EE239AS Project 3

Popularity Prediction on Twitter

Winter 2015

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 from project 2 is on hashtags related to the 2015 Super Bowl. We will use data from some of the 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 selected features for them, and then fit the regression model on it. The regression model will try to fit a line 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]. The regression analysis can be implemented in Python using the *statsmodels*[2] package. 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.

1) Download the training tweet data[3] and calculate these statistics for each hashtag: average number of tweets per hour, average number of followers of users posting the tweets, and average number of retweets. 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`. 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.

2) Fit a Linear Regression model using 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, total number of retweets, sum of the number of followers posting the hashtag, maximum number of followers in users posting the hashtag, and time of the day (which could take 24 values that represent hours of the day with respect to a given time reference). Explain your model's training accuracy and the significance of each feature using the t-test and P-value results of fitting the model.

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.

3) Design a regression model using any features from the paper or other new features you may find useful for this problem. Fit your model on the data and report fitting accuracy and significance of variables. For the top 3 features in your measurements, draw a scatter plot of predictant (number of tweets for next hour) versus feature value, using all the samples you have extracted.

4) Split the feature data (your set of *(features,predictant)* pairs for windows) into 10 parts to perform cross-validation. Run 10 tests, each time fitting your model on 9 parts and predicting the number of tweets for the 1 remaining part. Calculate the average prediction error $|N_{predicted} - N_{real}|$ over samples in the remaining part, and then average these values over the 10 tests.

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.

Report 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 for training and testing each time, so for each period you will run 10 tests.

5) Download the test data[4] and run your model to make predictions for the next hour in each case. Each file in the test data contains a hashtag's tweets for a 6-hour window. The file name shows 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.

Report your predicted number of tweets for the next hour of each sample window.

---

[4]

Submission: Please submit a zip file containing your codes, any output files, and a short report to ee239as.winter2015@gmail.com . If your zip file is too large to email, you can upload it on Dropbox (or Box at ucla.box.com) and send the link to this email. You can use any programming language but there will be samples of Python code given to you for crawling and parsing that you can use and modify. If you had any questions you can send an email to the same address.