AAAI 5001 Machine Learning and Pattern Recognition
Homework #1: Gaussian mixture model (GMM) classifier
27 April 2020
Due date: 10 May 2020

Zero-tolerence policy

Do not share your codes and reports with the other students. Do not try to refer to the other students' codes and reports.

Items to be submitted to YSCEC

1) Report in pdf (hw1_honggildong.pdf)
2) Single zip file (hw1_honggildong.zip) containing your source code
(Your final submitted code must be run without any problem. Check this in advance. Specify your programming environment (language, version, library, etc.) in your report.)

Problem description

In this homework, the data sets of two artificial classification problems will be used. Each data set is for a 2-class classification problem and consists of N training data and N test data, where N=225 for problem 1 and N=220 for problem 2. The input data are within a range of $[-1,1]^2$, and the target value is either 0 or 1, meaning class 1 or class 2, respectively. Four data files are provided for each problem as follows (where ? is equal to 1 or 2):

- p?_train_input.txt: training samples (Nx2 matrix)
- p?_train_target.txt: target category (0 or 1) for each training sample (Nx1 vector)
- p?_test_input.txt: test samples (Nx2 matrix)
- p?_test_target.txt: target category for each test sample (Nx1 vector)

Coding environment

You can choose which language to use, but Python is recommended. In the case of Python, you will need to use the scikit-learn library, in particular, refer to the *GaussianMixture* object.

Experiments

Write your code for training and testing a GMM classifier. Try different structures of GMMs, including the number of Gaussian components and the type of covariance matrices (spherical, diagonal, and full), and observe how the performance changes.

Your objective is not to simply write code to train and test GMMs, but to develop your own research questions and conduct experiments to answer the questions. Report your results with thorough discussion. Some example questions are (but not limited to):

- How does the choice of the number of Gaussian components influence the performance?
- Which is better among different types of covariance matrices? In which sense?
- Does the performance vary according to the initialization of the model parameters?
- What is a good strategy to initialize the model parameters?
- What is a good strategy to determine the convergence of the EM algorithm?
- Does overfitting occur?

Some tips are:

- The primary measure for performance will be the test accuracy. However, the training time and run time can be also considered.
- Since the input data are 2-dimensional, you can easily visualize the given data and results, such as the decision boundary, the final locations and shapes of the Gaussian components after training, the process that the Gaussian components 'move' in the input space during training, etc.