EE 573 Pattern Recognition Project 3

Oğuzhan Sevim

I. INTRODUCTION

In this project, we design a classifier based on Bayesian Decision Rule. Due to some characteristics of the given dataset, feature reduction will first be applied. In the remaining sections, we will assume that the class conditional probability distribution of the samples are not know. In order to estimate the likelihoods, we will use maximum likelihood estimation (MLE) and Bayesian estimation methods. Then, by combining them with Bayesian decision rule, we will separately make classifications on training and test sets.

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 $d^{'} = 50$.

II. TASK 1: MAXIMUM LIKELIHOOD ESTIMATION

In this section, we assume that the likelihoods $p(x|w_c)$ are approximated by the normal distribution $\mathcal{N}(\mu_c,\Sigma_c)$ in which μ_c and Σ_c parameters are not known. In order to estimate these parameters for each class c, MLE method will be used with the following log-likelihood function:

$$l(\theta_c) = \sum_{k=1}^{|D_c|} ln[p(x_k|\theta_c)]. \tag{1}$$

Here, θ_c is the vector which composed of the elements of Σ_c matrix and μ_c vector. In order to find the θ_c vector that maximizes $l(\theta_c)$, we simply need to find θ_c vector where the gradient of $l(\theta_c)$ is equal to zero. This condition is shown as follows:

$$\nabla_{\theta_c} l(\theta_c) = \sum_{k=1}^{|D_c|} \nabla_{\theta_c} ln[p(x_k|\theta_c)] = \mathbf{0} \tag{2}$$

By solving (2), the maximum likelihood estimates (biased) are calculated as:

$$\hat{\mu}_c = \frac{1}{|D_c|}\sum_{x\in D_c} x, \tag{3}$$

$$\hat{\Sigma}_c = \frac{1}{|D_c|}\sum_{x\in D_c} (x-\mu_c)(x-\mu_c)^T. \tag{4}$$

Also, the prior probability for each class c is assumed to be known by

$$P(w_c) = \frac{|D_c|}{\sum_{j=1}^{C}|D_j|}. \tag{5}$$

Then, by using (3) and (4), the class can be calculated as

$$p(x|w_c) = \frac{1}{(2\pi)^{d/2}|\hat{\Sigma}_c|^{1/2}} exp[-\frac{1}{2}(x-\hat{\mu}_c)^T\hat{\Sigma}_c^{-1}(x-\hat{\mu}_c)]. \tag{6}$$

A. Decision Rule

After estimating the likelihood, we now can derive the decision rule. For a given sample x, by the Bayes Rule, we decide w_c if

$$P(w_c|x) > P(w_j|x), \forall j \neq c. \tag{7}$$

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

$$P(w_c|x) = \frac{p(x|w_c)P(w_c)}{p(x)}. \tag{8}$$

By using (5), posteriors can be rewritten by

$$P(w_c|x) = \frac{1}{p(x)\sum_{j=1}^{C}|D_j|} p(x|w_c)|D_c|. \tag{9}$$

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

$$p(x|w_c)|D_c| > p(x|w_j)|D_j|, \forall j \neq c. \tag{10}$$

We can also simplify the same decision rule by taking the natural logarithm of both sides of (10) in the following way: Decide w_c if

$$(x-\hat{\mu}_c)^T\hat{\Sigma}_c^{-1}(x-\hat{\mu}_c) - (x-\hat{\mu}_j)^T\hat{\Sigma}_j^{-1}(x-\hat{\mu}_j) < ln\frac{|D_c|^2|\hat{\Sigma}_j|}{|D_j|^2|\hat{\Sigma}_c|}, \forall j \neq c. \tag{11}$$

In order to evaluate the performance of the classification algorithm, precision and recall metrics are used. These metrics are given in Table I for both training (75%) and test (25%) separately.

The corresponding code for this section can be found in task1.m file.

TABLE I
METRICS FOR TRAINING AND TEST SETS WHEN MLE IS USED

Class	c1	c2	c3	c4	c5	c6	c7	c8
Tra Prec.	0.95	0.96	1.00	0.99	0.98	1.00	1.00	1.00
Tra Recall	1.00	0.97	1.00	0.94	1.00	1.00	1.00	1.00
Tst Prec.	1.00	0.79	0.82	0.75	0.59	0.67	0.53	0.69
Tst Recall	0.67	0.79	0.83	0.79	0.53	0.67	0.67	0.71

III. TASK 2: BAYESIAN ESTIMATION

In this part, we are trying to calculate the posterior probability $P(w_c|x)$ for each class c when μ_c of the likelihood $p(x|w_c) \sim \mathcal{N}(\mu_c, \Sigma_c)$ is a random variable. This random vector μ_c is initially assumed to be distributed by $p(\mu_c) \sim \mathcal{N}(\mu_{c,0}, \Sigma_{c,0})$. Here, $\mu_{c,0}$ is our prior guess about μ_c, and $\Sigma_{c,0} = \sigma_{c,0}\mathbf{I}$ is the uncertainty about our guess. In our implementations, we will test different values of $\sigma_{c,0}$ where our guess will taken as $\mu_{c,0} = \hat{\mu}_c$.

