CSE 482 Class Project (Fall 2017)

You may choose one of the topics below for the project. If you have other suggestions, you are welcome to discuss it with the instructor.

A. Development of a recommender system Web site

There are many user ratings data sets available online. This includes:

a. MovieLens movie ratings (https://grouplens.org/datasets/movielens/)
b. lastFM million song dataset (http://labrosa.ee.columbia.edu/millionsong/lastfm)
c. Other data sets (https://gist.github.com/entaroadun/1653794)

In addition to the ratings data, you should try to incorporate other information as well to improve the recommendation (e.g., in movie recommendation, add information about movie genre, actors, directors, etc). Below, is a summary of the tasks you need to perform to complete the project:

1. Apply user-based and item-based recommendation algorithm as follows.
    - For each user, calculate its top-k nearest neighbors (i.e., other users who share the most similar item preferences). Use the weighted average ratings of the neighbors to estimate whether the user likes an item he/she has not rated.
    - For each item, calculate its top-k nearest neighbors (i.e., other items whose ratings are most correlated to it). Use the weighted average ratings of the user on the most similar items to estimate whether the user likes an item he/she has not rated.
    - The similarity between every pair of users/items should be computed using the Hadoop framework.
2. Create training and test sets from the data. Compare the performance of the two approaches described above in terms of their accuracy on the test set. You are free to consider other approaches as well (e.g., Mahout's recommender system).
3. Develop a web-based interface that provides the following functionalities:
    - Allows a user to login and logout from the system.
    - Displays items the user has rated.
    - Provide recommendation to items the user has not rated.
    - Provide recommendation of other users who share similar preference.

B. Event Detection and Summarization from Twitter Data

The goal of this project is to use Twitter data for event detection and summarization. For example, you may use Twitter to detect disease outbreaks, natural disasters such as earthquakes, wildfires, tornados, etc. To do this, you have to use the Twitter streaming API to monitor tweets that contain a set of keywords about the event. The data collection should take at least 6-8 weeks to make sure you have enough data to do your analysis. Note that the size of data can be very large depending on the keywords you use. If possible, you should limit the data collection to those tweets originating from the United States only. You should make the set of keywords as comprehensive as possible to make sure you don't miss anything. The keywords alone are insufficient as they may produce significant false alarm (i.e., tweets unrelated to the actual event). Thus, in addition to keyword search, you should train a classifier that can distinguish between tweets related to the event and those unrelated to the event. To build such a classifier, you will

need to preprocess the tweets (e.g., converting the tweets into lower case, remove stopwords, perform stemming, etc). You may use the NLTK (http://www.nltk.org/index.html) python library to do many of the text preprocessing. Once you have identified the tweets, the next step is to summarize them using cluster analysis. For visualization, you would need to draw a map that shows locations of tweets belonging to each cluster. You may use the Google Map API to do the plotting. The visualization should also display the tweets that were assigned to the cluster.

C. Sentiment Analysis from Twitter Data

The goal of this project is to identify trends in user sentiment on a specific topic and to monitor their changes over time. You need to use the Twitter streaming API to monitor a set of keywords that capture all the tweets about the given topic. The data collection should take at least 6-8 weeks to make sure you have enough event data to do your analysis. You should retrieve only tweets that have geolocations and remove all the retweets. The tweets should be preprocessed (see the previous discussion in project B above). You also need to develop a sentiment analysis method that will predict whether a tweet has positive, negative, or neutral sentiment on the topic. To do this, you must first manually label some of the tweets as positive, negative, or neutral classes, and then train a classifier to predict the sentiment of the rest of the tweets. To improve the classifier, you may add other features for classification, such as the number of positive or negative words in the tweet. A list of words with positive and negative polarity is available at https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html. Use the geolocation to group together tweets from the same "region" (city, county or state, depending on whichever granularity you prefer). Aggregate the tweets for each region per day and create a time series that shows the net daily sentiment (#positive tweets - #negative tweets) in the region. If you want, you may also cluster the regions based on similarity of their time series. Finally, develop a web-based interface that shows changes in the sentiment over time for each region (or each cluster of regions).

D. Sports Analytics

There are many publicly available databases for professional sports players and teams. Examples are:

1. NFL (http://www.databasefootball.com/)
2. NBA (http://www.databasebasketball.com)
3. MLB (http://www.databasebaseball.com/)
4. NHL (http://www.databasehockey.com)

You can use the database to perform various types of analysis. For this project, you will need to do extensive preprocessing to convert the raw data into their appropriate formats. The following is an example of a prediction task you can perform using the sports database:

Given a pair of teams, (X,Y), where X is the home team and Y is the away team, predict who will win the game or what is the point difference at the end of the game. You will need to extract features for the home team and the away team (e.g., their offensive statistics over the last, say, 10 games, the defensive statistics over the last 10 games, statistics about their players, etc). You will need to create a data set containing examples of games where X had beaten Y, Y had beaten X, or X drew with Y. Train a classifier to make the prediction for future games. Report the accuracy of your classifier. For visualization, you can show how your predictions changes week by week for various games.

Instead of predicting the game outcome, you can also try predict teams that will make it to the playoffs, who are the best players, the final rankings of all teams, etc.