Applying Unsupervised and Supervised Machine Learning Methodologies in Social Media Textual Traffic Data

Konstantinos Kokkinos[1(✉)], Eftihia Nathanail[2], and Elpiniki Papageorgiou[1,3]

[1] Computer Science Department, University of Thessaly, Lamia, Greece
kokkinos@uth.gr
[2] Civil Engineering Department, University of Thessaly, Volos, Greece
[3] Electrical Engineering Department, University of Applied Sciences, Technological Educational Institute of Central Greece, Lamia, Greece

Abstract. Traffic increasingly shapes the trajectory of city growth and impacts on the climate change in modern cities. Traffic patterns' monitoring can provide with innovative practices in understanding city traffic dynamics, especially via utilizing sensory and textual data analytics. State-of-the-art research recently has focused on processing voluminous real time data in vast quantities by capturing real time sensory observations and/or social network (textual) data regarding city traffic. In this paper, we investigate the feasibility of using Big Data produced by Twitter textual streams for extracting traffic related events. After describing a generic yet innovative application used for data capturing, we preprocess this data so they fit into the structuring of the machine learning models for clustering (unsupervised learning) and classification (supervised learning). For the case of clustering we use Apache Spark on a MapR sandbox with the use of KMeans algorithm. For the classification case we compare various machine learning methodologies including Multi-Layer Perceptron Neural Networks, (MLP-NN), Support Vector Machines, (SVM) and a Deep Convolutional Learning, (DCL) approach to contextualize citizen observations and responses via tweets. The criteria of precision, accuracy, recall and F-score are used as statistical metrics to determine the accuracy and performance of each model. Our experiments include clustering, a 2-class and a 3-class classification, where, MLP-NN gave accuracy of 89.6%, SVM 92.73% and DCL was inferior performing at 81.76%.

Keywords: Unsupervised · Supervised · Deep Learning · Big Data Textual · Traffic

1 Introduction

It has been noted by many researchers, lately, that the recent accelerated urbanization has altered the equilibrium state of urban road systems in modern cities. Traditional Transportation Systems (TS) in big cities can no longer meet the needs of today's complex transport and congestion caused by the continuous increment of vehicle use.

E. G. Nathanail and I. D. Karakikes (Eds.): CSUM 2018, AISC 879, pp. 665–672, 2019.
https://doi.org/10.1007/978-3-030-02305-8_80

This global urbanization trend delivers a new set of challenges to authorities as they must reconfigure city services according to the new priorities imposed for planning and mitigating unforeseen traffic incidents. For this reason, the Intelligent TS (noted as ITS) research topic was created having as primary goals to: (a) advance the traffic monitoring methodologies and (b) offer better transportation planning and traffic management in congested cities [1]. Present works include sensor based monitoring schemes where the sensory equipment is installed/maintained by the city. Additional data generated from a variety of devices installed in vehicles (such as GPS, radio transceivers, small-scale collision radars and sensing devices to enhance travel safely etc.) are also used in recent studies towards the aforementioned goals. However these data cannot be stored centrally since the devices are designed to operate using short-range communication protocols. The tremendous shift in data-induced methodologies has happened due the use of social media platforms (local online forums, Facebook, Twitter) which nowadays are used as the primary and richest source of real time data [1–3]. The utilization of social media traffic related data (traffic jams, collisions, alternative route suggestions etc.) help ITS to improve traffic monitoring and management. But all this comes also with new challenges: (a) ITS have to overcome the Big Data consequences emanating by the rate of data generation (8,000 tweets/sec, hundreds of trillions per week) and the storage inability, (b) new intrinsic spatiotemporal principles of Big Data must feedback innovative machine learning solutions to optimize cloud computing and processing, (c) open availability of data poses social challenges of geospatial significance and (d) textual analysis must coexist with specialized Deep Learning approaches to decode human responses.

After we present in Sect. 2 a concise State of the Art, we formulate in Sect. 3 the methodology of an ITS used for clustering and classification of traffic related data extracted by a Twitter extractor that incorporates the stemming, IDF and similarity index techniques to choose traffic-incident related keywords. The classification methodology is also presented for a 2-class and a 3-class classifier. For the 2-class classification we provide performance metrics incorporating a MLP-NN and a SVM. For the 3-class classification we do the same using a DCL network.

