Pattern Recognition
CS669

ASSIGNMENT 4

Bayes Classifier using PCA and GMM

Group Number 13

Suryavanshi Virendrasingh	B16037
Abhishek Pal	B15203
Nemani Sri Hari	B15224

Contents

1. Problem Description

Data-sets:

- Data-set: Real world data set:
  - 3 class scene image dataset: Consider the 32-dimensional BoVW representation from Assignment-2.

Classifiers:

- Bayes Classifier using PCA and GMM

Objective:

1. Build Bayes classifier using Gaussian mixture model (GMM) with 1, 2, 4 and 8 mixtures on the reduced dimensional representations of Dataset-2 obtained using PCA.
2. For each data-set we do :
   - Classification accuracy, precision for every class, mean precision, recall for every class, mean recall, F-measure for every class and mean F-measure on test data.
   - Confusion matrix based on the performance for test data.
   - Plot of eigen values in assending order during PCA.

2. Solution Approach

1 PCA (Principal Component Analysis)

Principle Component Analysis is defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.PCA is a dimensionality reduction technique.It is not a classifier.PCA gives a set of values of linearly uncorrelated variables called principal components.

Method

- Subtract the mean of each dimension from its data. This produces the data set whose mean is zero.
- Compute Covariance matrix.
- Perform eigen analysis of Covariance matrix.

  q_i =eigen vectors of Covariance matrix.

  λ_i= eigen values of Covariance matrix

  By taking the eigenvectors of the covariance matrix, we have been able to extract the directions that characterise the data most appropriately. The rest of the steps involve transforming the data so that it is expressed in terms of these directions.
- Sort the eigen values i in descending order, to sort the directions according to which directions store maximum information content.
- Take the eigen vectors corresponding to the top L eigen values.

  q_i now refers to the top L eigen vectors.
- We project these vectors, qi onto the original data by taking its transpose and then dot product with the original data.
- Now, this is our new data to work on, which has only L dimensions. We apply K-Means and GMM on it and get our results.

Eigen vectors are the directions corresponding to the variance of the projected data. The directions that we have chosen are therefore, those in which the information content is maximum. We project our original data onto these directions to get the important data, in reduced number of dimensions.

2 K-Means Clustering

K-Means clustering is the simplest clustering algorithm. We have to specify the number of clusters and it gives a decent approximation of the different partitions. K-Means is mostly used as a pre-clustering algorithm to get a decent starting point for the actual clustering algorithm.

In this assignment also, we use K-Means for pre-clustering the data obtained from PCA that is then classified using GMM.

Method

Using K-Means clustering, we assign data points to clusters, as well as a set of vectors μ_{k}, such that the sum of the squares of the distances of each data point to its closest vector μ_{k}, is minimum. This sum is given by,

$$J \;=\; \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}||x_n \;-\; \mu_{\mathrm{k}}||^2 \tag{2..1}$$

The steps of K-means clustering are as follows :

First we choose some initial values for the μ_{k}. In the first phase, we minimize J with respect to the r_{nk} , keeping the μ_{k} fixed. This tells us which point belongs to which cluster .This is given by the formula,

$$r_{nk} = \begin{cases} 1, & \text{if } k = argmin_j||x_n \;-\; \mu_{\mathrm{k}}||^2 \\ 0, & \text{otherwise} \end{cases} \tag{2..2}$$

In the second phase we minimize J with respect to the μ_{k}, keeping r_{nk} fixed. This gives us the new k for each cluster. This is given by,

$$\mu_{\mathrm{k}} \;=\; \frac{\Sigma_n r_{nk} x_n}{\Sigma_n r_{nk}} \tag{2..3}$$

This two-stage optimization is then repeated until convergence criteria is satisfied.(i.e. Log likelihood is lesser than a certain threshold).

3 GMM (Gaussian Mixture Model)

In Gaussian Mixture Model, we find the responsibility that each cluster takes in generating each point.Basically, we find the probability of each point being in each cluster. Even in the GMM technique, the way of initialising means is not specified. Since we already have a decent approximation from the K-means algorithm, we use the results of the K-means clustering only. Taking the final means and variances

of the different clusters from the K-means clustering results, we apply GMM to it.

For each data point x_n we define a set of binary indicator variables p_{ik} which describe which cluster the point belongs to. To find π_{k}, we need to know which data points belong to which cluster, and how many data points belong to which cluster. So, using K-means clustering, we first classify them according to minimum distance measure and calculate π_{k} for each cluster. Once we have initial values for mean, variance and π for each cluster, we find the responsibility term γ for each data point π with respect to each cluster k,using the formula,

$$\gamma(\mathrm{z_{nk}}) = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j, \Sigma_j)} \tag{2..4}$$

This is known as the expectation step of the EM algorithm. Using γ, we estimate the parameters μ_{k}, Σ_{k} and π_{k} again such that the total data likelihood is maximized.

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \tag{2..5}$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T \tag{2..6}$$

$$\pi_k^{new} = \frac{N_k}{N} \tag{2..7}$$

where,

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \tag{2..8}$$

This is known as the Maximization step of the EM algorithm.

