Homework Assignment 6: Bayesian Classifier

Due Saturday, March 26, 2022 at 11:59 pm EST

Description and dataset instructions

In class, we studied Bayesian approaches for classification. In this assignment, we are going to experiment with two different approaches: Naïve-Bayes classifier and minimum distance classifier based on the Mahalanobis distance.

For this assignment, we are going to use the Pima Indians Diabetes problem dataset. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes (outcome = 0 or 1), based on certain features. The feature (predictors) set includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

Carefully read the dataset description (link). This dataset consists of 768 entries (rows); download the dataset to the **input** directory then write a function/script that **randomly** picks 540 entries for training set and the reaming 228 entries for testing. Chose proper naming for your variables; e.g., X_train, X_test, Y_train, and Y_test (the vector with the ground truth labels of your testing set).

What to submit

Create a folder: `ps6_LastName_FirstName` and add in your solutions:

`Ps6_xxxx_LastName_FirstName/`

- input/ - input images, videos or other data supplied with the problem set
- output/ - directory containing output images and other files your code generates
- ps6.m - code for completing each part, esp. function calls; all functions themselves must be defined in individual function files with filename same as function name, as indicated
- *.m Matlab/Octave function files (one function per file), or any utility code
- ps6_LastName_FirstName_debugging.m – one m-file that has all of your codes from all the files you wrote for this assignment. It should be a concatenation of your main script and all of your functions in one file (simply copy all the codes and paste them in this file). In fact, this file in itself can be executed and you can regenerate all of your outputs using it.
- ps6_report.pdf - a PDF file with all output images and text responses

Zip it as `ps6_LastName_FirstName.zip`, and submit on Canvas.

Guidelines

1. Include all the required images in the report to avoid penalty.
2. Include all the textual responses, outputs and data structure values (if asked) in the report.
3. Make sure you submit the correct (and working) version of the code.
4. Include your name and ID on the report.
5. Comment your code appropriately.
6. Please avoid late submission. Late submission is not acceptable.
7. Plagiarism is prohibited as outlined in the Pitt Guidelines on Academic Integrity.

Questions

1. **Naïve-Bayes classifier**

The Naïve-Bayes classifier (discussed in section 2.5.7; see attached handout) is one of the most simple and robust classifiers. In this problem, you are asked to build a Naïve-Bayes classifier and test its performance on real problem (Pima Indians Diabetes problem). The main assumption of a Naïve-Bayes classifier is that all features are statistically independent. Therefore,

$$p(\boldsymbol{x}|\omega_k) = \prod p(x_j|\omega_k) \qquad (1)$$

where $\boldsymbol{x} = [x_1, \dots, x_j, \dots, x_n]^T$ is the feature vector and x_j is the j^{th} feature, and ω_k is the k^{th} class.

In this part, we will assume that all the features are statistically independent and all of them follow a Gaussian distribution. Therefore, we need to, separately, compute the mean and standard deviation for each feature given a particular class ω_k. Thus, in this problem, we are going to assume that each feature $(x_j\,;\, j = 1\ to\ n)$ belonging to class k has a Gaussian distribution with mean $\mu_{j,k}$ and standard deviation $\sigma_{j,k}$.

a. Separate the training matrix instances by class label so that we can calculate statistics for each class. Now, since we have only two classes, you should have two training matrices X_train_0 and X_train_1 that containin the training samples belonging to class 0 and class 1, respectively.

Output: (`textual response`):
- Size of X_train_0 and X_train_1.

b. For each class, calculate the mean and the standard deviation of each feature (i.e., column). After finishing this step, you should now have 16 values for the mean (8 for the feature values belonging to class 0 and 8 for the values in class 1) and 16 values for the standard deviation. Therefore, you can calculate $p(x_j|\omega_0)$ and $p(x_j|\omega_1)$ using the normal distribution equation for every feature $x_j; j = 1, 2, \dots, 8$. For example:

$$p(x_j|\omega_0) = \frac{1}{\sqrt{2\pi}\ \sigma_{j,0}} \exp\left(-\frac{(x_j - \mu_{j,0})^2}{2\,\sigma_{j,0}^2}\right)$$

Output: (`textual response`):
- A table that list the values of mean and standard deviation fro each feature and each class.

c. Assume that $p(\omega_0) = 0.65$ and $p(\omega_1) = 0.35$. For each entry in the testing set, make a prediction:
   I. For each feature j, calculate $p(x_j|\omega_0)$ and $p(x_j|\omega_1)$.
   II. Use equation (1) to calculate $p(\boldsymbol{x}|\omega_0)$ and $p(\boldsymbol{x}|\omega_1)$.
   III. Compute the posterior probabilities for classes 0 and 1:
   $p(\omega_k|\boldsymbol{x}) \approx p(\boldsymbol{x}|\omega_k)p(\omega_k)$.
   IV. Assign this testing entry to class 0 if $p(\omega_0|\boldsymbol{x}) \geq p(\omega_1|\boldsymbol{x})$, or to class 1 otherwise.

V. Compare your prediction to the actual class value found in Y_test.

Output: (`textual response`):
- Report the accuracy of your classifier

2. Minimum distance classifier

In this part we investigate the performance of the minimal distance classifier under the assumption of a non-diagonal covariance matrix. Thus, we need to compute the Mahalanobis distance between the testing feature vector and each class mean vector. The testing vector gets the label of the closest class.

a. Get an estimate of the covariance matrix C by using this MATLAB instruction: `C = cov(X_train).`

Output:
- Size of the covariance matrix
- A snapshot of your covariance matrix

b. The mean vector for class k is simply $\mu_k = [\mu_{1,k}, \mu_{2,k}, \cdots, \mu_{8,k}]^T$. Use the values of the mean you obtained in part 1.b or the MATLAB `mean` function; compute the vectors μ_0 and μ_1; the mean vector for class 0 and class 1, respectively.

c. Assume that $p(\omega_0) = 0.65$ and $p(\omega_1) = 0.35$. For each entry in the testing set, make a prediction:

i. Compute d_0 and d_1; the Mahalanobis distance between this feature vector and μ_0 and μ_1, respectively. Use the equation in Lecture 14, slide 31.
ii. Also, to account for the fact the two classes are not equally probable, you need to subtract twice the natural log of class probability from the corresponding distance. Get d_0' and d_1', the modified distances after accounting for the class probabilities.
iii. Assign this testing entry to class 0 if $d_0' < d_1'$, or to class 1 otherwise.
iv. Compare your prediction to the actual class value found in Y_test, and compute your classifier's accuracy.

Output (`textual response`):
- Report the accuracy of your classifier
- Compare between the naïve classifier and the Mahalanobis classifier.