Final Project Spec

TA email: `ml2015ta@csie.ntu.edu.tw`

1 Problem Description

In this project, you are going to play with data from a Massive Open Online Course (MOOC) platform, which contains information of many courses and students. Your goal is to predict whether a student will drop a course that she/he enrolled in. We provide you all the logs of each enrollment within the first 30 days of the course. Then, if the student has no logs within the following **10 days**, i.e. the 31st-40th days from the start date of the course, we label him/her as a dropout. (Note that the 31st day starts from 00:00:00).

2 Data Description

There are several files which contain all the information we have about these logs:

- `enrollment_train/test.csv`: match the enrollment_id to student and course.
  - enrollment_id: enrollment ID
  - username: student ID
  - course_id: course ID
- `log_train/test.csv`: Logs for each enrollment.
  - enrollment_id: enrollment ID
  - time: the time of the event
  - event_source: event source (server or browser)
  - event_type: the type of the event
    - problem: operations on course problems
    - video: operation on course videos
    - access: accessing other courseware objects
    - wiki: accessing the course wiki
    - discussion: accessing the course forum
    - navigate: navigating to other part of the course
    - page_close: close the web page
  - object: the object related to the event (see `object.csv`)
- `object.csv`: Contain information about courses. Each course is represented as a tree of modules. For instance, a course contains multiple chapter modules, a chapter contains sequentials, and a sequential contains verticals and videos.
  - course_id: the course to which the module belongs
  - module_id: the ID of a courseware module
  - category: the category of the courseware module
  - children: the children modules' id of the courseware module
  - start: the time that the module was released to students
- `sampleSubmission.csv`: The required submission file should be a 24109×2 matrix, with no header or other information, like this file. The first column should be the enrollment ID, and the second column is your prediction (float or 0/1). The two columns should be split by a comma. An error will be reported if a submission file is of a wrong format.

For your convenience, the TAs have kindly provided you some basic features which can be directly used in training your models. The features are extracted from within `log_train.csv`. But of course, the basic features are not what you should be satisfied with—feature engineering is also an important issue when solving real world problems. Thus, you are highly encouraged to conduct your own feature extraction in order to get better performance.

- `sample_train/test_x.csv`
  - ID: enrollment ID
  - user_log_num: total number of logs of the user (student) in all the courses
  - course_log_num: total number of logs belongs to the course
  - take_course_num: number of courses the user takes
  - take_user_num: number of users who take the course
  - log_num: number of logs belongs to the enrollment
  - (event_source)_(event_type): 9 dimensions, number of logs with different event_sources and event_types (refer to log_train/test.csv)
  - (chapter/sequentail/video)_count: 3 dimensions, number of logs with certain objects

3 Evaluation

We will have two tracks of competition, each evaluated with a different goodness measure.

- Track 1: Mean Average Precision (MAP)

  Mean Average Precision is a popular evaluation criterion in many ranking problems. Your hypothesis g should provide the **probability estimate** that each enrollment would be a dropout. Then, we will take the average precision of your top-M estimates (with ties mysteriously broken). That is,

  $$\text{MAP}(g) = \frac{1}{M}\sum_{n=1}^{M}\frac{dropout_n(g)}{n}, \text{ where}$$
  $$dropout_n(g) = \text{number of dropouts in top } n \text{ enrollments}$$

  We will specify M later.

- Track 2: Weighted Accuracy

  Several students take multiple courses on the platform. Thus, we have more information about their behavior, which makes it easier to predict whether they will drop a course. We decide to make it **harder** for you by re-weighting those enrollments to be of less importance. Your hypothesis g should provide the **binary** (0 or 1) prediction on whether each enrollment will be a dropout. Then, we will take

  $$\text{WeightedAccuracy}(g) = \frac{\sum_{n=1}^{N} c_n^{-1} \times [\![g(n^{th} \text{ enrollment}) = y_n]\!]}{\sum_{n=1}^{N} c_n^{-1}}, \text{ where}$$
  $$c_n = \text{total number of courses the student of the } n^{th}\text{enrollment takes}$$