After every iteration,we have to find log likelihood value using,

$$\ln p(X|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln \{ \sum_{k=1}^{K} \pi_k N(x_n|\mu_k, \Sigma_k) \} \tag{2..9}$$

And we repeat the process till l_{new} - l_{old} is less than a particular threshold value. This is called **Convergence Criteria.**

3. Results and Plot

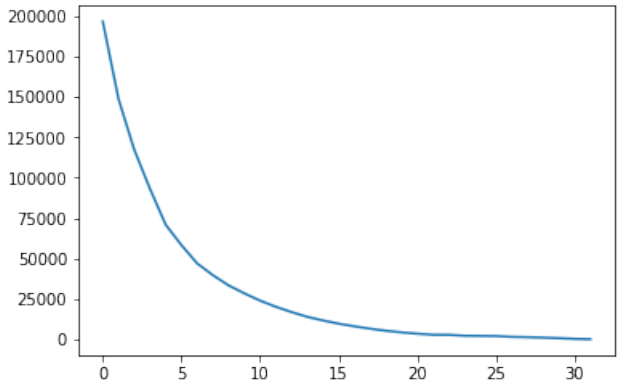

Figure 3..1. Information Loss w.r.t the dimensions

Confusion Matrix, Precision, Recall and F-measure

Dimensions : 2
Mixtures : 1
Accuracy : 32.66%

	Class1	Class2	Class 3
Class1	4	3	43
Class2	2	10	38
Class3	8	7	35

(a) Confusion Matrix

	Class1	Class2	Class3
Accuracy	62.67	66.67	36.00
Precision	28.57	50.00	30.17
Recall	8.00	20.00	70.00
F-Measure	12.50	28.57	42.16

(b) Analysis (in Percentage)

Table 3..1. Result for all the 3 classes

Dimensions : 2
Mixtures : 2
Accuracy : 34.67%

	Class1	Class2	Class 3
Class1	1	7	42
Class2	3	17	30
Class3	1	15	34

(a) Confusion Matrix

	Class1	Class2	Class3
Accuracy	64.67	63.33	41.33
Precision	20.00	43.58	32.07
Recall	2.00	34.00	68.00
F-Measure	3.63	38.20	43.58

(b) Analysis (in Percentage)

Table 3..2. Result for all the 3 classes

Dimensions : 2
Mixtures : 4
Accuracy : 41.99%

	Class1	Class2	Class 3
Class1	33	5	12
Class2	27	14	9
Class3	25	9	16

(a) Confusion Matrix

	Class1	Class2	Class3
Accuracy	54.00	66.67	63.33
Precision	38.82	50.00	43.24
Recall	66.00	28.00	32.00
F-Measure	48.88	35.89	36.78

(b) Analysis (in Percentage)

Table 3..3. Result for all the 3 classes

Dimensions : 2
Mixtures : 8
Accuracy : 37.99%

	Class1	Class2	Class 3
Class1	36	3	11
Class2	29	8	13
Class3	29	8	13

(a) Confusion Matrix

	Class1	Class2	Class3
Accuracy	52.67	64.67	59.33
Precision	38.29	42.10	35.13
Recall	72.00	16.00	26.00
F-Measure	50.00	23.18	29.88

(b) Analysis (in Percentage)

Table 3..4. Result for all the 3 classes

Dimensions : 8
Mixtures : 1
Accuracy : 40.66%

	Class1	Class2	Class 3
Class1	39	4	7
Class2	29	7	14
Class3	28	7	15

(a) Confusion Matrix

	Class1	Class2	Class3
Accuracy	54.67	64.00	62.67
Precision	40.65	38.89	41.67
Recall	78.00	14.00	30.00
F-Measure	53.42	20.58	34.88

(b) Analysis (in Percentage)

Table 3..5. Result for all the 3 classes

Dimensions : 8
Mixtures : 2
Accuracy : 30.66%

	Class1	Class2	Class 3
Class1	4	5	41
Class2	8	8	34
Class3	8	8	34

(a) Confusion Matrix

	Class1	Class2	Class3
Accuracy	58.67	63.33	39.33
Precision	20.00	38.09	31.19
Recall	8.00	16.00	68.00
F-Measure	11.42	22.53	42.76

(b) Analysis (in Percentage)

Table 3..6. Result for all the 3 classes

Dimensions : 8
Mixtures : 4
Accuracy : 33.99%

	Class1	Class2	Class 3
Class1	1	35	14
Class2	3	32	15
Class3	3	29	18

(a) Confusion Matrix

	Class1	Class2	Class3
Accuracy	63.33	45.33	59.33
Precision	14.28	33.33	38.29
Recall	2.00	64.00	36.00
F-Measure	3.50	43.83	37.11

(b) Analysis (in Percentage)

Table 3..7. Result for all the 3 classes

Dimensions : 8
Mixtures : 8
Accuracy : 40.66%

	Class1	Class2	Class 3
Class1	39	5	6
Class2	29	17	4
Class3	27	18	5

(a) Confusion Matrix

	Class1	Class2	Class3
Accuracy	55.33	62.67	63.33
Precision	41.05	42.50	33.33
Recall	78.00	34.00	10.00
F-Measure	53.79	37.78	15.38

(b) Analysis (in Percentage)

Table 3..8. Result for all the 3 classes

4. Observations and Inferences

1. We observed that there is less than ten percent information loss after eight dimensions, so we considered data in eight dimensions and two dimensions for comparison in this report.