Classifying Traffic Related Tweets for City of Chicago

Illinois Institute of Technology

Chicago, IL 60616

[Ishan Bilolikar, Latika Singh]

[ibilolik@hawk.iit.edu, lsingh3@hawk.iit.edu]

Abstract

Real-time road traffic congestion monitoring is an important and challenging problem. Social media is great resource of user-generated contents. Since social media can be retrieved in real time with relatively small building and maintenance costs, traffic operation authorities probably identify the social media data such as Twitter tweets as another type sensor for traffic demand. The first step in this process is to filter useful information from the sea of data. This paper proposes exploring traffic related information from Twitter. However, there is major challenge for this problem. This data which is collected from twitter is often times noisy, and therefore, further filtration technique is required to remove noise from the data to provide accurate congestion estimation. This paper proposes a model to extract reliable traffic related features from big and noisy twitter data. Major regions demonstrate the efficiency and effectiveness of our proposed approach. Using topic modeling and logistic regression, we analyze the features depicting the traffic condition in a particular. We compare different classification algorithms used for predicting the political ideology given some by testing the accuracy of classifiers.

Introduction

Traffic congestion is becoming a central transportation issue in big cities. People living in urban areas, like Chicago and New York, are increasingly concerned about real-time traffic conditions, calling for data mining technologies that can instantly estimate citywide traffic congestions. In place of expensive and inefficient traditional methods, researchers put forward an alternative traffic congestion estimation solution with lower cost and wider spatial coverage by utilizing social media. In this paper we are using one of the most popular social media sites Twitter as the data source. With the rising popularity of social media, Twitter has become an indispensable platform for online users to share real-time traffic information. By regarding each Twitter user as a traffic sensor, traffic information can be freely obtained. But that information contains a lot of noisy data along with useful data which can affect traffic analysis in various ways.
The goal of this paper is to gather as reliable data as possible so that further congestion analysis can be done accurately. Thus for this report we consider the following **hypothesis** :

"All tweets which seem to be traffic related because of the occurrence of traffic related words in it are not indicative of traffic conditions. There is some amount of noisy data even after targeting only traffic related tweets in search query."

The scientific community has made a great effort to provide effective solutions to analyze, structure, and process the large amount of on-line reviews in social media. A wide set of techniques of Feature Extraction (FE) are used in tweet texts to extract most indicative words that users use in these texts. In this respect, Twitter has become a popular micro-blogging site in which users express their opinions on a variety of topics in real time. The texts used in Twitter are called tweets, which are short texts of a maximum of 140 characters and a language that does not have any restriction on the form and content. In this paper, we answer the following question: how precisely can we extract tweet-based data related to traffic conditions of a particular area? To answer this question, we first manually label small dataset of tweets. Then we directly extract semantics from tweets via a sparse matrix, and incorporate the semantics into the logistic-regression model to predict label for other unlabeled tweets. Finally, the sparse matrix is obtained by solving an optimization framework, whose goal is to minimize the prediction error in the traffic related tweets.

Background

The Twitter Platform

Twitter is a popular social networking and microblogging site where users can broadcast short messages called 'tweets' to a global audience. A key

feature of this platform is that, by default, each user's stream of real-time posts is public. This fact, combined with its substantial population of users, renders twitter an extremely valuable resource for educational, commercial data mining and research applications. Of particular interest to this study, the role of Twitter as a platform is for determining traffic related tweets.

Data Mining and Feature Extraction

Owing to the fact that Twitter provides a constant stream of real-time updates from around the globe, much research has focused on detecting noteworthy, events and proper utilization of real-time data, as they rise to prominence in the public feed. Analyzing traffic condition is one such example with millions of tweets related to traffic were posted in Chicago alone.

