**LE MINH DONG**

**HOME WORK 2**

* **KMEAN Problem**

K means minimizes the sum of the squared Euclidean distances of each data vector from its closest parameter vector.

**10 code words of cluster representatives with the feature vectors of 2048 dimensions by**

1. Choose randomly 10 code words from the range of (0,1).
2. Choose 10 code words from mean values of data\_train for each class.
3. Choose 10 code words from mean value of data\_test for all classes
4. Choose 10 code words from the average of mean value of (2) and (3) for each class.
5. Choose 10 code words by combining (2), (3) and the priority probability of each class in data\_train.

**Stop condition:**

The Representative Vector of each class are not changed between two successive iterations

**Results:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NO** | **Initial codewords** | **Stop condition** | **Sum of distance variation on the whole of data\_test** | **No. Interaction** | **error** |
| 1 | 10 code words from random value in the range of (0,1) | The Representative Vector of each class are not changed between two successive iterations | 2.95 e08 | 9 | 0.998 |
| 2 | 10 code words from Mean value of data\_train for each class | The Representative Vector of each class are not changed between two successive iterations. | 1.74 e08 | 26 | 0.7806 |
| 3 | 10 code words from Mean value of data\_test for all class | The Representative Vector of each class are not changed between two successive iterations | 5.96e08 | 1 | 0.9618 |
| 4 | 10 code words from the average of (2) and (3) | The Representative Vector of each class are not changed between two successive iterations | 1.63 e08 | 22 | 0.7731 |
| 5 | 10 code words from the average of (2) (3) and the prior probability of each class in data train. | The Representative Vector of each class are not changed between two successive iterations | 1.63e08 | 22 | 0.7794 |

**Discussion:**

Generally, the result of K mean in application to data\_test are not good. Because K means minimizes the sum of the squared Euclidean distances of each data vector from its closest parameter vector which is suitable for classifying compact clusters. In case of dat\_test that contain the feature vector of 2048 dimensions. This is complex data, so the performance of K means are not too sanguine.

* In case (1): using 10 random code words from the range of (0,1). The results are really bad, the initial code words are far from real data. On the other hand, K mean returns clustering corresponding to local minimal value of distance to center vectors. Therefore, the algorithm stops at the interaction of 9, the error of 0.998 and the sum of distance variation on the whole of data is 2.95 e08
* In case (2): using 10 code words from the mean values of data\_train for each class. The outputs are better with the error of 0.7806 and the sum of distance variation of 1.74 e08. These results prove that initial code words have little relation with data\_test.
* In case (3): using 10 code words from mean value of data\_tes for all classes. Of course, the performance are extremely bad because the initial code words are same for all classes. It makes the algorithm stops at the first interaction.
* In case (4), (5): combining the information mean values , prior probability of each class in data\_train and the mean values of data\_test to create initial code words. The results are better with the sum of distance variation of 1.63 e08, the error of case (4) and (5) are 0.7731 and 0.7794, respectively
* **GMM PROBLEM**

**Idea: Set the number of Gaussian mixture**

1. **2 Gaussian mixture for each class.**

* Divide the data\_train into 2 parts.
* Estimate the Gaussian model for each class in each part.
* Base on the results of mean value, covariance matrix and prior probability of each class in two parts create mixed Gaussian model for each class
* Test the performance of mixed Gaussian model for each class in data\_test.

1. **3 Gaussian mixture for each class.**

* Divide the data\_train into 3 parts.
* Estimate the Gaussian model for each class in each part.
* Base on the results of mean value, covariance matrix and prior probability of each class in three parts create mixed Gaussian models for each class
* Test the performances of mixed Gaussian models for each class in data\_test.

**Current problem: facing the problem of exceeding memory to create GMM models and test.**