Traffic Condition Monitoring Using Social Media Analytics

Taiwo Adetiloye and Anjali Awasthi

Abstract Scientist and practitioner seek innovations that analyze traffic big data for reducing congestion. In this chapter, we propose a framework for traffic condition monitoring using social media data analytics. This involves sentiment analysis and cluster classification utilizing the big data volume readily available through Twitter microblogging service. Firstly, we examine some key aspects of big data technology for traffic, transportation and information engineering systems. Secondly, we consider Parts of Speech tagging utilizing the simplified Phrase-Search and Forward-Position-Intersect algorithms. Then, we use the *k*-nearest neighbor classifier to obtain the unigram and bigram; followed by application of Naïve Bayes Algorithm to perform the sentiment analysis. Finally, we use the Jaccard Similarity and the Term Frequency-Inverse Document Frequency for cluster classification of traffic tweets data. The preliminary results show that the proposed methodology, comparatively tested for accuracy and precision with another approach employing Latent Dirichlet Allocation is sufficient for predicting traffic flow in order to effectively improve the road traffic condition.

1 Social Media Analytics for Traffic Condition Monitoring

Perhaps the emergence of big data technology could not have been more disruptive anywhere else than in transportation and traffic engineering systems. This is considering that daily traffic flow of human transportation holds vast big data yet to be fully harnessed for real time estimation and prediction. Lu et al. [1] observed that such rapid development of urban "informatization", in the era of big data, offers several details entrenched in some spatio-temporal characteristics, historical correlations and multistate patterns. Undoubtedly, big data have increasingly been used

T. Adetiloye (✉) · A. Awasthi
Concordia Institute for Information and Systems Engineering (CIISE), Montreal, Canada
e-mail: t_adeti@encs.concordia.ca

A. Awasthi
e-mail: anjali.awasthi@concordia.ca

S. S. Roy et al. (eds.), *Big Data in Engineering Applications*,
Studies in Big Data 44, https://doi.org/10.1007/978-981-10-8476-8_13

for discovering subtle population patterns and heterogeneities that are not possible with small-scale data [2]. For these reasons amongst others academia, governments, federal and state agencies, industries, and other organizations continue to seek innovations to manage and analyze big data; providing them the prospect of increasing the accuracy of predictions, improving the management and security of transportation infrastructures while enabling informed decision-making to gain better insight into their transportation and traffic engineering phenomena [3].

The practical significance of real-time traffic flow state identification and prediction using big data lies in the ability to identify and predict traffic flow state efficiently, timely and precisely [1]. Various articles [3–5] have employed big data resources to examine traffic demand estimation, traffic flow prediction and performance as well as integration, and validation with existing models. A noteworthy aspect is that the rapidly increasing (big data) volume of leading social media microblogging services such as Twitter (twitter.com) can be pragmatically challenging, and nearly impossible to manually analyze [6]. Nevertheless, the huge volume of data derived from Twitter makes it ideal for machine learning.

Few years ago, researchers developed sentiment and cluster analysis to monitor twitter messages, identify followers and followings, find word resemblances and examine the nature of the comments i.e. positive, negative or neural. Such promising twitter analytic tools appear to be sufficient in solving the aforementioned traffic flow problems. Our objective in this study is tweet mining of the twitter UK traffic delays and to perform sentiment analysis and cluster classification for traffic congestion prediction. The proposed methodology is based on tweet crawling, preprocessing steps, feature extraction and social network generation and cluster.

