Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

ANALYZING EVOLUTION OF RARE EVENTS THROUGH SOCIAL MEDIA DATA

by
Xiaoyu Lu

Recently, some researchers have attempted to find a relationship between the evolution of rare events and temporal-spatial patterns of social media activities. Their studies verify that the relationship exists in both time and spatial domains. However, few of those studies can accurately deduce a time point when social media activities are most highly affected by a rare event because producing an accurate temporal pattern of social media during the evolution of a rare event is very difficult. This work expands the current studies along three directions. Firstly, we focus on the intensity of information volume and propose an innovative clustering algorithm-based data processing method to characterize the evolution of a rare event by analyzing social media data. Secondly, novel feature extraction and fuzzy logic-based classification methods are proposed to distinguish and classify event-related and unrelated messages. Lastly, since many messages do not have ground truth, we execute four existing ground-truth inference algorithms to deduce the ground truth and compare their performances. Then, an Adaptive Majority Voting (Adaptive MV) method is proposed and compared with two of the existing algorithms based on a set containing manually-labeled social media data. Our case studies focus on Hurricane Sandy in 2012 and Hurricane Maria in 2017. Twitter data collected around them are used to verify the effectiveness of the proposed methods. Firstly, the results of the proposed data processing method not only verify that a rare event and social media activities have strong correlations, but also reveal that they have some time difference. Thus, it is conducive to investigate the temporal pattern of social media activities. Secondly, fuzzy logic-based feature extraction

and classification methods are effective in identifying event-related and unrelated messages. Lastly, the Adaptive MV method deduces the ground truth well and performs better on datasets with noisy labels than other two methods, Positive Label Frequency Threshold and Majority Voting.

ANALYZING EVOLUTION OF RARE EVENTS THROUGH SOCIAL MEDIA DATA

by
Xiaoyu Lu

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Helen and John C. Hartmann Department of
Electrical and Computer Engineering

August 2019

APPROVAL PAGE

ANALYZING EVOLUTION OF RARE EVENTS THROUGH SOCIAL MEDIA DATA

Xiaoyu Lu

Dr. Mengchu Zhou, Dissertation Advisor — Date
Distinguished Professor, Department of Electrical and Computer Engineering, NJIT

Dr. Nirwan Ansari, Committee Member — Date
Distinguished Professor, Department of Electrical and Computer Engineering, NJIT

Dr. John D. Carpinelli, Committee Member — Date
Professor, Department of Electrical and Computer Engineering, NJIT

Dr. Qing Liu, Committee Member — Date
Assistant Professor, Department of Electrical and Computer Engineering, NJIT

Dr. Qiong Shen, Committee Member — Date
Co-Founder and CEO, Big Data Machine Learning LLC.

BIOGRAPHICAL SKETCH

Author: Xiaoyu Lu

Degree: Doctor of Philosophy

Date: August 2019

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2019
- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2015
- Bachelor of Science in Electrical Engineering and Automation,
  Nanjing University of Technology, Nanjing, P. R. China, 2011

Major: Electrical Engineering

Presentations and Publications:

X. Lu, M. C. Zhou, A. C. Ammari, and J. C. Ji, "Hybrid Petri Nets for Modeling and Analysis of Microgrid Systems," *IEEE/CAA Journal of Automatica Sinica*, 2016, 3(4): 347-354.

X. S. Lu, M. C. Zhou, L. Qi and H. Y. Liu, "Clustering Algorithm-based Evolution Analysis of Rare Events by Using Social Media Data," *IEEE Transactions on Computational Social Systems*, 2019, 6(2): 301-310.

X. S. Lu, M. C. Zhou, and K. Y. Wu, "A Novel Fuzzy Logic-based Text Classification Method for Tracking Rare Events on Twitter," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019. (Accepted)

W. H. Han, X. S. Lu, M. C. Zhou, X. H. Shen, J. X. Wang, and J. Xu, "An Evaluation and Optimization Methodology for Efficient Power Plant Programs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017. DOI: 10.1109/TSMC.2017.2714198, early access.

X. S. Lu, M. C. Zhou, and L. Qi, "Analyzing Temporal-spatial Evolution of Rare Events by Using Social Media Data," in *Proceedings of 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Banff, Canada, Oct. 5-8, 2017, pp. 2684-2689.

X. S. Lu, and M. C. Zhou, "Analyzing the Evolution of Rare Events via Social Media Data and k-means Clustering Algorithm," in *Proceedings of 2016 IEEE 13th International Conference on Networking, Sensing, and Control (ICNSC)*, Mexico City, Mexico, Apr. 28-30, 2016, pp. 1-6.

X. S. Lu, M. C. Zhou, H. Y. Liu, and L. Qi, "A Comparative Study on Two Ground Truth Inference Algorithms based on Manually Labeled Social Media Data," in *Proceedings of 2019 IEEE 16th International Conference on Networking, Sensing, and Control (ICNSC)*, Banff, Canada, May 9-11, 2019, pp. 436-441.

K. Y. Wu, M. C. Zhou, X. S. Lu, and L. Huang, "A Fuzzy Logic-based Text Classification Method for Social Media Data," in *Proceedings of 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Banff, Canada, Oct. 5-8, 2017, pp. 1942-1947.

H. Han, M. C. Zhou, X. S. Lu, and Y. J. Zhang, "Multi-layer Feature Histogram with Correlative Degree for Cross-camera-based Person Re-identification," in *Proceedings of 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Banff, Canada, Oct. 5-8, 2017, pp. 1322-1327.

H. Y. Liu, M. C. Zhou, X. S. Lu, and C. Yao, "Weighted Gini Index Feature Selection Method for Imbalanced Data," in *Proceedings of 2018 IEEE 15th International Conference on Networking, Sensing, and Control (ICNSC)*, Zhuhai, China, Mar. 27-29, 2018, pp. 1-6.

Dedicated to the people I love and the people that love me.

ACKNOWLEDGMENT

Undergoing this PhD has been a tremendous treasure during my career and has led me to a new stage in my life. This amazing and unforgettable experience helped me grow up. Indeed, this whole process would not have been possible without the support and guidance that I received from many people.

My deepest gratitude is to my dear advisor: Dr. Mengchu Zhou. You have been a tremendous mentor for me. I would like to thank you for leading my research, cultivating my skills, and encouraging me to enjoy the life. Your advice on my research as well as my career has been invaluable. Without you this dissertation would have not been achievable.

I also give many thanks to Dr. Qiong Shen who brought me into the fields of machine learning, data mining, and computational intelligence. To my committee members Dr. Nirwan Ansari, Dr. John D. Carpinelli, and Dr. Qing Liu for their valuable time, advice and encouragement.