Data and Methods

The Python Tweepy API

We are using the tweepy API for getting the Twitter data (http://docs.tweepy.org/en/v3.2.0/). Tweepy provides python interface. Twitter exposes a `web services API' and this library is intended to make it even easier for Python programmers to use. The API class provides access to the entire twitter RESTful API methods which would be needed for the purpose of this project to fetch user information and user tweets. The tweepy API provides authentication handler using the consumer_token and consumer_secret token

Fetching Tweets

This analysis focuses on 1-2 weeks of twitter data collected using 'chicago traffic' as the search parameter and geocode coordinates of 29 regions of Chicago obtained from www.data.cityofchicago.org. The data from every region was stored in different files.

Explicit labelling of the traffic related data

Since twitter data do not have pre-defined label for the tweets, 3 region files were manually labeled. Tweets that gave information on traffic were labeled as 1 and tweets that gave traffic unrelated information were labeled 0. 'region1.txt', 'region8.txt' and 'region22.txt', these three files were manually labeled.

Tokenizer Function

Tokenizer function of the data has a very important role in feature extraction. The parameters considered in the tokenizer can vary the average cross validation to a great extent.

In our tokenizer function we have considered 5 parameters that greatly define a tweet. The five parameters are:

1. Lowercase
2. Punctuations
3. Hashtags
4. URL
5. Mentions

It can be chosen in the function which parameter is to be selected.

Generating CSR Matrix

The content of a file, that is, all the tweets in a particular file are then converted to a document vs. features CSR matrix**.** The features are defined by the tokenizer function.

To generate the CSR matrix CountVectorizer from sklearn feature extraction, was used which takes input as content of file and tokenizer function.

Cross Validation function

Perform n-fold cross validation, calling get_clf() of LogisticRegression to train n different classifiers. Using sklearn's KFold class accuracy for each fold is calculated and in the end mean of all the accuracies is taken to give average accuracy of the training data.

Discriminative Classifiers

Discriminative classifiers, such as logistic regression, model the conditional distribution P(y|x) of the class labels given the features, and learn the model parameters through maximizing the conditional likelihood based on P(y|x).

$$\theta_i = \frac{1}{1 + \exp\left[-\left(\beta_0 + \sum_{j=1}^{k} \beta_j x_{ij}\right)\right]}$$

Fig2: *Formula for the Logistic Classifier*

A discriminative classifier tries to model by just depending on the observed data. It makes fewer assumptions on the distributions but depends heavily on the quality of the data

Experiments

1. Predicting Labels for unlabeled tweets using Logistic regression:

Firstly, we labelled one region's dataset having **978** tweets manually as class(y=1) traffic indicative tweets and class(y=0) non-traffic indicative tweets. Then performed tokenization and vectorization on it, trained our data on it and observed its accuracy using cross validation technique to be **88.47%**. Using vocabulary from this dataset, we predicted

labels for unlabeled tweets in other regions' datasets. After this experiment, we found the following results. Hypothesis of this paper that "*All tweets which seem to be traffic related because of the occurrence of traffic related words in it are not indicative of traffic conditions. There is some amount of noisy data even after targeting only traffic related tweets in search query"* is proved by Fig1 shown below which shows percentage of traffic data in tweets of different unlabeled datasets of different regions. The accuracy of labeled file 'region22.txt' observed for this prediction was **92.57%**.