1.1 Traffic Twitter Sentiment Analysis

Following the launch of twitter in 2006, sentiment analysis has been applied to various areas of interests e.g. extracting adverse drug reactions from tweets [7], news coverage of the nuclear power issues [8], and in the tourism sector for capturing sentiment from integrated resort tweets [6]. Terabytes of twitter data could be from traffic road users expressing their opinions on traffic jam, road accidents and other information which constitute general traffic news update. The question, of course, is how to determine traffic flow state based on the weight as measured by the opinion contained in a twitter message (called "tweet")—a short message that a sender post on twitter that cannot be longer that maximum 140 characters? According to Abidin et al. [9], certain special characters including @, RT, and # symbols used in a tweet creates a collective snapshot of what people are saying about a given topic. An in-depth process of computationally identifying and automatically extracting opinions from a writer's piece of text to determine whether the attitude or emotions towards a topic is positive, negative or neutral is known as sentiment analysis [10, 11]. The technique of sentiment analysis is generally

expected to yield a high accuracy rate of roughly 70–80% in training-test data matching tasks [12], while objectively seeking useful insights from a large quantity of aggregated data instead of achieving perfect classification of all data points [6]. Sentiment mining using corpus based and dictionary based methods for semantic orientation of the opinion words in tweets has been presented by Kumar and Sebastian [13].

In drawing the relevance of twitter sentiment analysis to traffic flow state prediction, He et al. [14] consider improving long-term traffic prediction with tweet semantics; and, then, analyze the correlation between traffic volume and tweet counts with various granularities. Finally, an optimization framework to extract traffic indicators based on tweet semantics using a transformation matrix, while integrating them into the traffic prediction using linear regression is proposed. Real-time traffic improvement by semantic mining of social networks has been captured by Grosenick [15]. Abidin et al. [9] introduce the use of Twitter API to retrieve traffic data serving as input to Kalman Filter models for route calculations and updates while fine-tuning the output for new, accurate arrival estimation.

1.2 Traffic Twitter Cluster Classification

Tweets could have a hashtag which consist of any word that starts with "#" symbol. Hashtags help to search messages containing a particular tag. Also of interest is the Part of Speech (POS) tagging in tweets, which has been applied by Elsafoury [16] to monitor urban traffic status. The main idea of POS tagging, also known as word-category disambiguation, is to mark up a word in a corpus and to assign it to a corresponding POS based on its definition and its context. The former is an example of exact term search while the latter, POS, can be considered a typical example of full-text search, which is usually thorough in its search process but can be more challenging to perform when compared to the exact text search. One instance of such text search is classification of tweets into positive and negative sentiments using multinomial Naïve Bayes' unigram with mutual information based on n-grams and POS that has been presented by Go et al. [11]. It outperforms other classifier approaches under consideration. In between the exact and full-text search is the phrase text search for searching a particular word phrase. For instance, an exact term search might be required to search the term "delay" in a tweet stream. This would bring out only tweets containing the term "delay". On the other hand, a phrase term search could be a phrase like "Traffic delay" in which there are more details of the search term. Phrase text search is often more useful when performing cluster classification than the other text search methods. It is noteworthy that using a particular search operation is based on measuring the relevance of the query to

efficiently match the terms appropriately. Azam et al. [17] present the functional clustering details of their tweets mining approach which has the following steps:

(1) *Tweet crawling*: It is the process of retrieving tweets from twitter server using Twitter Application Program Interface (API). The crawled tweets are stored on local machine for further processing.
(2) *Tweets pre-processing and tokenization*: It involves the filtering of the crawled tweets of non-entirely textual items like emoticons, URL, special character, stop words etc. A common tokenization method known as the n-gram technique can then be applied to tokenize the tweets into bag-of-works ($n = 1$, known as a unigram is recommended for such tweets tokenization by Broder et al. [18]).
(3) *Feature extraction and social network generation*: It is the process of extracting important features from the preprocessed and tokenized tweets while transforming the feature sets into a social network generation comprising a term tweet matrix A of order $m \times n$, where m is the number of candidate terms and n is the number of tweets. The resulting matrix A is used to compute the weight $w(t_{i,j})$ using the following two equations:

$$w(t_{i,j}) = tf(t_{i,j}) \times idf(t_i) \tag{1}$$

$$idf(t_i) = \log \frac{|D|}{\{d_j : t_i \in d_j\}} + 1 \tag{2}$$

where $tf(t_{i,j})$ is the number of times t_i occurs in jth tweet.
$|D|$ is the total number of tweets and $\{d_j : t_i \in d_j\}$ represents the number of tweets with term, t_i. The objective is to normalize matrix A such that the tweet vectors' length equals to 1.
(4) *Social network clustering*: After generating the social network for the complete set of tweets, Markov clustering is used to achieve the social network clustering by crystallizing the network into various cluster each representing individual events. The Markov clustering algorithm (introduced by van Dongen [19]) is a fast and scalable unsupervised cluster algorithm for graphs (also known as *networks*). It serves as an iterative method for interleaving of the matrix expansion and inflation steps based on simulation of (stochastic) flow in graphs.

