Program Assignment #1

- Due Date: Nov. 12 (Mon) 23:59
- Submission: Source code & summary report including the description on your program (within 4 pages)

Consider 3 classes W_1, W_2, $W_{3,}$ in the four-dimensional feature space. Each class data has the distribution as follows:

$$W_1 \sim \sum_{k=1}^{N_1} w_{1,k} N(\mu_{1,k}, \Sigma_{1,k})$$
$$W_2 \sim \sum_{k=1}^{N_2} w_{2,k} N(\mu_{2,k}, \Sigma_{2,k})$$
$$W_3 \sim \sum_{k=1}^{N_3} w_{3,k} N(\mu_{3,k}, \Sigma_{3,k})$$

- $N(\mu, \Sigma)$ represents Multivariate Normal Distribution (Multimodal Gaussian Mixture)
- The sum of w_k in each class is 1.
- We will not notice the values of N_1, N_2, N_3.
    - Choose the format of data according to your preference: .txt or .mat
    - Ex) load('W1_train.mat') ➔ Variable W1_train will be generated in MATLAB workspace

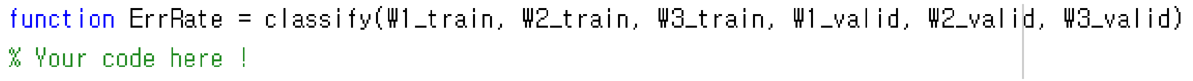
- Training data: 300 samples for each class
- Valid data: 300 samples for each class
    - Valid data is provided to verify the performance of your implemented algorithms. It is not used in actual grading.

Training data include some noise data that are not following the given probability distribution. The number and the distribution of the noise data is different from each class. So, the finest tuning of your algorithm with the given training data would not guarantee the good results with the TA's performance evaluation data. Submit the highest performance algorithm (source code) by implementing classifier using Bayesian, Maximum Likelihood, N-N, k-N-N algorithms, etc., that are discussed in our class.

<Precautions>

1. (Important) Submit a MATLAB function file with the following input and output. If not, the score will be deducted.

   Example)

   ```
   function ErrRate = classify(₩1_train, ₩2_train, ₩3_train, ₩1_valid, ₩2_valid, ₩3_valid)
   % Your code here !
   ```

   Input: (W1_train, W2_train, W3_train, W1_test, W2_test, W3_test)

   - ✓ Input matrix: The total number of row is the number of samples (=300), and the total number of column is the number of the dimensions (=4).
   - ✓ When choosing the text format data, you should read and save the data in a 300 x 4 array, and then use it as input to the function.

   Output: $Error\ \ Rate = \frac{E_{total}}{N_{test}} = \frac{(E_{w1}+E_{w2}+E_{w3})}{N_{test}}$

   - ✓ N_{test} : The total number of valid data (=900).
   - ✓ E_{w1} : The number of valid samples that are belonged to class W$_{1,}$ but classified to class W$_{2}$, or W$_{3.}$
   - ✓ E_{w2} : The number of valid samples that are belonged to class W$_{2,}$ but classified to class W$_{1}$, or W$_{3.}$
   - ✓ E_{w3} : The number of valid samples that are belonged to class W$_{3.,}$ but classified to class W$_{1}$, or W$_{2}$

   Function name: classify(…)

   - ✓ classify.m

2. File name: ID_name.zip (Source code and report should be compressed in zip format)

   예) 20180000_HeejinChoi.zip

3. Submit only one of the best-performing algorithm!!!

   - ✓ **It is impossible to output the best performance after testing with various algorithms in 'classify.m' function.**
   - ✓ For the k-N-N algorithm, you should use the fixed *k* value.

4. Implement your algorithm using MATLAB. Do not use other programming languages.

5. Test data is not provided to you. TA will evaluate final performance of your algorithms using the test data. ➔ To avoid overfitting & overtuning.

6. Scoring method: Evaluate the error rate using test data by running your implemented 'classify' function.

7. PA1 score = 100 – (Performance rank X 1 – 1)

   - ✓ ex) First place: 100 points / Second place: 99 points / Third place: 98 points / …

Good Luck!