I would like to thank my family members, especially my parents, Ms. Yuhua Wu and Mr. Dong Lu, and my cousin, Ms. Shang Gao, for their support during my PhD career. I would like to express my appreciation to the group members of Discrete Event Systems Laboratory: Dr. Liang Qi and Ms. Haoyue Liu with whom I have collaborated with my research, conducted insightful discussions, and undertook difficulties for many years. I also want to thank my good friends: Dr. Xilong Liu, Mr. Jingchu Ji, Mr. Wenhan Lu, Mr. Di Wu, Mr. Keyuan Wu, Dr. Lianghua He, Dr. Qinghua Zhu, Mr. Hongjie Liu, Ms. Chenwei Zhao, Dr. Xiwang Guo, Dr. Qi Kang, Dr. Hua Han, Mr. Yi Wang, Mr. Zhihao Zhao, Mr. Sibo Zhang, Mr. James Zhang, and many others, who gave me strength and encouragement over the past four years. I also want to thank the faculty and staff members in our Department of Electrical and Computer Engineering for their support throughout my doctoral studies.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF TABLES
(Continued)

Table **Page**

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

Events are occurring over the world all the time, and as the main part of the world, people cannot be ignored and isolated from the events. People's ideas, feelings, and altitudes describe the characteristics and attributes of an event from multiple angles and perspectives. Laituri and Kodrich treat people as sensors that can help to build a rapid response database [58]. Sheth involves people in a citizen-sensor network that refers to an interconnected network of people who actively observe, report, collect, analyze, and disseminate information via text, audio, or video messages [86]. With the profound development of Internet, communications and networking, mobile devices, and computers, exchanging information among people becomes rapid, efficient and accurate. Social media as a part of interactive Web 3.0 provides users with a simple and convenient channel to share their observations, feelings, altitudes and views. Consequently, social media occupies a crucial position in human life and receives a high level of attention [72]. This allows people, companies, and organizations to create, share, broadcast, and exchange various information in virtual communities and networks; the information covers important events, ideas, and human attitudes at a specific time span. Different from the traditional paper-based or industrial media, the advantages of social media contain quality, reach, frequency, usability, immediacy, and permanence [7] [101]. Mobile technologies and social platforms provide a path for people to post their messages anytime and anywhere. This leads a way to analyze event-related information such as the relationship between happiness and mobility patterns [32], and tourist origins and attractions [11, 65, 100]. Thus, a citizen-sensor network via social media connects people together and perceives the occurrences around the world.

A disaster is viewed as a disruption on the earth and involves environmental and economic loss. A serious disaster may greatly threaten human beings' and animals' lives and property safety. It is treated as a rare event, since it occurs rarely but has really serious destructions. Disasters are described as social events in [76]. A deeper concept that any physical events alone does not constitute disasters unless they negatively affect human beings and social systems is presented in [93]. Thus, a disaster is not an isolated event and its crisis arises because of its caused the vulnerability of human beings, natural environment and technological systems [21]. Chen *et al.* [20] emphasize that a focus on social disruptions is a key to understand and assess a disaster. Their work connects the physical disasters and human beings' social activities. Figure 1.1 shows the relationship between the real world and virtual one. The former may impact the latter, because the latter may be struck by the former. In the opposite direction, the virtual world characterizes the real event in the real world. For example, if a wind storm passes, people may post photos and videos regarding some phenomena and its damages, such as trees' falling down or high waves near shores. This kind of information characterizes how strong wind storm is by human beings' real observations, feelings, and attitudes. Thus, if we are able to find a temporal-spatial pattern that shows how a real event impacts social media and how social media characterizes the event, we can definitely help people understand the event better and assist relevant departments of government to cope with and evaluate the real event.

Preis *et al.* [74] compare the number of Hurricane Sandy-related photos with the atmospheric pressure data. The real variations of atmospheric pressure are defined as the evaluation of Hurricane Sandy in the real world. Guan and Chen [39] calculate their proposed metric, disaster-related ratio (DRR), during the occurrence of Hurricane Sandy and confirm a close connection between the activities on social media and the extent of disruptions related to the hurricane. However, their work only

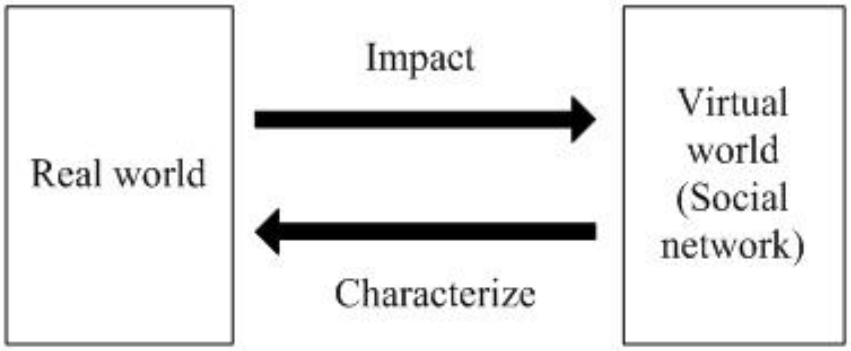

Figure 1.1 Relationship between the real world and virtual one.

calculates a few days' DRRs in specific cities. Its DRR curves can only describe the roughly impacted date by the disaster. The error rates are high since their work finds only the peak dates and cannot get more accurate time points. Preis *et al.* [74] use an hour as the time granularity, but cannot obtain more accurate time points than an hour. No matter how Preis *et al.* and Guan *et al.* choose the time granularities, once the time granularity is chosen, the time span is separated into fixed time intervals. It is easy to ignore the intensity of instances in a time domain, because the intervals are set subjectively in advance. For example, there are a lot of posted messages around a time point, but some of them may belong to an earlier time interval while others belong to a later one. In this case, they are cut into two time-intervals, and then the intensity of them is broken and reduced. Thus, finding proper time intervals, not fixed ones, is very important. Such intervals tend to be different. In order to conquer this issue and increase the accuracy in our study, we adopt a clustering algorithm that is able to focus on the intensity of posted messages in a time domain, automatically assign the messages into corresponding classes, and then find more accurate time when the social media is impacted by a rare event, e.g., Hurricane Sandy. Thus, the time intervals are automatically selected based on the intensity of data points. In addition, the time difference between a rare event's occurrence and the peak of social

media data intensity reveals the difference between the real and virtual worlds in a time domain.

Meanwhile, many studies, such as [39, 63, 74], investigate the relationship between social media data and a rare event like Hurricane Sandy in 2012. Yet, they use only the key words to distinguish whether a message is related with it or not. Such an approach can easily miss the messages that contain no key words but are actually closely related to the event. For example, "No power" and "No school" are two very short messages that were posted on the arrival date of Hurricane Sandy. Since keywords do not contain them, they are not selected to conduct hurricane-related message analysis. The incompletely extracted messages may lead to an inaccurate estimation of the relationship between social media and rare events, especially for computing DRR in [39]. Thus, accurately extracting rare-event-related tweets is imperative. Then, this problem is converted to a binary classification problem. In other words, a tweet is identified as either a rare-event-related instance or unrelated one.