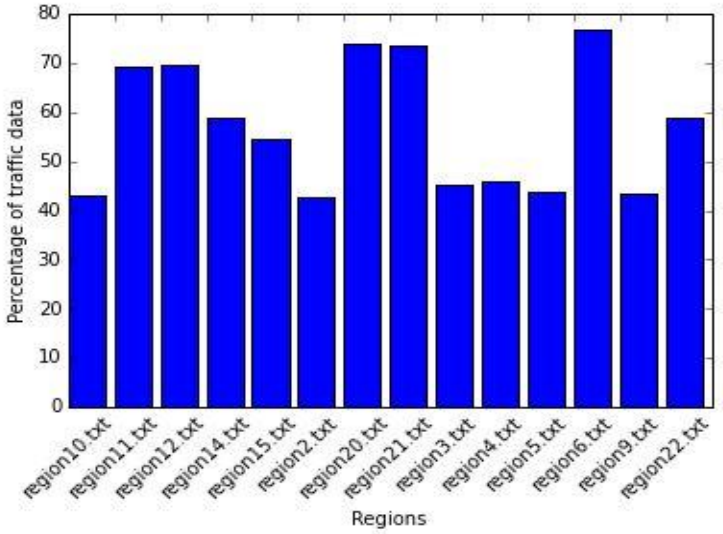


Fig.1. percentage of traffic data in tweets of different unlabeled datasets

2. *Testing Classifier for optimized results:*

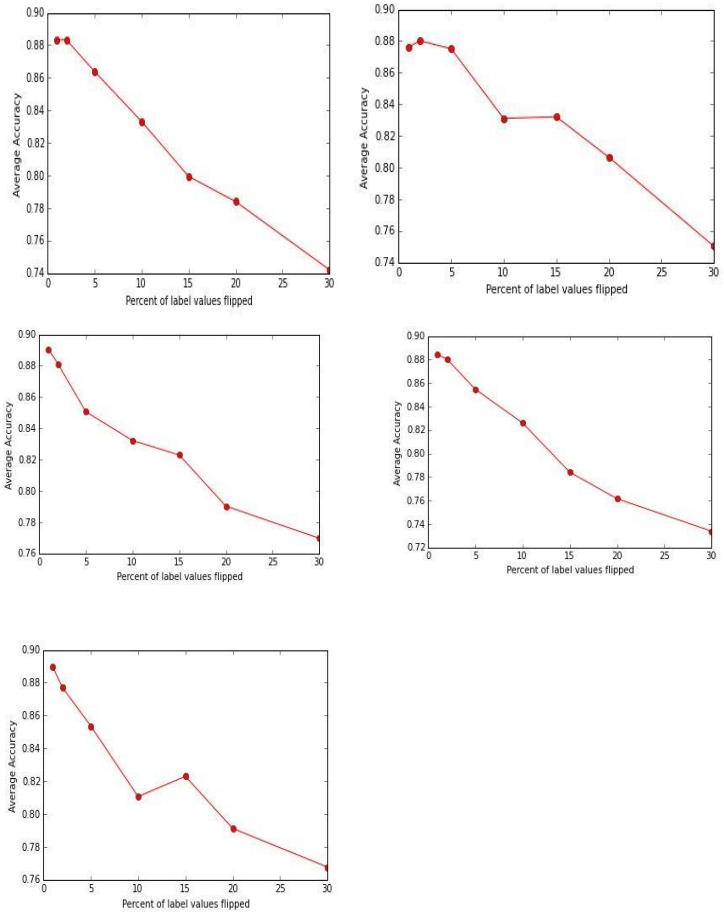
1) ***Removing Retweets:*** We removed retweets from our training data and trained the classifier again then checked the results.
accuracy after removing Re-tweets: 0.8407
We observed that removing the retweets decreases the accuracy of the classifier.

2) ***Reducing size of vocabulary/number of features:*** Here we have reduced the number of features of the classifier by training the classifier with less number of tweets compared to the previous. Using region22.txt labeled data which contains 202 tweets.

# of features	**890 features**		**336 features**	
accuracy score	region1.txt	region22.txt	region22.txt	region1.txt
	88.47%	92.57%	79.74%	61.25%

It can be observed that when number of features are less, the accuracy score for larger data reduces. In other words, the discriminative classifier performs better with larger training sets.