2 State of the Art

There has been a variety of initiatives (both academic and commercial) dealing with traffic alert systems. At the same time all these systems harvest input information from a variety of sources including sensory equipment, human reactions, police traffic reports etc. In relevance to textual incoming sources from social media, while there is a lot of literature regarding data analytics methodologies, there hardly exists research that deals with the stages of data discovery, collection, and preparation from textual data [4]. Recently, Twitter started to provide services where, users can post geo-tagged tweets via the GPS interface of their smart devices [5, 6]. The reported information when relevant can support any traffic monitoring and alerting system by just logging a repertoire of traffic incidents. Towards these lines, TWITRAFFIC, [7] created a smart app that monitors and reports traffic events in UK. MISNIS [8] is another platform that facilitates these issues and allows a non-technical user to easily mine any given topic

from Twitter's corpus in order to obtain relevant contents and indicators such as user influence or sentiment analysis but it is focused mainly in the Portuguese language. Lately, [9] developed a clustering tool called I-TWEC, which utilizes Twitter data lexical and semantic similarities. I-TWEC uses the Longest Common Subsequence technique as a similarity metric to produce clusters presented with different visualizations enabling users to merge them based on their semantic similarity. Because the traffic topics attract global attention, such data suffer from the Long Tail Effect [10], thus an effective textual analytics tool must be a traffic event data extraction model, however it must be able to distinguish learning of specific locations utilizing only tweets via geolocation attributes. Even though it is somewhat difficult for travelers to read and/or for drivers to participate in such activities, experience has shown that almost all drivers and passengers during traffic rush hours, announce this on social media. Thus the optimum solution is to analyze the information available on social network platforms, perform sentiment analysis and machine learning methodologies to classify and cluster traffic cases and to predict traffic situations.

With the improvement of big data processing technologies, we now have the ability to perform traffic sensing and learn human mobility patterns from updated location information in network interaction log data (mostly GPS and textual). Recently, [11] extracted traffic patterns from big data using regression models. Also the research shown in [12] adopts Spark on Hadoop and MongoDB technologies to store, handle and process real time and historical traffic data from heterogeneous sources including social media. Similar work is also recognized in [13] where the distributed file system HDFS is used to store urban traffic data and the Spark is used to realize road traffic congestion state detection with lower cost, shorter period and more credible results.

The K-means methodologies are the most popular for data clustering. However, for the case of high-dimensional text data, K-means clustering becomes the only known solution. The cosine similarity property metric [14] is used to measure cohesion between produced clusters since it is a similarity measure between two non-zero vectors of an inner product. Finally regarding the ITS traffic, condition recognition is very important and K-means methods have been tried lately towards this issue [15, 16].

3 Methodology and Results

Gathering of tweets was achieved using the Twitter4J [17] open source Java library. The usage of the Twitter API allows us to mine tweets using criteria based on hashtags, limited time, longitude and latitude and any keyword. In this paper we focus on two main investigations: (a) Clustering of traffic data of numeric nature via the use of KMeans algorithm with the Euclidean distance as a cost function and (b) Classification considering two cases: (i) binary classification regarding tweets related to traffic either due to weather conditions or not and (ii) a ternary classification related to heavy traffic due to accidents, seasonality affected events (for example, Christmas Eve) and external unexpected events (basketball game, strikes, demonstrations etc.).

3.1 Traffic Big Data Clustering Using Unsupervised Machine Learning

We used one of the most commonly used clustering algorithms (KMeans) to cluster twitter data into a predefined number of K-clusters. Data was gathered using the Twitter4J Java library for the city of New York during the Christmas period of 2017 (Dec. 11th 2017–Jan. 3rd 2018). The area of interest was chosen to be the virtual rectangle (left upper corner: Hawthorne NJ Lat: 40.939825, Log: –74.160612, right lower corner: Jones Beach State Park NY Lat: 40.597646, Log: –73.505552). Apart from the geolocation and the date and time searching criterion of data acquisition, an additional searching criterion included keywords such as: congestion, traffic jam, traffic etc. Initial filtering of the aforementioned tweets was performed to mine the ones originated by people riding vehicles and therefore excluding pedestrians. The methodology used was based on the calculation of velocity of the tweet transmitter by taking two consecutive tweets. However, we owe to mention that the method does not guarantee to exclude all pedestrians since in heavy traffic conditions, vehicles may move at pedestrian speeds. Around 2.7 million of tweets are fed to a single machine Spark ML and SQLContext schema. After setting $k = 7$ clusters each geolocated tweet was assigned to its nearest centroid based on the Euclidean distance metric. These centroids depict the epicenters of intensified heavy traffic activities. The centroids were then updated in each pass of the algorithm and the process was repeated until there was a minimum change of the centroids. The structure used in the data frames of Spark was: (Date, Time, Latitude, Longitude, Keyword). Separate runs were performed for the aforementioned keywords. Figure 1 shows the centroid locations of the scenario with the keyword "congestion" (Table 1).