Conventional text classification mainly focuses on text documents and divides them into predefined categorizes or classes. Researchers have proposed many text classification algorithms and methodologies. For example, [88] and [49] propose a Naive Bayes-based approach. The study [114] presents a term frequency-inverse document frequency (TF-IDF) technique. A combination of a regular classifier and heuristic algorithm is deeply discussed in [15, 19, 75]. The above studies mainly focus on a document or paragraph that has a large number of sentences and words with abundant information [35]. A short-text classification problem is different from the conventional one. For example, a tweet from Twitter has a limited number of characters. Usually, a tweet is short and has a couple of sentences, or only a couple of words. Many short texts exist around a human's life and in a variety of forms, such as blogs, image captions, and short message service (SMS) messages. In addition, short

texts use oral formats, and often ignore those syntax structures and grammar. To address this issue, some studies, such as [107] and [108], treat each individual word as a research object. In detail, a complete sentence is split into words. It may break the original meaning of a sentence, and ignore some phrases and oral words. Other researchers [12,33,34,44,69,80,82,103,105] adopt a semantic enrichment approach. It searches similar information, concepts and contents via web search engines. It enriches the short texts by adding more features from external resources. Nevertheless, it has some noise, such as meaningless and useless words derived from a search engine, which may reduce classification efficiency. In [22], a Tweet2Vec is proposed based on a character composition model and converts each tweet as a vector. It adopts a Bi-directional Gate Recurrent Unit (Bi-GRU) neural network for learning tweet representations. In [68], Word2Vec is proposed by giving word representations in a vector space. Two log-linear models, continuous Bag-of-Words model (CBOW) and Continuous Skip-gram model (Skip-gram), are created and pay an attention to continuous words. By using deep learning methods, each word is represented by a high dimension vector. Both Tweet2Vec and Word2Vec represent a short-text and a word by using a high dimension vector, respectively. It takes a lot of space complexity and tends to increase computing time greatly. In addition, the methods provide each short text with a numerical vector, and the vagueness and ambiguity of a text are completely ignored. The vector contains exact values, but it cannot reflect the vagueness and ambiguity of text well, because sometimes the meanings of texts are not obvious and cannot simply be represented by a numerical vector. Thus, we aim at taking the vagueness and ambiguity into consideration.

Fuzzy logic can deal well with vagueness and ambiguity and is a technique close to human thinking [55]. One important contribution of fuzzy logic is its superiority in computing with words. In more detail, fuzzy logic provides a way to convert people's words and thinking into proper numerical values that can be handled by

computers and artificial intelligence with the concern of vagueness and ambiguity. Zadeh *et al.* [106] claim that no other method serves this purpose. In this work, a fuzzy-logic based text classification method is proposed to avoid the disadvantages mentioned above. The features are directly from human beings' natural language and thinking. They not only consider each individual written word, but also some fixed phrases and oral words in our real life. All the features are obtained directly from the original texts. Thus, no extra information or noise is introduced to impact the effectiveness of a classifier. The fuzzy logic-based method contains two parts: feature extraction and classification. The former one extracts features from a short text by using membership functions. The later one classifies the short-texts based on the fuzzy rules and defuzzification methods. The extracted features, variables and parameters of membership functions, and fuzzy rules are obtained according to human beings' empirical knowledge and subjective understanding.

Even though social media data are helpful to understand and analyze the evolution of rare events, the obvious weakness of using them does exist. They lack ground truth. In other words, many short texts do not have any true labels, namely ground truth. Note that in our cases, short texts are classified into binary classes, i.e., rare-event-related and unrelated classes. Then, the ground truth corresponds to the two classes. In some studies, such as [102] and [97], they choose hashtags as their ground truth. Yet many do not have such hashtags. But Korolov *et al.* [54] indeed claim that they are not fit for all messages because only one-third of messages contain hashtags and they are often inconsistent. Even worse, hashtags may still increase redundant information and noise. For example, "*#sandycantstopme Don't let her stop you.*", where "*#sandycantstopme*" is the hashtag of this text. Even if there is no space among words in the hashtag, people can still understand it quite well. However, it is quite difficult and challenging for machines to understand, since "sandycantstopme" is not a correct English word and not a normal phase. Furthermore, different people

have different understandings of a same sentence. These aforementioned issues suggest that understanding and analyzing short texts are not easy tasks. Moreover, finding the ground truth of short-texts is more difficult and, most of time it is impossible. Acquiring the true meaning from a poster is difficult and time-consuming. There are also some methods, such as TF-IDF [112] and word2vec [61], can work on a text classification problem, but sometimes they do not perform well on the short-text classification. It is because there are much fewer words in short-texts than regular articles. Additionally, there is an imbalance problem, since even though many users post a huge number of rare event-related messages, a much higher percentage of posted messages are unrelated. Thus, in this work, we focus on social media data, i.e., short-texts, and bring human being's intelligence into their classification process.

We ask some labelers to label the data and synthesize the labels as ground truth. Note that in some work, such as [111], this process is called learning from crowdsourced labelers and sometimes such systems are called crowdsourcing systems. One of the basic strategies of a crowdsourcing system is to vote. Usually, the minority is subordinated to the majority. It is also called majority voting in such a system. However, obviously, this strategy fails in many cases. It is still possible that the majority has to be subordinated to the minority. Thus, many methods and strategies are proposed to deal with different kinds of problems. Some of them focus on investigating the consistency of labelers and tasks [99] [77]. In their assumption, labeling a few tasks should be consistent when a labeler labels them and the difficulty of labeling a task is consistent. The studies [110] and [109] concentrate on discovering the pattern of labeled data by labelers. They deeply analyze the data, explore their distribution, and then make final decisions. In order to deduce the ground truth, we first compare four ground truth inference algorithms while dealing with the short-text classification problem. They are Majority Voting (MV), Positive Label Frequency Threshold (PLAT), Generative Model of Labels, Abilities, and Difficulties (GLAD),

and Ground Truth Inference using Clustering (GTIC). Then we propose an adaptive majority voting method and compare it with MV and relatively better method, PLAT.

This work covers three core aspects of rare event analysis via social media data. They are 1) exploring the evolution of rare events, 2) classifying short-texts and 3) deducing ground truth of short-texts. Its contributions have four parts. First, it verifies that there is a strong connection between the real world and virtual one. Second, by using our proposed method and finding proper time intervals, we can deduce the temporal evolution of a rare event like Hurricane Sandy and confirm that the time difference does exist and varies for different cities. Then, a novel feature extraction approach and a fuzzy logic-based classification method are proposed to cope with the short-text classification problem. Lastly, ground truth inference algorithms that deduce the ground-truth of short texts are compared and a new one called Adaptive MV is proposed.