More details on the abovementioned steps can be found in Azam et al. [17]. For traffic flow prediction using big data analysis and visualization, McHugh [20] considered among other approaches the use of traffic tweets to test the effectiveness of geographical location of vehicles to determine the location of an incident. A useful method that analyzes traffic tweets in order to generate real-time city traffic insights and predictions for traffic management and city planning has been introduced by Tejaswin et al. [21].

2 Using Tweet Traffic Data for Traffic Condition Monitoring

The logs of twitter traffic data for the sentiment analysis and cluster classification were obtained using twitterR package. The tweets were connected to the Twitter API and OAuth authentication was performed using the ROAuth package all in RStudio. The plyr and stringr packages are used to crawl a number of tweets into RStudio while ensuring they are clean of unwanted symbols. More details of this twitter text mining technique can be found in Rais [22]. Detail documentation of the widely used twitter data mining statistical program can be found in cran.r-project.org [23]. We perform a phrase search based on the phrase using a POS tag: *Uk traffic delay*. This is made possible with a simplified phrase search algorithm derived from Eckert [24], with the original simplified version by Manning et al. [25], given by the following:

Algorithm 1

Phrase-Search(index , phrase)

1. $t \leftarrow$Terms(phrase)
2. $k \leftarrow 1$
3. answer $\leftarrow$ Index-Get(index, t)
4. $t \leftarrow$ next(t)
5. while $t \neq$ NIL and answer $\neq \{\}$
6. do nextTweet $\leftarrow$ Index-Get(index, t)
7. answer $\leftarrow$ Forward-Positional-Intersect(answer, nextTweet, k)
8. k $\leftarrow$ k +1
9. t $\leftarrow$ next(t)
10. return answer

In order to apply the above algorithm for our problem, a positional *index* containing a list of a data mined tweets with a list of positions is used to indicate the search phrase. The *Terms* is taking to be a split-normalization tokenizer that splits the *phrase* into list of tokens, normalizing them and assigning its outputs to k as a bag of words. We consider the weighted k-nearest neighbor classifier [26] which assigns a weight $1/k$ to the outputs. This is done by finding the vector of non-negative weights that is asymptotically optimal while minimizing the

misclassification error rate, R_R [26]. Essentially, the asymptotic expansion is needed to ensure strong consistency in the search. This is subject to a regularity class distribution condition:

$$R_R\left(C_n^{wnn}\right) - R_R\left(C^{Bayes}\right) = (B_1 s_n^2 + B_2 t_n^2)\{1 + o(1)\}, \tag{3}$$

Let C_n^{wnn} be the weighted nearest classifier with weights $\{w_{ni}\}_{i=1}^n$ where B_1 and B_2 are constants determined by:

$$\begin{aligned} B_1 &= \int_S \frac{\bar{f}(x_o)}{4\|\dot{\eta}(x_o)\|} dVol^{d-1}(x_o) \\ B_2 &= \int_S \frac{\bar{f}(x_o)}{\|\dot{\eta}(x_o)\|} dVol^{d-1}(x_o), \end{aligned} \tag{4}$$

Vol^{d-1} denotes the natural $(d-1)$ dimensional volume with measure inherent in $S \in \mathbb{R}^d$ while $\bar{f}(x_o)$ denotes the first derivative of the initial point x_o; $s_n^2 = \sum_{i=1}^n w_{ni}^2$ and $t_n = n^{-2/d} \sum_{i=1}^n w_{ni}\{i^{1+\frac{2}{d}} - (1-i)^{1+\frac{2}{d}}\}$ represent variance and squared bias contributions. C^{Bayes} denotes the Bayes classifiers, minimizing the risk over R. Both are given by:

$$\begin{aligned} C_n^{wnn}(x) &= \begin{cases} 1, & if \quad w_{ni\,i=1}^n \geq 1/2 \\ 2, & otherwise \end{cases} \\ C^{Bayes}(x) &= \begin{cases} 1, & if \quad \eta(x) \geq 1/2 \\ 2, & otherwise \end{cases} \end{aligned} \tag{5}$$