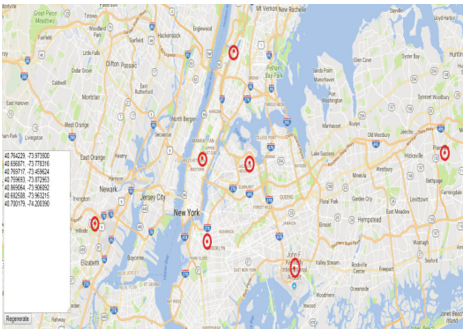

Fig. 1. Centroid locations in Google Maps.

Table 1. Centroid coordinates.

Latitude	Longitude
40.76122883211822	–73.1234999829014
40.69877065064515	–73.7231638518956
40.76971711652399	–73.4544454489798
40.75969284538119	–73.7365292246709
40.869063991800815	–73.3287442854547
40.40856985998673	–73.8958754983675
40.768743687636723	–74.980658943283

3.2 Traffic Big Data Classification Using Supervised Machine Learning

Classification Data Set Acquisition. For the case of the binary classifier, the set of tweets either include the weather condition in a traffic event or not. The tweets were gathered from the same area as in clustering and had the same structure (Date, Time, Latitude, Longitude, Keywords) where an *m*-at most tuple creates the keywords.

For m up to 10 such candidate keywords included the words {traffic, rain, snow, sleet, accident, slowdown, congestion, stuck, thunder, crash} when investigated heavy traffic due to extreme weather conditions. For the case of the ternary classifier the same tweet structure was used with keyword tuples of the form {game, strike, demonstration, flight, Christmas, year, accident, crash, ambulance, shopping}.

Data Set Preprocessing. Data fetching was followed by a set of preprocessing procedures dealing with:

1. *Removal of tweet meta-associations* using a Java Regular Expression Filter [18] to discard hashtags, links, mentions and user-ids out coming a set of strings S_i, $i = 1 \ldots N$. The S_i's are further converted to lower case characters via the *tolower*(S_i) procedure.
2. *Tokenization of* S_i*'s* using a Java tokenizer [19] so that, all S_i's were transformed into a larger set of syllables or words called tokens with the synchronous extraction of non-text characters (apostrophes, hyphens etc.)
3. *Extraction of stop-words* [20] i.e. words with no statistical significance, conjunctions, articles, pronouns etc.
4. *Stemming* of tokens using the Porter's algorithm [21] to remove suffices of tokens and to group words of similar semantics. The outcome of this process was the set of ST_i, $i = 1 \ldots N$ stemmed tokens. N was be the training set for the machine learning algorithms used later on thus is denoted as N_{tr}. For each stemmed token st_j in N_{tr} we compute its importance in the training set using the *Inverse Frequency Index* (IDF) as:

$$w_{st} = \ln(N_{tr}/N_{st}) \tag{1}$$

   where N_{st} is the occurrence index of the stemmed token in N_{tr} [22].
5. For the set of calculated IDF's we built a feature representation vector $F = (f_{j_1}, f_{j_2}, \ldots, f_{j_{Ntr}})$ where each element was set according to:

$$f_j^{st} = \begin{cases} w_{st} & \text{if stemmed token} \in N_{tr} \\ 0 & \text{if stemmed token} \notin N_{tr} \end{cases} \tag{2}$$