CHAPTER 2

LITERATURE REVIEW

The social media data-based analysis of evolution of rare events contains three major study directions: temporal-spatial analysis of rare events, understanding the meanings of contents and deducing the ground truth of short texts. This section reviews the related work, respectively.

2.1 Temporal-Spatial Analysis of Rare Events

Many researchers have analyzed and investigated the rare event called Hurricane Sandy by using social media data. There are two major categories of their interests. The first one investigates the awareness and moods of human beings during Hurricane Sandy [25, 28, 45] in a temporal or spatial domain. They rely on natural language processing, machine learning and semantic analysis. The studies [25, 45] uncover the changes of human reactions and awareness during Hurricane Sandy. As the hurricane unfolds, influential users are identified, topical changes are observed, and the community evolvement is demonstrated by using the spectral clustering algorithm in [45]. Caragea *et al.* [19] exploit a combination of bag of words and sentiment features such as emoticons, acronyms, and polarity clues. Then a support vector machine (SVM) is used to classify the tweets into three classes, i.e., positive, neutral and negative moods. With the geo-tags of tweets, they map these tweets as points into a global map and observe that the mean center of tweets shifts accompanying with the movement of Hurricane Sandy. Ediger *et al.* [28] deal with a large volume of data in real time. Their proposed platform identifies the immediate and critical information that increases situational awareness during Hurricane Sandy.

Another category aims at finding the relationship and connections between social media data and a rare event [39, 63, 74]. The main idea is to study the

temporal-spatial patterns of both. Then they compare the pattern of the former in the virtual world with the real meteorological data during a disaster. Preis *et al.* [16] compare the number of Hurricane Sandy-related photos with the atmospheric pressure data recorded among meteorological stations. The variations of atmospheric pressure are defined as the evaluation of Hurricane Sandy. A correlation coefficient is used to verify whether there is any relationship between social media and Hurricane Sandy or not. Choosing a reliable metric is important to estimate the influence of an event. It is less meaningful to just count the total number of event-related messages during a specific time span as discussed in [39, 116]. A metric pioneered in [39] named DRR replaces the number of messages and illustrates the relationship between a disaster and social media activities. It calculates the ratio between the numbers of related and unrelated messages at a same time span in the same area. If a topic is discussed many times and has a high percentage of attention among other topics, this denotes that more people pay much attention on it. Thus, DDR is more useful than only counting the number of disaster-related messages. Guan and Chen [39] calculate the proposed DRRs during Hurricane Sandy and confirm a close connection between the activities on social media and the extent of disruptions related to Hurricane Sandy. The time is closer to the landed time of Hurricane Sandy and the location is closer to the coast while higher DRRs are obtained in the temporal-spatial pattern. In addition, since both studies [39] and [74] cannot obtain the accurate impacted time point of the hurricane in the virtual world, they are not able to find the existence of time difference between the hurricane's occurrence and peak of social media data volume. In other words, they fail to discover that there is a difference in a time domain for a rare event between the real world and virtual one.

2.2 Short-Text Classification

In general, text classification aims to assign text documents into predefined categorizes or classes. Researchers have proposed a variety of text classification algorithms and methodologies, such as the Naive Bayes-based approach [49, 88], term frequency-inverse document frequency (TF-IDF) technique [114] and a combination of a regular classifier and a heuristic algorithm [19, 75]. Spielhofer *et al.* train a Naive Bayes classifier for relevant data detection by suggesting that the irrelevant data removal and noise reduction are similar to the email spam filtering [88]. Jiang *et al.* introduce a deep feature weighting Naive Bayes by using the maximum likelihood estimation to calculate prior and conditional probabilities [49]. Zhang *et al.* propose an improved TF-IDF method for text classification by using stemming and lemmatization techniques. They adopt synonymous techniques to reduce computational complexity [114]. When dealing with text documents with a massive size, Latent Semantic Indexing (LSI) is the best for comparing both TF-IDF and multi-word methods [112]. Caragea *et al.* [19] propose a sentiment classification method by using a SentiStrength algorithm combined with a Support Vector Machine (SVM) and Naive Bayes classifier. Prusa *et al.* use Convolutional Neural Networks (CNN) and a new encoding approach for text classification [75]. Note that though CNN is mainly used for image processing, text data can be converted into an image with an encoding method such that CNN can be used as a text classifier. The conventional text classification focuses on the document or paragraph classification that have a large number of sentences and words with abundant information [87]. However, short-text classification is different from the conventional one, due to its limited number of characters. Some studies [108] and [107] treat each individual word as a research object and use LSI to deal with it. Other researchers search similar concepts online, find the semantic similarity between unlabeled and labeled texts, and link them to some explicit semantic information derived from external resources or web search engines [12,33,34,44,69,80,82,103,105].

This type of approaches is called semantic enrichment, since it enriches short texts to better their classification. Sathe *et al.* propose a novel method by using a Neural Network (NN) for sentiment classification combined with fuzzy logic [81]. Fuzzy logic is used to deal with symbolic and vague information, which builds a fuzzification matrix for NN to use. Besides, text summarization and intelligent tagging can be handled by utilizing fuzzy logic [89]. Those studies mentioned above motivate our work.

2.3 Ground-truth Inference Algorithms

Social media data obtain more and more attention because of their important role in the analysis of the evolution of rare events. Classifying them into classes accurately is a key process to unfold humans' social activities. However, the obvious weakness of using social media data is the lack of ground truth. In other words, many short texts do not have any labels that tell a poster's real meaning, e.g., related to a rare event or not, i.e., ground truth. Acquiring the true meaning from a poster is difficult and time-consuming. Thus, automatically deducing ground truth is imperative. Ground truth inference algorithms are proposed to deal with this issue.

In general, a ground-truth inference algorithm should satisfy two conditions [111]. One is that it infers integrated labels for instances at least. The other is that it does not depend on any additional information, such as no historical labeling qualities, features, and true labels of instances. In other words, an integrated label of an instance is determined by the labels that are given by a few human labelers. In addition, since different people have different experience, background and education, noisy labels do exist. Thus, for each instance, integrating a few labels into a finalized one is not easy. In our case, as discussed in Section 2.1, short-text labeling may not be easier than and can be even worse than the labeling issue in biological and medical fields, as in [18] and [95]. Furthermore, the labeling quality depends on the text

comprehension and interpretation ability of labelers. Some words adopted by a user may not be popular, some are shortened from a particular environment and some are very professional that common people do not know them unless they are in the field. In order to deal with this issue, a straight-forward direction is to improve label quality. There are two categories of approaches. One derives from a data collection phase. It focuses on designing a quality-controllable labeling task. Its idea is to design some mechanisms to train and guide labelers to provide high quality labels [8] and [27]. Its defect is the complexity and difficulty in designing a perfect labeling task. It heavily relies on the background and historical information [111]. Also, training and guiding labelers with different background and experience are not easy and can take much time.

