Large-Scale Data Mining: Models and Algorithms **ECE 219 Winter 2018**
Prof. Vwani Roychowdhury TA: Qiujing Lu, Tianyi Wang
UCLA, Department of ECE Arash Vahabpour

Project 5: Popularity Prediction on Twitter

Due on Wednesday, March 14 2018 by 11:59 pm

Introduction: A useful practice in social network analysis is to predict future popularity of a subject or event. Twitter, with its public discussion model, is a good platform to perform such analysis. With Twitter's topic structure in mind, the problem can be stated as: knowing current (and previous) tweet activity for a hashtag, can we predict its tweet activity in the future? More specifically, can we predict if it will become more popular and if so by how much? In this project, we will try to formulate and solve an instance of such problems.

The available Twitter data is collected by querying popular hashtags related to the 2015 Super Bowl spanning a period starting from 2 weeks before the game to a week after the game. We will use data from some of the related hashtags to train a regression model and then use the model to make predictions for other hashtags. To train the model, you need to prepare training sets out of the data, extract features for them, and then fit a regression model on it. The regression model will try to fit a curve through observed values of features and outcomes to create a predictor for new samples. Designing and choosing good features is one of the most important steps in this process and is essential to getting a more accurate system. There are examples of such analysis and useful features in literature [1] (You should look into the literature for this). You will be given training data to create the model, and test data to make predictions. The test data consists of tweets containing a hashtag in a specified time window, and you will use your model to predict number of tweets containing the hashtag posted within one hour immediately following the given time window.

Part 1: Popularity Prediction

Problem 1.1

Download the training tweet data [2] and calculate the following statistics for each hashtag:

- Average number of tweets per hour
- Average number of followers of users posting the tweets

[1] http://arxiv.org/abs/1401.2018

[2] https://ucla.box.com/s/nv9td9kvvfvg3tya0dlvbs1kn5o87gmv

- Average number of retweets.

Q: Plot "number of tweets in hour" over time for `#SuperBowl` and `#NFL` (a histogram with 1-hour bins). The tweets are stored in separate files for different hashtags and files are named as `tweet_[#hashtag].txt`.

Note: The tweet file contains one tweet in each line and tweets are sorted with respect to their posting time. Each tweet is a JSON string that you can load in Python as a dictionary. For example, if you parse it to object `json_object`, you can look up the time a tweet is posted by:

```
json_object['citation_date']
```

You may also assess the number of retweets of a tweet through the following command:

```
json_object['metrics']['citations']['total']
```

Besides, the number of followers of the person tweeting can be retrieved via:

```
json_object['author']['followers']
```

For the following problems, you may need to parse the time in the following way:

```
import datetime, time
import pytz
pst_tz = pytz.timezone('US/Pacific')
datetime.datetime.fromtimestamp(citation_date, pst_tz)
```

Use this to convert the datetime data in seconds to a datetime object with given time zone.

Problem 1.2

For each hashtag, fit a Linear Regression model using the following 5 features to predict number of tweets in the next hour, with features extracted from tweet data in the previous hour.

The features you should use are:

- Number of tweets (hashtag of interest)
- Total number of retweets (hashtag of interest)
- Sum of the number of followers of the users posting the hashtag
- Maximum number of followers of the users posting the hashtag
- Time of the day (which could take 24 values that represent hours of the day with respect to a given time zone)

For each hashtag, you should train a separate model.

Q: For each of your models, report your model's training accuracy and R-squared measure. Also, analyse the significance of each feature using the *t*-test and *P*-value. You may use the libarary `statsmodels.api` in Python.

Hint: You can create time windows from the data to extract features. Each window will provide a sample for your regression model. E.g. You can divide the data in 1-hour windows and use features from each 1-hour (or *n*-hour) window to predict number of tweets for the next 1-hour window.

Problem 1.3

Design a regression model using any features from the papers you find or other new features you may find useful for this problem. Fit your model on the data of each hashtag and report fitting accuracy and significance of variables.

Q: For each of the top 3 features in your measurements, draw a scatter plot of predictant (number of tweets for next hour) versus value of that feature, using all the samples you have extracted, and analyze it.