Therefore, there is the interpretation that for the point $x \in \mathbb{R}^d, \eta(\mathrm{x})$ belongs to class $C(x)$ with value of 1 in the sense of the weighted nearest neighbor classifier if $w_{ni\,i=1}^n \geq \frac{1}{2}$; and in the sense of the bayesian classifier, if the regression function $\eta(x) = \mathrm{P}(\mathrm{Y} = 1 | \mathrm{X} = \mathrm{x}) \geq \frac{1}{2}$ and; otherwise, both have a value of 2. Further interpretation of the asymptotic behavior towards optimal classification can be found in Samworth [26]. Subsequently, provided that a single term t from the index is not empty based on the resulting *answer* form the positional *index*, we can iterate over the number of incoming tweets while adapting the document list Forward-Position-Intersect algorithm [24, 25] as follows:

Algorithm 2

Forward-Positional-Intersect(p_1 , p_2 , k)

1. answer ← {}
2. while $p_1 \neq$ NIL and $p_2 \neq$ NIL
3. do if tweetId($p1$) = tweetId($p2$)
4. then:
5. $pp_1 \leftarrow$ positions(p_1)
6. $pp_2 \leftarrow$ positions(p_2)
7. while $pp_1 \neq$ NIL and $pp_2 \neq$ NIL
8. do if pos(pp_2) − pos(pp_1) = k
9. then Add(answer, tweetId(p_1), pos(pp_1))
10. $pp_1 \leftarrow$ next(pp_1)
11. $pp_2 \leftarrow$ next(pp_2)
12. elseif pos(pp_2) − pos(pp_1) > k
13. then: $pp_1 \leftarrow$ next(pp_1)
14. else: $pp_2 \leftarrow$ next(pp_2)
15. $p1 \leftarrow$next($p1$)
16. $p_2 \leftarrow$next(p_2)
17. elseif tweetId(p_2) > tweetId(p_2)
18. then $p_2 \leftarrow$next(p_2)
19. else $p_1 \leftarrow$next(p_1)
20. return answer

Re-defining the variables in Eckert [24] let p_1, p_2, pp_1 and pp_2 be the pointers to tweet lists and let p_1 and p_2 reference the tweet lists of the two terms to be intersected while pp_1 and pp_2 reference the inner position lists for each tweet with *tweetId* and *pos* dereferencing the pointers to their actual value in the list. Let *positions* extract the inner position list from an entry in the tweet list. *Add* adds a list identifier and a position to the resulting tweet list. The tweet lists represents the tweets logs of traffic information saved into file.

For our sentiment analysis, we consider the approach of Hu and Liu [27] lexicon of opinion words (LOWs). With our earlier derivations, we posit that the index of sentiments word would require correct interpretation of the word context in relevance to the topic of traffic delay and congestion by scoring the opinion contained in the traffic tweets based on the contextual polarity: positive, negative and neutral. The first method of the improved Naïve Bayes Algorithm (INB-1) by Kang et al. [28] was helpful in computing the score for the crawled filtered traffic tweets based on the following conditional probability:

$$Class(t_i) = \arg\max R_1\left(p_{ij}\right) P\left(c_j\right) \prod_{i=1}^{d} P\left(p_i|c_j\right) \tag{6}$$

$$R_1\left(p_{ij}\right) = \frac{\sum_{p_{ij} \in L_j}^{|L|} C(p_{ij})}{\sum_{p_{ij} \in L}^{|L|} C(p_{ij})} \tag{7}$$