Another direction is to improve the quality of labels after data collection. It conquers the defects of previous one by using ground-truth inference algorithms. It contains two steps: repeated labeling and integrated labeling. The former requests labels given by multiple labelers while the latter adopts some proper mechanisms that integrate the given labels as an estimated label. This estimated label is potentially to be the true one. Our work focuses on the second category to deduce the true label. Currently, ground-truth inference algorithms fall into two categories. One is based on an Expectation-Maximization approach (EM) and the other is based on linear algebra and statistics. For the former one, the representative studies are reported in [77] [98] and [99]. These methods model either the behaviors of labelers or difficulties of examples or both. Then, they use Bayesian estimation and maximize a likelihood function to obtain estimated labels. Even though EM-based algorithms are widely used, they have many defects [109] [113]. First, they may converge to a local optimal solution instead of a global one. Second, choosing initial values of parameters is not easy. Different initial values tend to produce different results. Meanwhile, a dataset may not exactly fit the probability distribution assumed by

these algorithms. Last, its convergence speed is uncertain and depends on the data and initial parameter setting. These uncertain reasons lead the execution time to vary toward the longer end. The representative methods of the latter can be found in [53] [52] [109] and [110]. Karger *et al.* [53] [52] propose a method based on the reliabilities of labelers by using a belief propagation-like method. The disadvantage is that it is not a standard inference method based on a generative probabilistic model. Thus, it is difficult to extend to more complex models or real-world datasets. PLAT [110], based on statistics, counts the number of positive labels and dynamically searches an optimal threshold. GTIC [109] is based on Bayesian statistics and works on multi-class problems. Furthermore, there are two obvious issues that are not considered in many studies: biased labeling and imbalance data issues. Both of them need to be further investigated. Note that biased labeling is a common case. It does exist not only because of different labelers, but also depending on judgment criteria when labelers perform labeling tasks.

CHAPTER 3

CLUSTERING ALGORITHM-BASED EVOLUTION ANALYSIS OF RARE EVENTS BY USING SOCIAL MEDIA DATA

In order to focus on the intensity of instances, we propose a clustering algorithm-based data processing method in this chapter. It analyzes the evolution of rare events in both temporal and spatial domains. First, three k-based clustering algorithms are introduced. Next, the clustering algorithm-based data processing method is proposed. Then, the definition of time difference and the selection of the number of clusters are discussed. Finally, the social media data collected in the virtual world and the meteorological data in the real world during Hurricane Sandy 2012 are utilized to verify the effectiveness of the proposed method.

3.1 *K*-based Clustering Algorithms

Clustering algorithms, such as hierarchical and k-means clustering algorithms, aim to discover the natural groupings of patterns, points, or objects [48]. Kang *et al.* [51] introduce clustering algorithms that divide a given dataset into multiple classes according to data similarity. This section first introduces the classical k-means clustering algorithm. Then, it is followed by its extension, i.e., k-means++ and k-MWO where MWO represents mussel wandering optimization.

3.1.1 *k*-means Clustering Algorithm

About 60 years ago, the k-means clustering algorithm, called k-means for short, was proposed. Its simplicity, efficiency and easy implementation make it one of the most popular clustering methods [48, 107]. It has been successfully used in many fields. For example, studies [60, 92] adopt it and its extensions in the texture and image

segmentation. Oyelade, *et al.* [73] utilize it to predict students' academic performance. The work [14] adopts it for customer management. It is formally described as follows:

Let $X = \{x_i\} \subset R^d$, $i \in \{1, 2, ..., n\}$, be the set of n d-dimensional points where R^d denotes a d-dimensional real number set, and $C = \{C_k\}$, $k \in \{1, 2, ..., K\}$, be a set of K clusters that partition X where $K > 0$ is a positive integer. The mean of cluster C_k is defined as:

$$\mu_k = \frac{1}{|C_k|} \sum_{x \in C_k} x \tag{3.1}$$

where $|C_k|$ is the cardinality of C_k. The squared error between μ_k and the points in C_k is defined as:

$$\phi(C_k) = \sum_{x \in C_k} ||x_i - \mu_k||^2 \tag{3.2}$$

The objective function is given as:

$$J = \sum_{k=1}^{K} \sum_{x \in C_k} ||x_i - \mu_k||^2 \tag{3.3}$$

It computes the sum of squared errors over all K clusters. The goal of k-means is to find the minimized sum of squared errors over all K clusters, i.e.,

$$J_{min} = min(\sum_{k=1}^{K} \sum_{x \in C_k} ||x_i - \mu_k||^2) \tag{3.4}$$

Its main steps are as follows [47, 48]:

1. Select an initial partition with K clusters;

2. Compute a new partition by assigning each point to its closest cluster center;

3. Compute new cluster centers according to (3.1); and

4. Repeat Steps 2 and 3 until the objective function J reaches its minimum value, J_{min}.

3.1.2 *k*-means++ Clustering Algorithm

The work [10] extends the k-means clustering algorithm and proposes the k-means++ algorithm. Initially, every data point can be chosen as a center with the following probability:

$$P(x) = \frac{Dist(x)^2}{\sum_{x \in X} Dist(x)^2} \tag{3.5}$$

where $Dist(x)$ is the shortest distance from a data point x to the closest center that has already been chosen. Usually, $Dist(x)$ is computed based on Euclidean distance. The steps of this algorithm are described as follows:

1.1. Choose first center C_1 uniformly at random from X;

1.2. Take a new center C_k by choosing $x \in X$with probability obtained from (3.5);

1.3. Repeat Step 1.2. until K centers are found;

2. Compute a new partition by assigning each point to its closest cluster center;

3. Compute new cluster centers according to (3.1); and

4. Repeat Steps 2 and 3 until the objective function J reaches its minimum value.

Note that Steps 2-4 are the same as the standard k-means algorithm mentioned in Section 3.1.1.

3.1.3 *k*-MWO Clustering Algorithm

k-MWO is a new clustering method based on swarm intelligence. It is proposed in [51] and is as good as a k-PSO (particle swarm optimization) method. It combines mussel wandering optimization (MWO) with the classical k-means. As a new heuristic method, MWO is inspired by mussels' leisurely locomotion behavior when they form bed patterns in their habitat [9]. It is an ecologically inspired optimization algorithm and mathematically formulates a landscape-level evolutionary mechanism of the distribution pattern of mussels through a stochastic decision and Levy walk. In [51], each mussel $Y_i = (y_{i1}, y_{i2}, ..., y_{iK})$ represents a set of centers of K classes,

where y_{ik}, $k \in \{1, 2, ..., K\}$, is the coordinate vector of the center of the k-th class of the i-th mussel. The algorithm first initializes N_N mussels, and then evaluates each mussel's fitness by using a squared sum error (SSE) as follows:

$$E = \sum_{i=1, x_i \in C_k}^{M_k} \sum_{k=1}^{K} ||x_i - \mu_k||^2 \tag{3.6}$$

Based on the fitness values, the top $\eta\%$ mussels are used to update their position coordinates during the next generation. The learning process from mussels with the top fitness values guide the evolution to better directions. The updating process is accomplished dimension by dimension. When updating, a Levy walk, between 0 and 1, is calculated to decide mussels' displacement. The new position should not be beyond the limits which avoid the mussels going to an unsuitable field. The detailed steps of k-MWO are given as follows:

1. Initialize N_N mussels, i.e., $Y_i = (y_{i1}, y_{i2}, , y_{iK})$ where $i \in \{1, 2, ..., N_N\}$;

2. Calculate the fitness of each mussel via (3.6) where x_i denotes the i-th data point in a dataset, M_k is the number of data points in every class. C_k is the center of the k-th class and μ_k is its mean. where K is the number of classes, and $k \in \{1, 2, ..., K\}$. Note that Y_i is associated with one set of centers $U = \{\mu_k\}$.

