MSC-BDT5002/MSC-IT 5210 Knowledge Discovery and Data Mining, Fall 2017

Assignment 3

Deadline: Dec. 1st, 11:59pm, 2017

Submission Guidelines

- Assignments should be submitted to mscbdt5002fall17@gmail.com (for BDT students)/ mscit5210fall17@gmail.com (for IT students) as attachments.
- You need to zip the following three files together:
  1. A3_itsc_stuid_answer.pdf/A3_itsc_stuid_answer.docx: Please put all your answers in this file including the readme pages for Q1 and output answers for Q2 & Q3.
  2. A3_itsc_stuid_code.zip: The zip file contains all your source codes for Q1, Q2 and Q3.
  3. A3_itsc_stuid_Q1prediction.csv: The prediction result for Q1.
  4. You don't need to submit the dataset.
- **Attachments should be named in the format of**: Ax_itsc_stuid.zip. E.g. for a student with **itsc account**: zxuav, student id: 20171234, the 3rd assignment should be named as: A3_zxuav_20171234.zip.
- **Submissions after the deadline or not following the rules above are NOT accepted.**
- You can use any programming language. In principle, python is preferred.
- Your grade will be based on the correctness, efficiency and clarity.
- Plagiarism will lead to zero mark.

(please read the guidelines carefully)

Attention

1. TA will check your source code carefully, so your code must be runable. Keep your code clean and comment it clearly.
2. Cheating is not allowed. Your result **MUST** be reproducible.

1. Classification Task (60 marks)

Clickstream is important for mining user's latent behavior. As for online learning platform, the lectuerer can monitor students' learning pattern by clickstream pattern analysis. Video-clickstream records students' click actions when watching lecture videos. A general video-clickstream log file contains the following events: *load_video*, *play_video*, *pause_video*, *seek_video*, *stop_video*, and *speed_change_video*.

In this part, you need to predict students' final exam performance, passed or failed, based on their clickstream event log from 63 lecture videos in the same online learning semester.

The dataset including:

1. TrainFeatures.csv: 5050 students' video clickstream log info covering 63 videos. You need to use it for training.
2. TestFeatures.csv: 1293 students' video clickstream log info covering 63 videos. You need to use it in the testing stage. The students in **TrainFeatures.csv** and **TestFeatures.csv** belong to the same learning semester and share the same grading strategy.
3. TrainLabel.csv: The label for 5050 students (1 for pass, 0 for fail).
4. TestData.csv: The students you need to give prediction for. Their learning log info can be found in TestFeatures.csv.
5. VideoInfo.csv: Video duration info for 63 lecture videos.
6. Sample_submission.csv: The sample submission file you may refer.
7. Description.pdf: Some description for the log events and attributes.

You need to submit:

1. Your test result in a csv file with the same format as **Sample_submission.csv**. Please name it as **A3_itsc_stuid_Q1prediction.csv**:
2. Your source code **A3_itsc_stuid_Q1.xxx** in a zip file named as **A3_itsc_stuid_code.zip**,
3. 1-2 page readme in **A3_itsc_stuid_answer.pdf** which illustrates:

   a. Your training environment, e.g. if you use python, please report your python version and the packages you use;

   b. Briefly introduce how do you engineer features;

   c. Briefly introduce which model you use to get the prediction;

   d. Your reference.

Notes:

1. You can use any classification algorithm you have learned.
2. Real-world data contains noise, missing values or even mistakes. Data cleaning and pre-processing are necessary.
3. Feature engineering is important, you need to generate features on your own.
4. Reasonable pre-processing and post-processing are allowed.
5. Your assignment will be graded by the testing accuracy.
6. If you borrow any idea from the internet or from your friends, please cite it in your readme.

Fuzzy clustering with EM algorithm (20 marks)

Based on the clickstream event frequency pattern in **Q2Q3_input.csv** for a given lecture video, apply EM algorithm to cluster the students into two classes with the following initial settings:

Initial centers: c1 =(1,1,1,1,1,1), c2 = (0,0,0,0,0,0)

Cluster features: frequency patterns for 6 given clickstream events: load_video,pause_video, play_video,

seek_video, speed_change_video and stop_video, you can find them in **Q2Q3_input.csv**.

You need to:

(a). Report the updated centers and SSE for the first two iterations.

(b). Report the overall iteration step when your algorithm terminates

(c). Report the final converged centers for each cluster.

You need to submit:

1. Your source code **A3_itsc_stuid_Q2.xxx** in a zip file named as **A3_itsc_stuid_code.zip**, and
2. Report your result for (a)(b)(c) in **A3_itsc_stuid_answer.pdf**.

Notes:

1. Please use the terminate condition below:

   Terminate condition: the EM algorithm will terminate when:

   1). The sum of L1-distance for each pair of old-new center

   $$\sum_{each\ center} ||C_{old} - C_{new}||_1$$

   is smaller than 0.001, or

   2). The iteration step is greater than the maxinum iteration step 50.

2. You **MUST** code by yourself to complete the EM algorithm.

Outlier detection with LOF (20 marks)

Based on the clickstream event frequency pattern in **Q2Q3_input.csv**, apply LOF algorithm to calculate LOF for each point with the following initial settings:

1. Set k = 2 and use Manhattan distance.
2. Set k = 3 and use Euclidean distance.

You need to:

1. Report top 5 outliers and their $LOF_k(o)$ for 1 & 2 in **A3_itsc_stuid_answer.pdf**.
2. Submit your source code **A3_itsc_stuid_Q3.xxx** in a zip file named as **A3_itsc_stuid_code.zip**.

Notes:

1. Use the corrected formula in the lecture notes.
2. You **MUST** code by yourself to complete the LOF algorithm. e.g. You can only use the basic packages in Python, like numpy, scipy, math, and pandas. **Using sklearn is not allowed for Q3**.

Data Description for Q1

TrainFeatures.csv & TestFeatures.csv

user_id: Identifies the individual who is performing the action.

session: This 32-character value is a key that identifies the user's session. All browser events include a value for the session. Other mobile events do not include a session value.

load_video: This tag appears when the video is rendered and ready to play.

play_video: This tag appears when a user selects the video player's play control.

pause_video: This tag appears when a user select the video player's pause control.

seek_video: This tag appears when a user selects a user interface control to go to a different point in the video file.

stop_video: This tag appears when the video player reaches the end of the video file and play automatically stops.

speed_change_video: This tag appears when a user selects a different playing speed for the video.

event_time: The time that this event occurs. Gives the UTC time at which the event was emitted in 'YYYY-MM-DDThh:mm:ss.xxxxxx' format.

new_time: The time in the video, in seconds, that the user selected as the destination point. This filed appears for *seek_video* action only.

old_time: The time in the video, in seconds, at which the user chose to go to a different point in the file. This filed appears for *seek_video* action only.

old_speed: The speed at which the video was playing. This filed appears for *speed_change_video* action only.

new_speed: The speed that the user selected for the video to play. This filed appears for *speed_change_video* action only.

TrainLabel.csv

user_id: Identifies the individual in the training part.

grade: Final performance status, 0 for not pass and 1 for pass

TestData.csv

user_id: Identifies the individual in the testing part.

VideoInfo.csv

video_id: Identifies the lecture videos appearing in clickstream

duration: The duration in seconds.

Sample_submission.csv

user_id: The users you need to give prediction for, same as the user_id in **TestData.csv**

grade: The sample prediction.