where $Class(t_i)$ denotes the function that determines whether a traffic tweet (t_i) is positive, negative or neural. The probability of class c_j is calculated by $P\left(c_j\right)$ while $P\left(p_i|c_j\right)$ computes the probability that p_i belongs to c_j. $R_1\left(p_{ij}\right)$ denotes the ratio of number of patterns. $C(p_{ij})$ present in the class j of LOWs when the number of patterns $|L|$ is counted over number of patterns $C(p_{ij})$ present in the class j of LOWs when the number of patterns $|L|$ is uncounted. The pattern essentially an n-gram, dwells on the form of $n-1$ Markov model, representing contiguous sequence of n items from a corpus widely known as shingles. We used the Jaccard index to know the extent of similarity between sample sets of shingles irrespective of the ordering. This is given by:

$$J(C_1, C_2) = \frac{|C_1 \ \cap C_2|}{|C_1 U C_2|} \tag{8}$$

$J(C_1, C_2)$ denotes the similarity between set C_1 and C_2. It follows that when item C_1 and C_2 are unrelated then $J(C_1, C_2) = 1$; otherwise $0 \leq J(C_1, C_2) \leq 1$. The cluster formation provide enough evidence to support the interrelations between traffic incidents with regards to the trending causatives of traffic congestions. Furthermore, we employ the term-frequency-inverse-document-frequency, *tdidf* [29] to classify each term in the traffic congestion clusters based on the frequency of occurrence. This is performed by invoking the TF log-normalization with the smooth *tdidf* weight-schemes as follows:

$$tf(t, d) = 1 + \log\left(f_{t,d}\right) \tag{9}$$

$$idf(t, D) = \log \frac{N}{n_t} \tag{10}$$

Such that tweet document term weight is given by:

$$tdidf(t, d, D) = tf(t, d) \cdot idf(t, D) \tag{11}$$

With N = |D| denoting the total number of document in the corpus; $n_t = 1 + |\{d \in D: t \in d\}|$ representing number of times term t appears in document d which belongs to D in the corpus. Notice that the addition of 1 to $|\{d \in D: t \in d\}|$ ensure that infinity value $idf(t, D)$ is avoided.

3 Experimental Evaluation

3.1 Discussion of Results

A sample of 121 tweets were retrieved based on the phrase search UK traffic delay. The data was cleaned of irrelevant symbols. After tweets crawling, preprocessing, tokenization and feature extraction, we obtained the sentiment analysis results as presented in Table 1.

In the time period of obtaining the traffic delay tweets, it was observed that possible severity of 22 were negative sentiments; most likely attributed to serious accidents on the road way (12 negative sentiments). Other relevant phrases are generated in the sentiment analysis such as "serious accidents", "long delays", "looking good", "serious delays" etc. The Jaccard index or similarity and *tdidf* is

Table 1 Traffic twitter sentiment analysis

Phrase	Negative	Neural	Positive	Total
Possible severity	22			22
Serious accident	12			12
Latest	9	3		12
Long delay	1	6		7
Looking good			6	6
Serious delays		6		6
Huge		5		5
Broken down	5			5
Heavy		5		5
Updates		4		4
Emergency	4			4
Blocked		4		4
Main work		4		4
Delays		3		3
Uninjured			3	3
Travel heavy		3		3
Accident	3			3
Update		3		3
Nightmare	2			2
Shocking	2			2
Severe accident	2			2
Bridge congestion delay	2			2
Severe	2			2
Total	69	43	9	121

