ECE8560 Spring 2019
Takehome #1

(Canvas submission only)
Assigned 1/31/2019
Due 2/28/2019 11:59 PM.
Late policy:
Solution is graded if received by the deadline.

Contents

1 Overview

This is a comprehensive takehome which is to be done **individually**. You may choose any programming language for the solution, however, **'canned' functions and/or libraries which directly implement classifiers, or parts of classifiers are prohibited.**

The objective is to design and test a Bayesian, minimum error classifier using given training and test sets (H and S_T). We may re-use this data for subsequent takehome activities.

Key remarks are:

1. Do all parts of the assignment. There are 2 cases, each with multiple parts, deliverables and questions. Use this document as a checklist for your submission.

2. Open book and notes, but **no collaboration**. (This is an individual effort.)

3. **Submit to Canvas by the deadline.**

4. **Clarity of the presentation (in addition to technical correctness) counts significantly**.

2 Problem and Given Training and Testing Data (Files)

This is a $c = 3$ class problem. The $d = 4$ training and test data is available on the course Canvas Assignment page in 4 *.dat files: `train_case1`, `test_case1`, `train_case2` and `test_case2`.

2.1 Training Data Format

Each training data file consists of 15,000 $d = 4$ dimensional vectors in physical coordinates, **with one feature vector (transposed) per line**. The first 5000 samples correspond to class w_1 (H_1), the second 5000 samples correspond to class w_2 (H_2) and the third 5000 samples correspond to class w_3 (H_3).

2.2 Test Data Format

Using `test_case_i` (i = 1 or 2) data, the objective is to classify each sample using your resulting Bayesian classifier design (based upon the training data and your model). Note that, at this time, you do not know the correct class for each vector in the `test_case_i` files. **Note: Do not do any training with the test data.**

3 Classification Cases

1. Case 1: $P(w_i) = 1/c$. Uses data in `train_case_1` and `test_case_1`.
2. Case 2: $P(w_1) = 1/2; P(w_2) = 1/3; P(w_3) = 1/6$. Uses data in `train_case_2`, `test_case_2`.

4 Bayesian Classifier Design and Implementation

The objective is to develop a Bayesian classifier for this data which minimizes classification error. I am interested in both your engineering judgment as well as the appropriateness and performance of your resulting classifier. You will need to determine (and justify) the appropriate statistical models, including apriori probabilities and pdfs for each class. **In this assignment, simplifying assumptions are not desired**. In other words, make the most from what you are given in designing the most accurate classifier.

5 Specific Results Desired for Each Case

5.1 Testing and Reporting Classification Results for Each Case

5.1.1 Resulting Classification File

In order for me to determine the classification error in your work, for each case you need to classify each feature vector in `test_case_i`. For future use, you will classify all 15000 test vectors in each `test_i` file sequentially. Since `test_case_i` contains one (transposed) feature vector per line, this is done by writing the class (from your Bayesian classifier) to a corresponding file named

`takehome1_case_i.txt`

with one integer per corresponding line. The desired format of this part of your answer is therefore an ASCII (text) file[1] with one ASCII integer entry, x, per line, where

$$x = \begin{cases} 1 & \text{if sample i is classified as class } w_1 \\ 2 & \text{if sample i is classified as class } w_2 \\ 3 & \text{if sample i is classified as class } w_3 \end{cases}$$

Please name this text file as indicated and include it in the zipped archive you submit for each case.

5.1.2 Display of Partial Classification Results in Report

To enable me to quickly check the quality of your classifier, for each case, you must **show your classification results for the first 30 samples of each test set in your report**, in the text form shown below:

```
Case i:
for sample 1 class is x
for sample 2 class is x
for sample 3 class is x
for sample 4 class is x
for sample 5 class is x
for sample 6 class is x
for sample 7 class is x
for sample 8 class is x
for sample 9 class is x
for sample 10 class is x
for sample 11 class is x
for sample 12 class is x
for sample 13 class is x
for sample 14 class is x
for sample 15 class is x
for sample 16 class is x
for sample 17 class is x
for sample 18 class is x
```

[1] The reason for this format is to enable me to judge your classifier.

```
for sample 19 class is x
for sample 20 class is x
for sample 21 class is x
for sample 22 class is x
for sample 23 class is x
for sample 24 class is x
for sample 25 class is x
for sample 26 class is x
for sample 27 class is x
for sample 28 class is x
for sample 29 class is x
for sample 30 class is x
```

Make sure this is included prominently in your report on page 4.

5.2 Reality Check and Question #1 For each Case

Training data may be used to check the reasonableness of your classifier for each case. When you are done designing the classifier, classify each sample in the training set file, `train_case_i`, and estimate P(Error) using this result. This would also be a good reality check on the quality of your classifier (or the difficulty of the problem, or both).

Question #1: How well does the classifier classify the training data?
My expectation is that your answer will be more elaborate than 'Yes' or 'No'.

5.3 Reality Check and Question #2 For Each Case

For each case, you can also use testing results to check for a class distribution of your classification results, and therefore determine if this supports the given apriori class probabilities (see Section 3).

Question #2: Does the classifier, used on the given test data, produce a distribution of classes consistent with the prespecified apriori probabilities?
My expectation is that your answer will be more elaborate than 'Yes' or 'No'.

6 Format of the Report Results

This aspect is critical. If you make it difficult for me to assess your effort, I won't. The final report must be in your solution archive in a PDF format file named `takehome1.pdf`. The report results **must be in the following order**:

1. p. 0: Title page (<name>, <CU username>, ECE 8560, Takehome #1)
2. p.1-2: Indicate your **engineering decisions and associated rationale**, e.g., density function form, parameter estimation, $P(w_i)$, design of the classifier, etc.
3. p. 3: Show the exact form of the discriminant function used for each class in each case.
4. p. 4: Show the classification results for the first 30 samples of each test set in the form indicated in Section 5.1.2 and answers to the 2 specific questions asked for each case.
5. p. 5: Estimate your P(Error), using the **training data with known class** for each case.
6. Pages beyond 5: Anything else you feel is relevant.

7 Additional Notes and Constraints

7.1 Format of the Electronic Submission

The final **zipped archive** is to be named **`<yourname>`-ece8560-takehome1.zip**, **where `<yourname>` is your (CU) assigned user name**. You must upload this to the ECE8560 Canvas page **prior to the deadline for your solution to be considered**.

The minimal contents of this archive are described below.

1. Include a `readme.txt` file listing the contents of the archive and a brief description of each file. Include 'the pledge' here. Here's the pledge:

   > **Pledge:**
   > On my honor I have neither given nor received aid on this exam.

2. Put all results in the single, top-level directory.

3. All documentation should be in a single pdf file named `takehome1.pdf` with the structure indicated in Section 6. **No MS Word (doc) files.**

4. Include all (your) source code used in your simulations.

5. Include the classifier result text files as specified.