3. Find the top $\eta\%$ mussels that have the best fitness and calculate their center y_g.

4. Update mussels' positions: calculate each mussel's Levy walk $l_i = \gamma[1 - \lambda]^{-1/(\rho-1)}$, where ρ is a shape parameter with $1.0 < \mu < 3.0$, λ is a randomly sampled value from the uniform distribution $[0, 1]$, and γ denotes the walk scale factor, which is a positive real number; then update its position via $y'_{ik} = y_{ik} + l_i(y_g - y_{ik})$.

5. Calculate the fitness of the updated mussels, find the new top $\eta\%$, and update y_g;

6. Examine if it satisfies the termination criterion. If so, output the best result; and otherwise, go to Step 4 to start the next iteration.

3.2 k-means Clustering Algorithm-based Data Processing Method in Spatial Domain

In this section, our analysis of a rare event pays attention to the spatial domain. The k-means clustering algorithm divides the hurricane impacted region into several sub-regions based on the intensity of tweets. A disaster-related ratio (DRR) performs as a metric and denotes the impact degree of Hurricane Sandy towards each sub-region. First, the k-means-based data processing method in the temporal domain is described. Next, experimental results are given and followed with their analysis and discussions.

Since Hurricane Sandy stormed through our selected region over time, we expect to divide up the area into several sub-regions and study those small ones. k-means provides a method that can cut and combine those nearest tweets. Thus, this section describes a data processing method based on k-means clustering. Because some tweets were posted too early or too late before or after Sandy landed, they are filtered and deleted in our filtered dataset. Figure 3.1 describes the procedure of data processing that clusters tweets in the spatial domain. The first step clusters the close tweets into spatial clusters based on their locations or geo-coordinates and obtains the coordinate mean of each cluster's centroid. Let $C^S = \{C_1^S, C_2^S, ..., C_i^S, ..., C_U^S\}$, $i \in \{1, 2, ..., U\}$, be a set of spatial clusters that partition set D into $U \geq 2$ spatial clusters. U is an integer and represents the number of clusters. Each element $C_i^S \in C^S$ represents an individual spatial cluster. The mean of a spatial cluster C_i^S is denoted with u_i^S. In a physical meaning, u_i^S also denotes a pair of geo-coordinates.

We specify the northeast region of the United States as our concerned region. It contains some states with a large population, such as New Jersey and New York, and some large cities, such as Boston, New York City and Washington D.C. This region was badly impacted by the hurricane and brought us a sufficiently large disaster-related dataset. Temporally, our study's time period spans from Oct 27, 2012, when

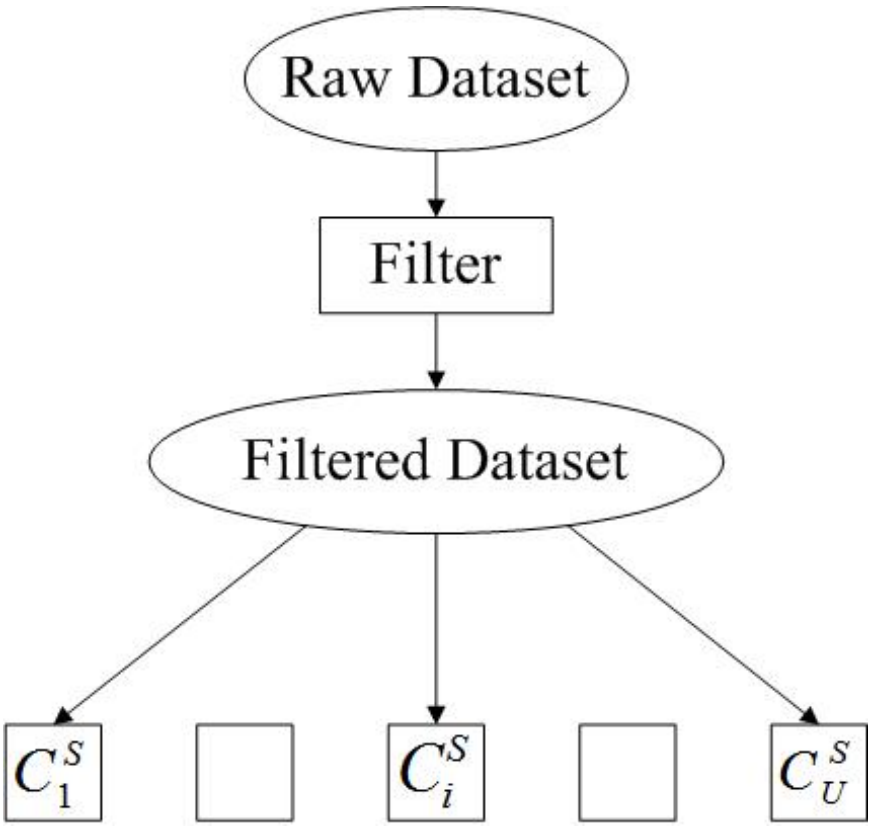

Figure 3.1 k-means based data processing method in the spatial domain.

the storm warning was issued, to Nov 7, 2012, a week after the hurricane landed in the selected region. Meanwhile, spatial geo-coordinates are limited by the latitudes from $37.84°$ to $42.86°$ and the longitudes from $-70.89°$ to $-78.8°$. After this filtering, it returns us about $1,281,000$ tweets. Then, we use keywords to filter out those disaster-unrelated tweets. These keywords are "Sandy", "hurricane" and "storm" as also used in [39]. This step returns about 74,000 tweets that are related to Hurricane Sandy.

Based on the procedure in Figure 3.1, the first step adopts the k-means clustering algorithm with parameter $k = 50$. This step partitions the disaster-related tweets into 50 clusters in the spatial domain. If a cluster's DRR is high, it means that the corresponding physical area is highly impacted by Hurricane Sandy and vice versa. In Figure 3.2, we use a point to represent the geo-coordinates of a cluster's center and the digital number next to it is its identifier. If the digital number is small, it represents that the DRR of its corresponding cluster is large; otherwise, it is small. Figure 3.3 shows the DRR values of the points in Figure 3.2.