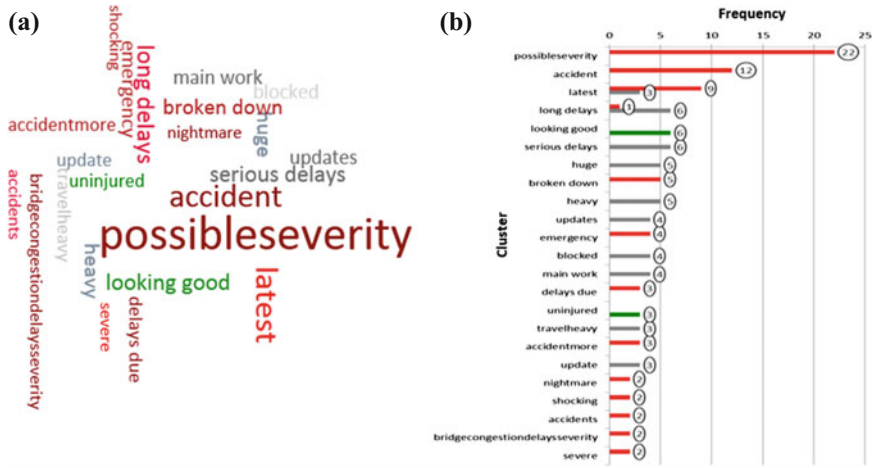

Fig. 1 **a** Traffic delay trending events. **b** Cluster classification index

used to generate the relevant traffic trending events contributing to the cluster classification index as shown in Fig. 1a, b.

3.2 Classification Accuracy

The sentiment classification accuracy of our model is measured in order to determine the performance following the split of the traffic tweet dataset into the training sets (70%) for which the true values are known; validation set (15%) for tuning the classifier during training; and testing set (15%) with unknown values associated with the traffic congestion situation. This is based on the following measures:

$$\text{Accuracy}, a = \frac{\sum TP + \sum TN}{TP_o} \tag{12}$$

$$\text{Precision}, P_r = \frac{TN}{TP + FN} \tag{13}$$

Let TP be the true positive rate denoting the number of the traffic tweets that were correctly identified. TN is the true negative rate denoting the number of traffic tweets correctly rejected; FN be the false negative rate denoting the number of traffic tweets incorrectly rejected; FP be the false positive rate denoting the number of traffic tweets incorrectly accepted; TP_o be the total count of traffic tweets which belongs to a set; P_r be the precision which represents the fraction of the tweets

Table 2 Sentiment classification accuracy

Traffic tweet data sets	Sentiment classification	TP rate	FP rate	Precision
Training	Positive	0.990	0.031	0.882
	Neural	0.987	0.008	0.964
	Negative	0.912	0.049	0.920
Validation	Positive	0.976	0.028	0.905
	Neural	0.989	0.006	0.977
	Negative	0.966	0.045	0.933
Testing	Positive	0.908	0.034	0.855
	Neural	0.955	0.042	0.977
	Negative	0.963	0.055	0.961

relevant to the search query; *a* be the (overall) accuracy which determines the number of correct queries as per the total number of queries. The results show an average accuracy and average precision of 0.95 and 0.91, respectively. Table 2 summarizes the performance of the classifiers for each class under consideration with regards to some clusters associated with the traffic congestion delay.

In the training set, the TP rate yields highest value of 0.990 for the positive sentiment traffic tweet classification with a least value of 0.908 in the testing set for the positive sentiment. The classifier of neural opinion has the least FP of 0.006 in the validation set while its highest value of 0.055 emerges in the testing set for the negative sentiments. The precision yields highest value of 0.977 in the neural sentiment found in the validation and testing set while its least value is in the positive sentiment classification contained in the testing set. We envisage that correctly classifying the traffic congestion based on the twitter sentiments would depend on the location of the user, internet accessibility and tweets time-proximity to the real time the traffic congestion persists with respect to the incident time leading to it.

3.3 Model Validation

To validate the model, the performance of Latent Dirichlet Allocation (LDA) is compared with the model employing the Naïve Bayes and Jaccard similarity with n-gram (JCn-g). The LDA is a typical example of a topic model that can be used for clustering data points; for instance, Azam et al. [17] applied it for clustering of tweets. It is also considered a generative probabilistic model that allows documents to be represented as random mixtures over latent topics characterized by a distribution over words [30]. Table 3 presents the comparative evaluation of JCn-g using unigram and bigram with LDA.