Hint: If you have designed good features, you should observe a relatively linear relationship between your top features and your target value.

Problem 1.4

The task of k-fold cross validation involved the following procedure. First of all, you split training data into k (equal) parts. Then, you perform cross-validation: you run k tests, each time fitting your model on the aggregate of $k - 1$ parts and predicting the number of tweets for the 1 remaining part.

In this part, we would like to perform 10-fold cross-validation on the models from the previous part and calculate the average prediction error over samples in the held-out part for the 10 tests. For this problem, you should split the feature data (your set of (features, predictant) pairs for windows) into 10 parts to perform cross-validation. Also, your evaluated error should be of the form $|N_{\text{predicted}} - N_{\text{real}}|$.

Since we know the Super Bowl's date and time, we can create different regression models for different periods of time. First, when the hashtags haven't become very active, second, their active period, and third, after they pass their high-activity time. Train 3 regression models for these time periods (The times are all in PST):

1. Before Feb. 1, 8:00 a.m.
2. Between Feb. 1, 8:00 a.m. and 8:00 p.m.
3. After Feb. 1, 8:00 p.m.

Q: For each hashtag, report the average cross-validation errors for the 3 different models. Note that you should do the 90-10% splitting for each model within its specific time window. I.e. Only use data within one of the 3 periods above for training and testing each time, so for each period you will run 10 tests.

Also, aggregate the data of all hashtags, and train 3 models (for the intervals mentioned above) to predict the number of tweets in the next hour on the aggregated data. Q: Perform the same evaluations on your combined model and compare with models you trained for individual hashtags.

Problem 1.5

Download the test data[3]. Each file in the test data contains a hashtag's tweets for a 6-hour window (note that these hashtags are different from those in training data). Fit a model on the aggregate of the training data for all hashtags, and predict the number of tweets in the next hour for each test file. The file names show sample number followed by the period number the data is from. E.g. a file named sample5_period2.txt contains tweets for a 6-hour window that lies in the 2nd time period described in part 4. One can be creative here, and use the data from all previous 6 hours for making more accurate predictions (as opposed to using features from the previous hour only).

Q: Report the model you use. For each test file, provide your predictions on the number of tweets in the next hour.

Note: Test data should not be used as a source for training. You are not bounded to only linear models. You can find your best model through cross validation of your training data.

Part 2: Fan Base Prediction

The textual content of a tweet can reveal some information about the author. For instance, users tweeting on a topic may have opposing views about it. In particular, tweets posted by fans of different teams during a sport game describe similar events in different terms and sentiments. Recognizing that supporting a sport team has a lot to do with the user location, we try to use the textual content of the tweet posted by a user to predict their location. In order to make the problem more specific, let us consider all the tweets including `#superbowl`, posted by the users whose specified location is either in the state of Washington or Massachusetts. For example, in order to include all the tweets with the author located in the state of Washington, we consider the tweets that include the following substrings in the location field:

- Seattle, Washington
- Washington
- WA
- Seattle, WA
- Kirkland, Washington
- etc.

Q: Train a binary classifier to predict the location of the author of a tweet (Washington or Massachusetts), given only the textual content of the tweet (using the techniques you learnt in project 1). Try different classification algorithms (at least 3) in your submission. For each, plot ROC curve, report confusion matrix, and calculate accuracy, recall and precision.

[3] https://ucla.box.com/s/ojvvthudugp9d2gze5nuep9ogwjydnur

Part 3: Define Your Own Project

The dataset in hands is rich as there is a lot of metadata to each tweet. Be creative and propose a **new** problem (something interesting that can be inferred from this dataset) other than the previous parts. You can look into the literature of Twitter data analysis to get some ideas. Implement your idea and show that it works. As a suggestion, you might provide some analysis based on changes of tweet sentiments for fans of the opponent teams participating in the match. You get full credit for briniging in novelty and full or partial implementation of your new ideas.

Submission:

Please submit a zip file containing your report, and your codes with a readme file on how to run your code to CCLE. The zip file should be named as "Project5_UID1_UID2_..._UIDn.zip" where UIDx are student ID numbers of the team members. You should **not** include data in your submission.