Testing robustness of the classifier: This method calculates the robustness of the classifier. In this method we change small percentage of y label. eg. change 1 to 0 or 0 to 1. Random function selects the number of labels to be changed according to the percent value. eg. 10 percent of 700 labels has to be changed, then the random function chooses 70 labels and flip its value.
From below figures we can see that the accuracy of the classifier decreases steeply after more than 2-5% of noise is induced in the labels



Standard deviation:

[0.05006212, 0.04367757, 0.0548806 , 0.04117709, 0.04162134]

3) ***Settings of Tokenizer:*** We have experimented with different settings of tokenizer function by setting its decision parameters such converting tweets into lowercase, considering url or not, considering mentions or not, considering hashtag or not , considering punctuations or not while tokenizing tweets. We have observed that best setting for the tokenizer function is when lower case is set True, collapse_url is set True, collapse_mention is set true, collapse_hashtag is set False and punctuation is set False.

Related Work

In this section, we review the existing work related to our study. One line of research is tweet classification for the purpose of information filtering. For example, in [Go et al., 2011], the authors test various algorithms for classifying the sentiment of tweets, such as SVM, Naive Bayes, etc; in [Sriram et al., 2010], the authors use a small set of domain specific features in addition to the bag-of-word features to classify tweets into a predefined set of classes; etc. Another line of related work is anomaly detection via mining social media content. Recently, microblogging services (e.g., twitter) have received much research attention in the fields of anomaly detection. Researchers consider the twitter tweets as real-time social streams and focus on analyzing the features of keywords in the specific context to detect events [7,8,9]. The key challenges in these works is to filter out the irrelevant contents in the tweets, which requires computationally expensive filtering, such as the Kalman filtering based model proposed by [7] and the Gibbs Random Field defined probabilistic model in [8]. However, in our work, we have conducted a simple filtering technique to separate out the irrelevant contents, as discussed in above sections. This is motivated by the presence of a large number of tweets related to traffic conditions. Our study is different from the above mentioned study as we are not using sentiment analysis and Naïve Bayes for tweets classification.

Lessons Learned

Our experimental results emphasize not only on advantage of using features extraction for classification and then comparing the results with true labels i.e., accuracy, but also point out the scalability limitation. The more features the more rules to classify tweets which impacts accuracy. Furthermore we have learnt the impact of different ways of tokenizing the tweets.

Conclusions and Future Work

This work, focusing on Traffic, presents how traffic indicative tweets can be extracted from big and noisy datasets. We built a model which classifies and predicts whether or not a tweet is traffic indicative or not .We (i) presented its challenges, (ii) motivated the use of feature exraction, and (iii) exposed its scalability together with its limitation. The experiments have shown accurate and consistent prediction of traffic tweets. Future domains of investigation are: (i) Investing traffic congestion on roads using our filtered data. (ii) Investigating particular locations of traffic congestion. (iii)Finding the cause of traffic congestion by extracting information about special events such as concerts, accidents, blocks, construction etc. from the tweets.(iv) Predicting

References

1.) Jingrui He, Wei Shen , Phani Divakaruni , Laura Wynter , Rick Lawrence : Improving Traffic Prediction with Tweet Semantics. Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence.

2.) Ming Ni, Qing He, Jing Gao : Using Social Media to Predict Traffic Flow under Special Event 2 Conditions. Submitted to TRB 93rd Annual Meeting for Presentation and Publication 26 January 2014, Washington D.C. 27 November 15, 2013.

3.) Po-Ta Chen , Feng Chen ,Zhen Qian :Road Traffic Congestion Monitoring in Social Media with Hinge-Loss Markov Random Fields. Published in: Data Mining (ICDM), 2014 IEEE International Conference.

4.) http://www.ibm.com/smarterplanet/us/en/traffic_congestion/article/traffic-management-and-prediction.html.

5.) http://www.popularmechanics.com/cars/a6524/will-in-car-social-media-kill-the-traffic-jam/

6.) http://cs229.stanford.edu/notes/cs229-notes2.pdf

7.) T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In WWW ’10

8.) C. X. Lin, B. Zhao, Q. Mei, and J. Han. PET: a statistical model for popular events tracking in social communities. In KDD ’10.

9.) H. Sayyadi, M. Hurst, and A. Maykov. Event detection and tracking in social streams. In ICWSM ’09). AAAI, 2009