Table 3 Comparative evaluation of JCn-g with LDA

Performance metrics	JCn-g ($n = 1$)	JCn-g ($n = 2$)	LDA
Accuracy	0.871	0.882	0.880
Precision	0.742	0.753	0.762

As observed JCn-g with bigram yields the best accuracy while LDA yields the most precise result. This can be attributed to the fact that LDA not only serves as a generative probabilistic model but also combines it topics interpretability with prior Dirichlet distribution form. Figure 2 presents the cluster generative probabilistic models for the JCn-g and LDA respectively. It shows the data compression of JCn-g (n = 2) and LDA as well as the better similarity between them to buttress our earlier statement. In fact, it can be seen that the green and black tweet clusters are approximately within the same dimensional vector space in the JCn-g (n = 2) and LDA. The best precision observe in LDA becomes obvious from the yellow tweets cluster data points which share same vector space with the JCn-g (n = 1).

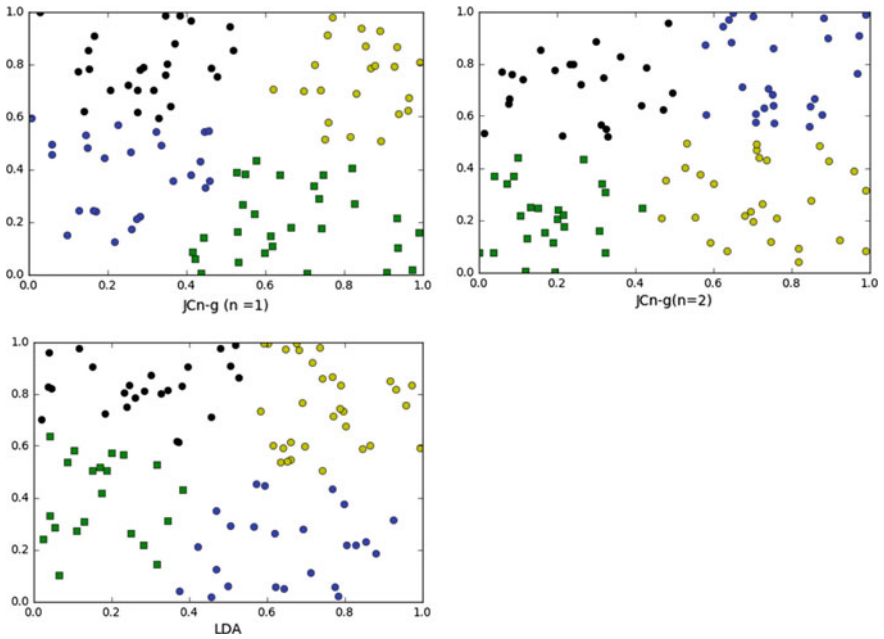

Fig. 2 Tweet cluster generative probabilistic model: JCn-g (n = 1, n = 2), LDA

4 Conclusions and Future Work

Exploring traffic condition using social media data, which can be readily obtained from Twitter, continues to influence traffic information and transportation engineering management decision makers. Applying the proposed data mining techniques on different strata of the UK traffic delay tweets yielded interesting results on traffic congestion, incidents and control.

The validation of JCn-g using LDA shows that the JCn-g with bigram has better accuracy than LDA; however, LDA maintained its high precision over the JCn-g with unigram and bigram. Precious works have suggested that LDA combines its topics interpretability with prior Dirichlet distribution form.

Future work should seek to improve the precision of our cluster classification algorithm. It should seek to improve our preliminary results with a view to seeing if a hybrid approach of the JCn-g with LDA can be more feasible. Also, investigating the reliability for seamless integration with well-known traffic management software system tools should be explored.

