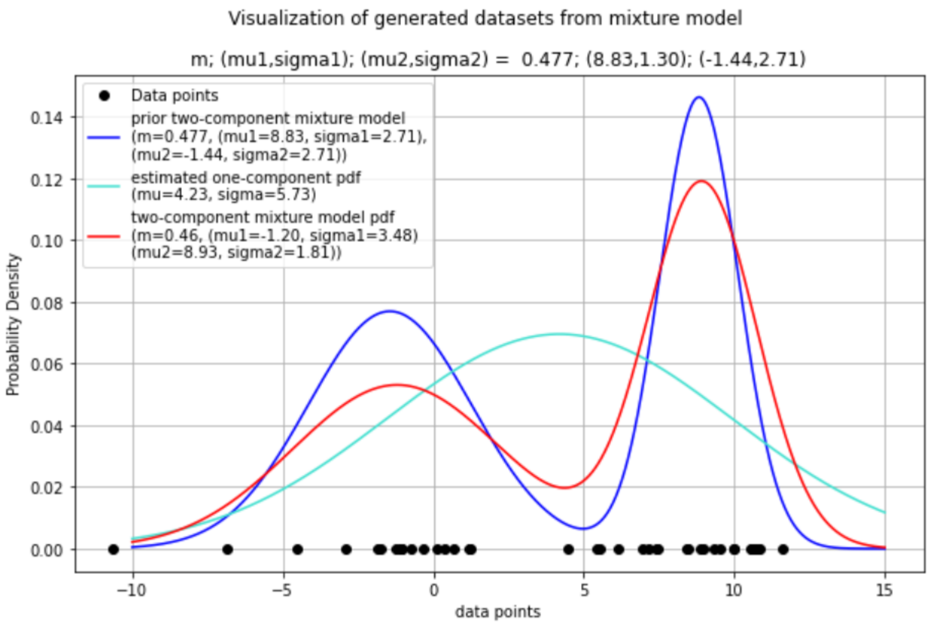

Figure3: Visualization of datasets m; (mu1,sigma1); (mu2,sigma2) = 0.477; (8.83,1.30); (-1.44,2.71)

The blue line represents the prior probability density function of two-component mixture model, the red line shows the estimated gaussian mixture model obtained by Expectation Maximization method. Note that while the turquoise-color line is the estimation of one-component model, which clearly does not fit the data points well because of its low complexity.

For the third dataset:

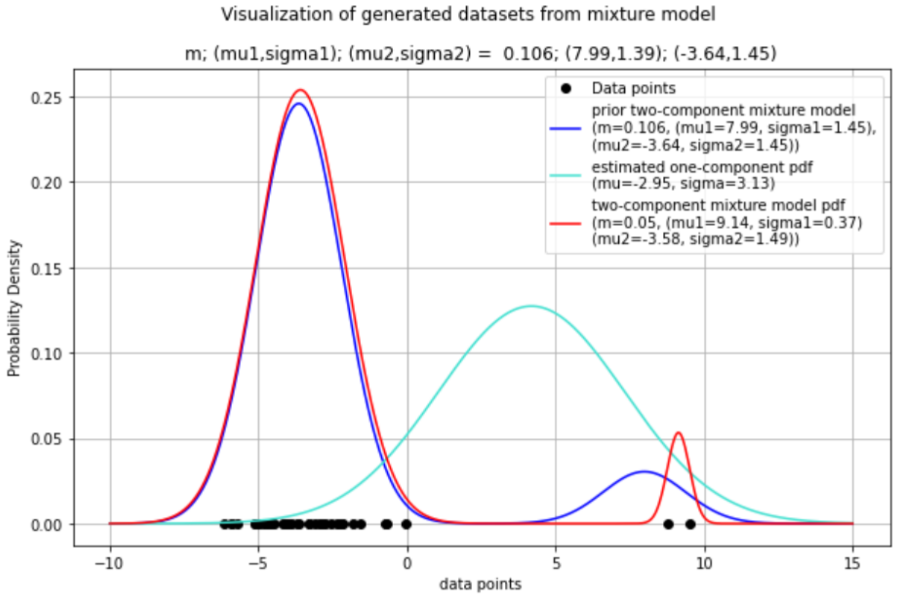

Figure3: Visualization of datasets m; (mu1,sigma1); (mu2,sigma2) = 0.477; (8.83,1.30); (-1.44,2.71)