6. *Information Gain,* (IG) calculation for each stemmed token ST_i for the class vector $C = \{c_1, c_2, \ldots, c_m\}$. Note that for our aforementioned scenarios |C| = 2 or 3. The IG(ST_i) is:

$$\begin{aligned} IG(ST_i) = &- \sum\nolimits_m P(C_m) \log P(C_m) + P(ST_i) \sum\nolimits_m P(C_m/ST_i) \log P(C_m/ST_i) \\ &+ P(\overline{ST_i}) \sum\nolimits_m P(C_m/\overline{ST_i}) \log P(C_m/\overline{ST_i}) \end{aligned} \tag{3}$$

where $P(ST_i)$ is the probability that the stemmed token ST_i occurs in (3), $\overline{ST_i}$ is the occurrence negation, $P(C_m)$ is the probability of the mth class value, $P(C_m/ST_i)$ is the conditional probability of the mth class value given that ST_i occurs and $P(C_m/\overline{ST_i})$ is the conditional probability of the mth class value given that ST_i does not occur.

3.3 Classification Using a MLP-NN and SVM

For the first experiment we used an MLP NN as a binary classifier from the April-ANN toolkit [23]. More in detail, we used the [MLP: ann.mlp.all_all.generate] call to involve an all-to-all connection between the hidden layers of the NN concentrating on the performance of the classifier that has only two classes-positive and negative. This allowed us the investigation of the: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) instances and the calculation of the performance metrics: Accuracy (4), Precision (5), Recall (6) and F-Score (7)

$$Acc = (TP + TN)/(TP + FP + FN + TN) \tag{4}$$

$$Prec = TP/(TP + FP) \tag{5}$$

$$Rec = TP/(TP + FN) \tag{6}$$

$$F\text{-}Score = (1 + \beta^2)\frac{\text{Prec} \cdot \text{Rec}}{(\beta^2 \cdot \text{Prec}) + \text{Rec}} \tag{7}$$

where $\beta = 1$ for class-balanced datasets.

For the second experiment we used a SVM as in [24] noting that the optimization problem under concern makes use of kernels, which map input features into a different space. This means that finding the derivative of the cost function and using gradient descent does not work, but instead, the SVM only weights examples that are close to the decision boundary. Table 2 depicts the classification results on the 2-class dataset for the two classifiers mentioned above, indicating the best performance in bold.

Table 2. Classification results for the 2-class dataset.

Classifier	Accuracy (%)	Precision (%) by class		Recall (%) by class		F-Score (%) by class	
		Weather caused	NotWeather caused	Weather caused	NotWeather caused	Weather caused	NotWeather caused
MLP-ANN	89.6	90.7	88.9	88.54	90.63	89.22	89.91
SVM	**92.73**	**92.06**	**93.4**	**92.80**	**92.66**	**93.02**	**92.44**

3.4 Classification Using DCL Network for Sentiment Analysis

For the case of the 3-class classifier we used the Deep Convolutional Neural Network shown in [25]. The training of the network was done by stochastic gradient descent via the use of a backpropagation algorithm to compute the gradients. The tendency of the network to over fit in the learning process of the decision function was confronted by augmenting the cost function. The testing of the model was done on the pre-processed tweet data in a 70% to 30% ratio between the train and test datasets. Unfortunately the results were inferior to the 2-class classifier as depicted in Table 3.

Table 3. 3-class deep convolutional learning classifier.

	Due to Accidents	Seasonality affected	External events
Accuracy (%)	81.76		
Precision (%) by class	79.65	80.92	84.72
Recall (%) by class	82.34	82.18	80.49
F-Score (%) by class	82.21	81.28	81.79

4 Conclusions

With the increase of vehicular traffic observed in recent years in urban areas, there has been a significant degradation of the efficiency of the traffic flow. The incorporation of machine learning methodologies is shown to be beneficial in identifying congestion centroids for the case of clustering traffic congestion related data generated by social media. Furthermore, for the case of classification in discovering the reasons of occurrence of congestion events, binary classifiers (MLP-NN and SVM) outperform the utilization of Deep Learning models. We suspect that this limited utilization of DCL is due to the fact that we have not used pre-trained embedding of neural language model thus, further investigation is apparent in justification of this comparison.

