EE 573 Pattern Recognition Project 2

Oğuzhan Sevim

I. INTRODUCTION

In this project, we design a classifier based on Bayesian Decision Rule. Because of some characteristics of the given dataset feature reduction will first be applied. In the remaining sections, first, the decision rule is explicitly derived. Then, by applying it to the training set (the set that is used to estimate distributions of the classes) and test set, we get some performance metrics.

A. Dataset

The given dataset consists of $n = 1315$ samples where each sample has $d = 500$ features. The given dataset is classified into $C = 8$ different classes. For each class c, where $c = 1,\ldots,8$, we portion the randomly selected 75% of the class c data as D_c and the remaining 25% as T_c. Since distribution of D_c sets are used for future predictions, they can be considered as training data. Therefore, in the remaining of this report, D_c will be referred as training sets, where we will call T_c as test sets.

For the covariance matrix of class c to be nonsingular (invertible), $n_c > d$ should be satisfied, where n_c denotes the number of samples in class c. Since each class in the dataset consists of few hundreds of samples, it is not feasible to continue with the given dataset. Instead, a feature reduction method (e.g., PCA) is required. In the remaining part of the project, feature size $d = 500$ will be reduced to three different sizes $d^{'} = 10, 30, 50$.

B. Gaussian Distribution

The prior probability for class c is simply defined as

$$P(w_c) = \frac{|D_c|}{\sum_{j=1}^{C} |D_j|}. \tag{1}$$

Also, for each class c, the likelihood $p(x|w_c)$ is approximated by the normal distribution $\mathcal{N}(\mu_c, \Sigma_c)$. Here, the mean μ_c and the covariance matrix Σ_c are defined as follows:

$$\mu_c = \frac{1}{|D_c|} \sum_{x \in D_c} x, \tag{2}$$

$$\Sigma_c = \frac{1}{|D_c|} \sum_{x \in D_c} (x-\mu_c)(x-\mu_c)^T. \tag{3}$$

Then, by using μ_c and Σ_c, the likelihoods can be calculated as

$$p(x|w_c) = \frac{1}{(2\pi)^{d/2}|\Sigma_c|^{1/2}} exp[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)]. \tag{4}$$

II. DECISION RULE (PART A)

For a given sample x, by the Bayes Rule, we decide w_c if

$$P(w_c|x) > P(w_j|x), \forall j \neq c. \tag{5}$$

Also, by the Bayes formula, the posterior probabilities can be expressed as

$$P(w_c|x) = \frac{p(x|w_c)P(w_c)}{p(x)}. \tag{6}$$

By using (1), posteriors can be rewritten by

$$P(w_c|x) = \frac{1}{p(x)\sum_{j=1}^{C}|D_j|} p(x|w_c)|D_c|. \tag{7}$$

Here, the first fractional term is just a constant independent of w_c. Therefore, the decision rule can be simplified as follows: Decide w_c if

$$p(x|w_c)|D_c| > p(x|w_j)|D_j|, \forall j \neq c. \tag{8}$$

We could also derive another decision rule by taking natural logarithm of both sides of (8). However, there no consecutive multiplication of density functions in the given equation. So, it is not necessary to use natural logarithms.

III. PERFORMANCE METRICS ON TRAINING AND TEST SETS (PARTS B-C-D)

In order to evaluate the performance of the classification algorithm, precision and recalls are used. For $d^{'} = 10, 30, 50$ these metrics are calculated both for training sets (75%) and test sets (25%) separately.

TABLE I
PRECISION OF TRAINING DATASET

Class	1	2	3	4	5	6	7	8
d' = 10	0.87	0.66	1.00	0.71	0.46	1.00	0.67	0.91
d' = 30	0.95	0.93	1.00	0.93	0.68	1.00	0.92	1.00
d' = 50	0.95	0.96	1.00	0.99	0.98	1.00	1.00	1.00

TABLE II
RECALL OF TRAINING DATASET

Class	1	2	3	4	5	6	7	8
d' = 10	1.00	0.68	1.00	0.67	0.43	1.00	0.96	0.48
d' = 30	1.00	0.90	1.00	0.73	0.98	1.00	1.00	0.90
d' = 50	1.00	0.97	1.00	0.94	1.00	1.00	1.00	1.00

TABLE III
PRECISION OF TEST DATASET

Class	1	2	3	4	5	6	7	8
d' = 10	0.87	0.65	0.83	0.61	0.38	0.67	0.37	0.35
d' = 30	0.71	0.70	0.74	0.65	0.39	0.53	0.47	0.55
d' = 50	0.54	0.64	0.66	0.64	0.39	0.45	0.44	0.47

TABLE IV
RECALL OF TEST DATASET

Class	1	2	3	4	5	6	7	8
d' = 10	0.67	0.52	0.83	0.69	0.40	0.67	0.63	0.21
d' = 30	0.67	0.73	0.83	0.50	0.49	0.67	0.45	0.49
d' = 50	0.67	0.71	0.75	0.55	0.46	0.39	0.31	0.60

It can be easily observed from Tables I and II that our classifier performs better on the training data as we increase the $d^{'}$. Since higher $d^{'}$ means more separated data, this result complies with the theoretical expectations.

As expected, performance metrics for the test data is generally lower than the training data. Also, from Tables I and II, classes $c = 3, 6$ our classifier performs perfect even for $d^{'} = 10$. As $d^{'}$ increases, the test data points of these particular classes seems to become outliers in their distributions which can be seen on Tables III and IV. Other than that, precision and recall of test data do not show any general behaviour (as I can see). I would expect the test data to fit their class approximations better as $d^{'}$ increases, but this is not the case.

The corresponding code of this report can be found in main.m file.