Since the prior probabilities $P(w_c)$ are known (as given in (5)), the Bayes formula can be written as:

$$P(w_c|x,D) = \frac{p(x|w_c,D_c)P(w_c)}{\sum_{j=1}^{C} p(x|w_j,D_j)P(w_j)}. \tag{12}$$

Also, the class conditional density $p(x|w_c,D_c) = p(x|D)$ and $p(x|D) \sim \mathcal{N}(\mu_{c,n}, \hat{\Sigma}_c + \Sigma_{c,n})$, where $\mu_{c,n}$ and $\Sigma_{c,n}$ are given as follows:

$$\begin{aligned} \mu_{c,n} &= \Sigma_{c,0}\left(\Sigma_{c,0} + \frac{1}{|D_c|}\hat{\Sigma}_c\right)^{-1}\hat{\mu}_c \\ &+ \frac{1}{|D_c|}\hat{\Sigma}_c\left(\Sigma_{c,0} + \frac{1}{|D_c|}\hat{\Sigma}_c\right)^{-1}\mu_{c,0}, \qquad (13) \\ \Sigma_{c,n} &= \Sigma_{c,0}\left(\Sigma_{c,0} + \frac{1}{|D_c|}\hat{\Sigma}_c\right)^{-1}\frac{1}{|D_c|}\hat{\Sigma}_c. \qquad (14) \end{aligned}$$

Here, $\hat{\mu}_c$ and $\hat{\Sigma}_c$ are defined in (4) and (5), respectively. By taking $\mu_{c,0} = \hat{\mu}_c$ for all c, (13) can be simplified as

$$\mu_{c,n} = \hat{\mu}_c. \tag{15}$$

Please note that $P(w_c|x,D)$ can be calculated by using (12), (14), and (15).

A. Decision Rule

After the estimations, we now can derive the decision rule. For a given sample x, by the Bayes Rule, we decide w_c if

$$P(w_c|x,D) > P(w_j|x,D), \forall j \neq c. \tag{16}$$

Since the denominator of (12) remains the same for all c, our decision rule becomes: Decide w_c if

$$p(x|w_c,D_c)|D_c| > p(x|w_j,D_j)|D_j|, \forall j \neq c. \tag{17}$$

By taking the natural logarithms of both sides, (17) can be further simplified as: Decide w_c if

$$(x-\mu_{c,n})^T\bar{\Sigma}_c^{-1}(x-\mu_{c,n}) - (x-\mu_{j,n})^T\bar{\Sigma}_j^{-1}(x-\mu_{j,n}) < ln\frac{|D_c|^2|\bar{\Sigma}_j|}{|D_j|^2|\bar{\Sigma}_c|} \tag{18}$$

is satisfied for all $j \neq c$. Here, $\bar{\Sigma}_c = \hat{\Sigma}_c + \Sigma_{c,n}$ can be found by adding (4) and (14).

In order to evaluate the performance of the classification algorithm, precision and recall metrics are used. We evaluated the algorithms for different values of μ_0 where $\mu_{i,0} = \mu_{j,0} = \mu_0$. The results are given in Table II for both training (75%) and test (25%).

TABLE II
METRICS FOR $\sigma_0 = 0.1, 10, 10000$ (SAME TABLES ARE OBTAINED FOR ALL)

Class	c1	c2	c3	c4	c5	c6	c7	c8
Tra. Prec.	0.95	0.96	1.00	0.99	0.98	1.00	1.00	1.00
Tra. Recall	1.00	0.97	1.00	0.94	1.00	1.00	1.00	1.00
Tst. Prec.	1.00	0.79	0.82	0.75	0.59	0.67	0.53	0.69
Tst. Recall	0.67	0.79	0.83	0.79	0.53	0.67	0.67	0.71

The corresponding code for this section can be found in task2.m file.

IV. COMPARISON OF ESTIMATORS

As it can be seen from Tables I and II, the performance metrics for both estimation methods are exactly the same. I was expecting this result. Since we fixed $\mu_{c,0}$ values (our prior guess about parameter distribution) equal to $\hat{\mu}_c$, we turned (13) into (15). Which means we force density $p(x|w_c,D_c)$ to always have the same mean of $\hat{\mu}_c$. Even though its variance is changing, our decision rule always gives the same result because of the Gaussian distribution.

If you can check "*bar_covariance_matrices*" ($\bar{\Sigma}_c$) and "*covariance_matrices*" ($\hat{\Sigma}_c$) variables in the task2.m file, you will actually see that they are different matrices. However, by using these different matrices in the same decision rules ((11) and (18)) with the same mean vectors gives us the same results.