References

1. Liu, B., Hu, M., Cheng, J.: Opinion observer: analyzing and comparing opinions on the web. In: Proceedings of the 14th International Conference on World Wide Web, pp. 342–351. ACM (2005)
2. Cao, J., Zeng, K., Wang, H., Cheng, J., Qiao, F., Wen, D., Gao, Y.: Web-based traffic sentiment analysis: methods and applications. IEEE Trans. Intell. Transport. Syst. **15**(2), 844–853 (2014)
3. Kim, S.M., Hovy, E.: Extracting opinions, opinion holders, and topics expressed in online news media text. In: Proceedings of the Workshop on Sentiment and Subjectivity in Text. Association for Computational Linguistics, pp. 1–8 (2006)
4. Stieglitza, S., Mirbabaiea, M., Rossa, B., Neubergerb, C.: Social media analytics – challenges in topic discovery, data collection, and data preparation. Int. J. Inf. Manag. **39**, 156–168 (2018)
5. Atefeh, F., Khreich, W.: A survey of techniques for event detection in Twitter. Comput. Intell. **31**(1), 132–164 (2015)
6. Ruchi, P., Kamalakar, K.: ET: events from tweets. In: Proceedings of the 22nd International Conference of World Wide Web Computing, Rio de Janeiro (2013)
7. Twitraffic Homepage. https://uk-traffic-news-twitraffic.soft112.com/. Accessed 10 Dec 2017
8. Carvalho, J., Rosa, H., Brogueira, G., Batista, F.: MISNIS: an intelligent platform for Twitter topic mining. Expert Syst. Appl. **89**, 374–388 (2017)
9. Arın, I., Erpam, M., Saygın, Y.: I-TWEC: interactive clustering tool for Twitter. Expert Syst. Appl. **96**, 1–13 (2018)
10. Liu, H., Ge, Y., Zheng, Q., Lin, R., Li, H.: Detecting global and local topics via mining Twitter data. Neurocomputing **273**, 120–132 (2018)

11. Alamy, I., Ahmedy, M., Alamy, M., Ulissesz, J., Faridy, D., Shatabday, S., Rossettiz, R.: Pattern mining from historical traffic Big Data. In: IEEE Region 10 Symposium (TENSYMP) (2017)
12. Guerreiro, G., Figueiras, P., Silva, R., Costa, R. Goncalves, R.: An architecture for Big Data processing on intelligent transportation systems. In: IEEE 8th International Conference on Intelligent Systems (2016). ISBN 978-1-5090-1354-8/16/$31.00
13. Guo, Y., Zhang, J., Zhang, Y.: A Method of traffic congestion state detection based on mobile Big Data. In: IEEE 2nd International Conference on Big Data Analysis (2017). ISBN 978-1-5090-3619-6/17/$31.00
14. Cosine Similarity. https://en.wikipedia.org/wiki/Cosine_similarity. Accessed 10 Dec 2017
15. Montazeri-Gh, M., Fotouhi, A.: Traffic condition recognition using the K-means clustering method. Trans. B Mech. Eng. Sci. Iran. **18**(4), 930–937 (2011)
16. Zhong, S.: Efficient online spherical K-means clustering. In: Proceedings of IEEE International Joint Conference on Neural Networks. Published in IJCNN (2005)
17. Twitter4J: Java Library for Twitter Mining. http://twitter4j.org/en/. Accessed 17 Dec 2017
18. Habibi, M.: Real World Regular Expressions with Java 1.4. Springer, Berlin (2004)
19. Hotho, A., Nürnberger, A., Paaß, G.: A brief survey of text mining, LDV Forum-GLDV. J. Comput. Linguist. Lang. Technol. **20**(1), 19–62 (2005)
20. Zhou, Y., Cao, Z.-W.: Research on the construction and filter method of stop-word list in text preprocessing. In: Proceedings of the 4th ICICTA, Shenzhen, vol. 1, pp. 217–221, (2011)
21. Porter, M.F.: An algorithm for suffix stripping. Program **14**(3), 130–137 (1980). Program electronic library and information systems
22. Aiello, L.-C., Petkos, G., Martin, C., Corney, D., Papadopoulos, S., Skraba, R., Göker, A.: Sensing trending topics in Twitter. IEEE Trans. Multimed. **15**(6), 1268–1282 (2013)
23. APRIL-ANN Toolkit: https://github.com/april-org. Accessed 16 Nov 2017
24. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In: Schoelkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods: Support Vector Learning, pp 185–208. MIT Press, Cambridge (1999)
25. Severyn, A., Moschitti, A.: Twitter sentiment analysis with deep convolutional neural networks. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2015, Santiago, pp. 950–962 (2015)