Final Specification

Members:

Peter (Gyu Young) Chang : chang04@college.harvard.edu

Ann Hwang : annhwang@college.harvard.edu

Zachry Bai : zachrybai@college.harvard.edu

Dor Baruch : dorbaruch@college.harvard.edu

Detailed Description

(1) Data extraction -

- We are going to use the offline imdb database that includes the following information about each movie: cast list, directors, producers, composers, distributer, genre, MPAA rating, budget, release month, runtime.

  The database can be found in: ftp://ftp.fu-berlin.de/pub/misc/movies/database/

  Following the guide in http://imdbpy.sourceforge.net/docs/README.sqldb.txt we will load the information to a MySQL database.

  The data will later be extracted from the database using the MySQLdb module in Python and filtered to only include the relevant data from imdb (only movies from the US that include user rating).

(2) Supervised Learning Algorithm

(i) Random Forest Algorithm

- The libraries (Python) we will use for this implementation are numpy and scipy.
- The program inputs a .csv file that represents a (5,000 X 11) matrix that describes the following information about each films: cast list, directors, producers, composers, distributers, genre, MPAA rating, budget, release month, runtime.
- We will have one function that creates one decision tree that will be trained using the training subset of our data set.
- Our next function will aggregate the decision trees and take a vote on the individual trees.

- The final function will be a validation function, one that checks the last column of the output csv with the true value for the ratings and get a number for the accuracy.

(ii) Support Vector Regression

- As a second supervised regression algorithm, we will implement the “soft” support vector regression algorithm with a slack variable E.
- The libraries (Python) we will use for this implementation are numpy, scipy, and cvxopt.
- The input should also be the .csv file that we extracted using database parser and a value for the slack variable to specify the “softness” of the algorithm.
- Dual problem derivation: assuming strong duality, a function in our program will transform the original “soft” constrained optimization problem to a dual problem.
- Quadratic programming: the second function in our program will take in the transformed dual constrained optimization problem and use the CVXOPT library to solve the quadratic program.
- The final function will then tell the trained algorithm to take in the test subset of the data and output a .csv file with the prediction for the ratings in the last column.
- We will use the validation function we wrote for the random forest algorithm to validate our prediction and get a number for the accuracy.

Timeline

Week 1 (April 11 ~ April 17)

- Finalize the specification
- Start implementing the interfaces for the random forest algorithm, and support vector regression

Week 2 (April 18 ~ April 24)

- Extract and parse the relevant information for the imdb database
- Separate the data into training and test data
- Finish the implementation of random forest algorithm and check the trained forest’s accuracy against the test data

Week 3 (April 25 ~ May 1)

- Finish the support vector regression
- Check the accuracy of support vector regression algorithm against the test data
- Compare the accuracies of the two supervised learning algorithms
- Polish and make our project more presentable (implement a basic user interface)

Progress Report

- We started implementing raw interfaces for the random forest and support vector machine.
- We created a GitHub repository for version control.