References

1. Lu, H-P., Sun, Z., & Qu, W. (2015). Big data-driven based real-time traffic flow state identification and prediction. *Discrete Dynamics in Nature and Society, 2015*, Article ID 284906, 1–11.
2. Villars, R. L., Olofson, C. W., & Eastwood, M. (2011). Big data: What it is and why you should care. IDC.
3. Vlahogianni, E. I, Park, B. B., & van Lint, J. W. C. (2015). Big data in transportation and traffic engineering. *Transportation Research Part C: Emerging Technologies, 58*(Part B), 1–161.
4. Stopher, P. R., & Greaves, S. P. (2007). Household travel surveys: Where are we going? *Transportation Research Part A: Policy and Practice, 41*(5), 367–381.
5. Wang, X., & Li, Z. (2016). Traffic and transportation smart with cloud computing on big data. *International Journal of Computer Science and Applications, 13*(1), 1–16.
6. Philander, K., & Zhong, Y. (2016). Twitter sentiment analysis: Capturing sentiment from integrated resort tweets. *International Journal of Hospitality Management, 55,* 16–24.
7. Korkontzelos, I., Nikfarjam, A., Shardlow, M., Sarker, A., Ananiadou, S., & Gonzalez, G. H. (2016). Analysis of the effect of sentiment analysis on extracting adverse drug reactions from tweets and forum posts. *Journal of Biomedical Informatics, 62,* 148–158.
8. Burscher, B., Vliegenthart, R., & de Vreese, C. H. (2016). Frames beyond words: Applying cluster and sentiment analysis to news coverage of the nuclear power issue. *Social Science Computer Review, 34*(5), 530–545.
9. Abidin, A. F., Kolberg, M., & Hussain, A. (2015). Integrating Twitter traffic information with Kalman filter models for public transportation vehicle arrival time prediction. In M. Trovati, R. Hill, A. Anjum, S. Y. Zhu & L. Liu (Eds.), *Big-data analytics and cloud computing* (pp. 67–82).
10. Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference of International language Resources and Evaluation (LREC' 10).*

11. Go, A., Huang, L., & Bhayani, R. (2009). Twitter sentiment analysis. Stanford University, Stanford California, USA, CS224N - Final Year Project.
12. Wang, J., Gu, Q., & Wang, G. (2013). Potentila power and problems in sentiment mining of social media. *International Journal of Strategic Decision Science, 4*(2), 16–26.
13. Kumar, A., & Sebastian, T. M. (2012). Sentiment analysis on Twitter. *International Journal of Computer Science Issues, 9*(4:3), 372–378.
14. He, J., Shen, W., Divakaruni, P., Wynter, L., & Lawrence, R. (2013). Improving traffic prediction with tweet semantics. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, Beijing, China.
15. Grosenick, S. (2012). Real-time traffic prediction improvement through semantic mining of social networks. Unpublished master thesis. University of Washington, Washington.
16. Elsafoury, F. A. (2013). *Monitoring urban traffic status using twitter messages* (pp. 1–46).
17. Azam, N., Abulaish, M., & Haldar, N. A.-H. (2015). Twitter data mining for events classification and analysis. In *Second International Conference on Soft Computing and Machine Intelligence*.
18. Broder, A. Z., Glassman, S. C., Manasse, M. S., & Zweig, G. (1997). Syntactic clustering of the web. *Computer Networks and ISDN Systems, 29*(8), 1157–1166.
19. van Dongen, S. (2000). *Graph clustering by flow simulation*. Utrecht, Netherlands: University of Utrecht.
20. McHugh, D. (2014). *Traffic prediction and analysis using a big data and visualisation approach*. Ireland: Blanchardstown.
21. Tejaswin, P., Kumar, R., & Gupta, S. (2015). Tweeting traffic: Analyzing Twitter for generating real-time city traffic insights and predictions. In *CODS-IKDD '15*, Bangalore, India.
22. Rais, K. (2014). Twitter analysis.
23. cran.r-project.org. https://cran.r-project.org/web/packages/.
24. Eckert, K. (2008). Simplified phrase search algorithm.
25. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.
26. Samworth, R. J. (2012). Optimal weighted nearest neighbour classifiers. *Annals of Statistics, 40*(5), 2733–2763.
27. Hu, M., & Liu, B. (2004). Mining opinion features in customer review. In *Proceedings of the 19th National Conference on Artificial Intelligence, AAAAI'04*.
28. Kang, H., Yoo, S. J., & Han, D. (2012). Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis. *Expert Systems with Applications, 39,* 6000–6010.
29. Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28*(1), 11–21.
30. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research, 3*(4–5), 993–1022.