In both Figures 3.2 and 3.3, points are marked with colors, red, blue, and green, that representing the corresponding clusters' DRRs are greater than 0.05, between 0.03 to 0.05, and less than or equal to 0.03, respectively. The two values, 0.05 and 0.03, are specified as thresholds and partition 50 points into 3 levels. Red points are the highly impacted regions; blue ones are moderately impacted regions; and green ones are slightly impacted ones. This matches Hurricane Sandy's impact pattern with four levels: very high (purple), high (red), moderate (yellow) and low (green), given by Federal Emergency Management Agency (FEMA) as shown in Figure 3.4.

In Figure 3.4, very high (purple) area means that greater than $10,000$ of county population was exposed to the surge; high (red) one indicates that $500 - 10,000$ of county population was exposed, or modeled wind damages were greater than $100M$, or high precipitation (> 8"); moderate (yellow) one represents that $100 - 500$ of

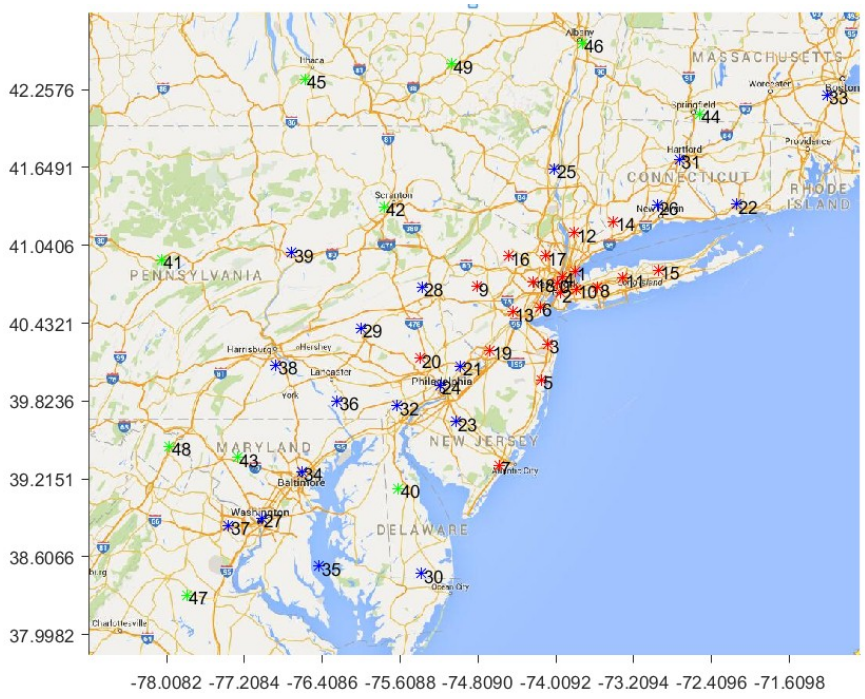

Figure 3.2 Hurricane Sandy impacted pattern with identifiers.

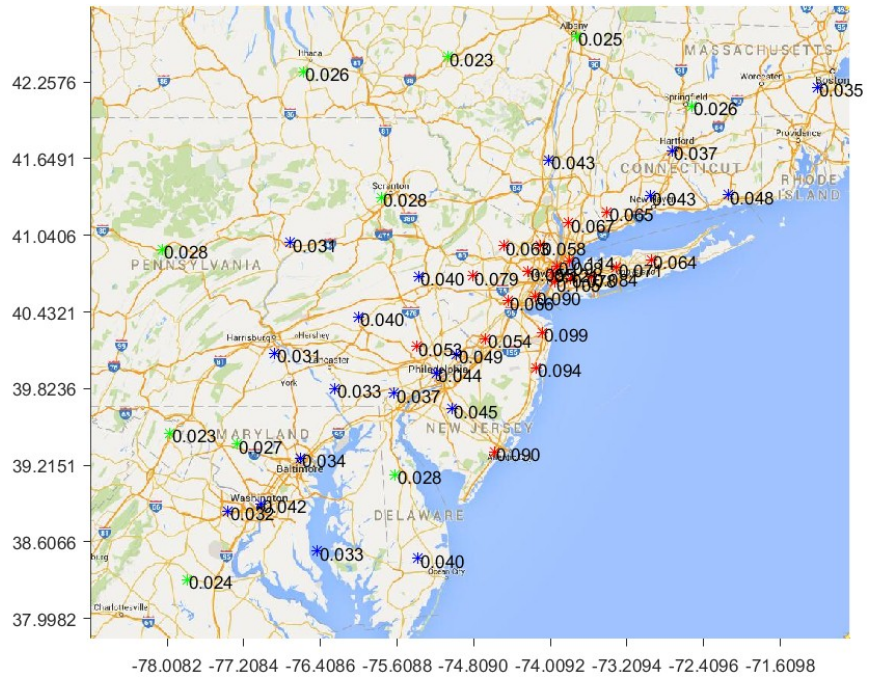

Figure 3.3 Hurricane Sandy impacted pattern with DRRs.

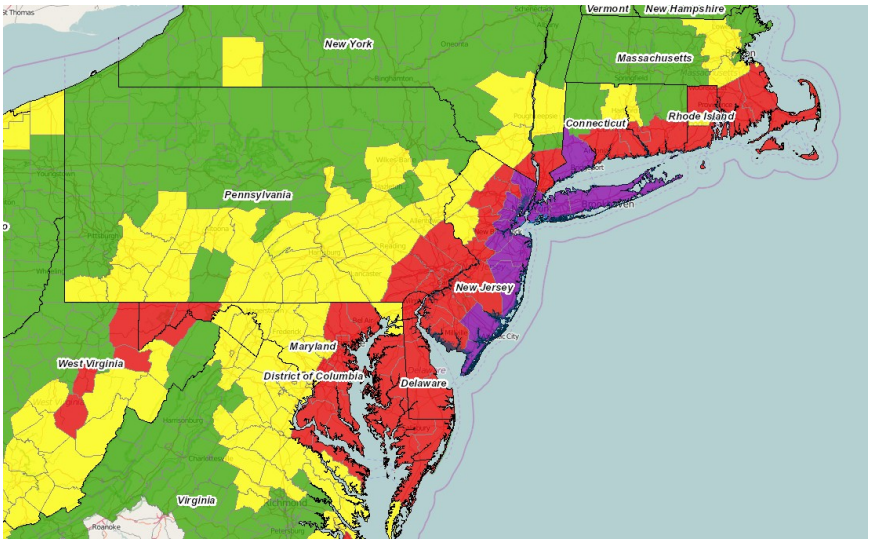

Figure 3.4 FEMA Hurricane Sandy impact analysis. [1]

county population was exposed, or modeled wind damages were between 10M and 100M, or medium precipitation (4" to 8"); and low (Green) one indicates that there were no surge impacts.

3.3 Clustering Algorithm-based Data Processing Method

In this section, a clustering algorithm-based data processing method is proposed. The flow chart is given in Figure 3.5. Three clustering algorithms mentioned in Section 3.1 constitute the main part of our proposed method. Each of them is adopted individually. The detailed descriptions of the method are described next.

Let $D = \{x_n\} \subset R^d$, $n \in \{1, 2, ..., N\}$, be the set of N d-dimensional points where R^d denotes a d-dimensional data set and N is a positive integer. $r = (r_1, r_2, ..., r_N)$ is an $1 \times N$ vector, where $r_n \in \{0, 1\}$, $n \in \{1, 2, ..., N\}$ is a binary variable indicating whether an instance is related with a rare event ($r_n = 1$) or not ($r_n = 0$). Note that if an instance is related with a rare event, it means that this instance has a good chance to contain information about the event. On the contrary, this instance has no relation with the event. Then the raw data set D is indicated by r, and is separated into two sets denoted by X_α and X_β, respectively, where X_α

represents the data set of all rare-event-related instances and X_β represents the set of remaining ones. For each data point x_n, there exists a corresponding set of binary variables $z_{nk} \in \{0, 1\}$, where $k \in \{1, 2, ..., K\}$ describes which of the K clusters x_n is assigned to. Thus, if x_n is assigned to cluster k, then $z_{nk} = 1$ for $j = k$, and $z_{nj} = 0$ for $j \neq k$. Note that Z is an $N \times K$ matrix. In order to distinguish the rare-event-related and unrelated ones, two more sets of binary indicators α_{nk} and β_{nk} are adopted. Note that $A = \{\alpha_{nj}\}$ and $B = \{\beta_{nj}\}$. If x_n is a rare-event-related one assigned to cluster k, then $\alpha_{nk} = 1$ for $j = k$; otherwise, $\alpha_{nj} = 0$, where $j \in \{1, 2, ..., K\}$ and $j \neq k$. Similarly, if x_n is a rare-event-unrelated one and it is assigned to cluster k, then $\beta_{nk} = 1$ for $j = k$; otherwise, β_{nj} is 0 for $j \neq k$. Thus, $Z = A + B$. Next, we define an objective function:

$$J = \sum_{k=1}^{K} \sum_{x \in C_k, r_n = 1} ||x_n - \mu_k||^2 \tag{3.7}$$

It represents the sum of the squares of the distances of each rare-event-related data point to its corresponding center μ_k. Our goal is to obtain two sets, i.e., $\{z_{nk}\}$ and $\{\mu_k\}$ such that J is minimized. Thus, in our data processing method, we substitute (3.3) with (3.7). After all rare-event-related instances are assigned into K clusters, all unrelated ones are assigned into these clusters by finding the shortest distance between a data point and a center. A rare-event-related data point x_n is assigned into its closest cluster center, and then α_{nk} is computed as:

$$\alpha_{nk} = \begin{cases} 1 & if\ r_n = 1\ and\ k = argmin_j ||x_n - \mu_j||^2 \\ 0 & otherwise \end{cases} \tag{3.8}$$

Since clustering algorithms focus on only rare-event-related data points, β_{nk} is not changed. Then Z is updated by $Z = A + B$. The mean of cluster C_k is computed as:

$$\mu_k = \frac{\sum_n r_n \alpha_{nk} x_n}{\sum_n r_n \alpha_{nk}} \tag{3.9}$$

After clustering is complete, β_{nk} is computed as:

$$\beta_{nk} = \begin{cases} 1 & if\ r_n = 0\ and\ k = argmin_j ||x_n - \mu_j||^2 \\ 0 & otherwise \end{cases} \tag{3.10}$$

Meanwhile, Z is updated by $Z = A + B$. DRR_k is computed as:

$$DRR_k = \frac{\sum_n \alpha_{nk}}{\sum_n z_{nk}} \tag{3.11}$$

where DRR_k denotes the DRR of the k-th cluster. The flow chart of proposed data processing method is given in Figure 3.5. The steps of proposed method are described as follows. Initially, each instance is labeled by 0 or 1. An instance is labeled as 1

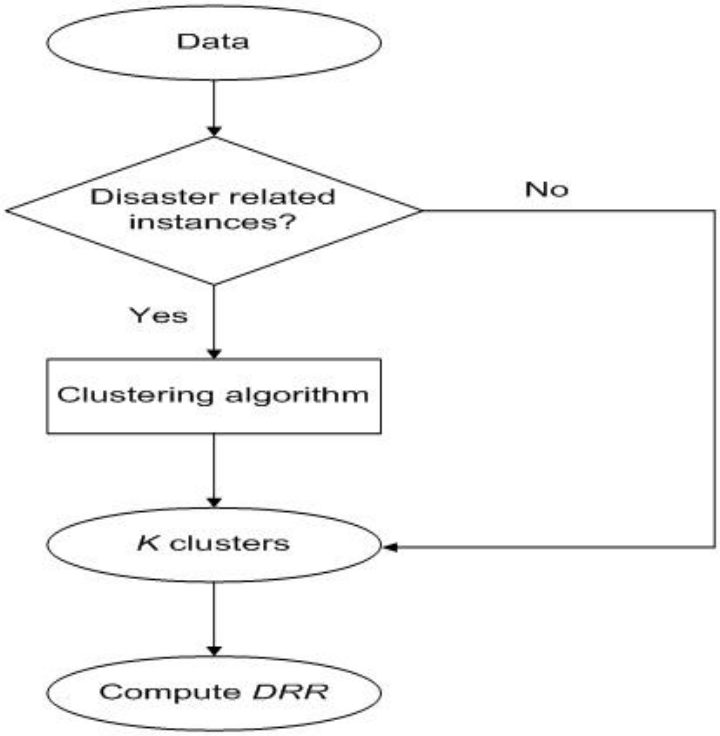

Figure 3.5 Proposed data processing method.

if it is related to the specific rare event; otherwise, it is 0. Then, vector r associated with binary values is obtained. In this work, a keyword search method is adopted to distinguish instances. Thus, by searching predefined keywords, it identifies the

rare-event-related and unrelated ones from the original dataset. Next, one of the three following updated clustering algorithms groups the rare event-related ones.

3.3.1 *k*-means and *k*-means++ based Data Processing Methods

The k-means based data processing method is given as follows:

1. Randomly generates K clusters and obtain an initial partition;

2. Assign every data point into its nearest cluster by (3.8);

3. Update the cluster centers by using (3.9); and

4. Steps 2 and 3 are repeated until J in (3.7) reaches its minimum value.

Finally, when the clustering is complete, rare-event-unrelated ones can be divided into K new clusters via (3.10). After that, $Z = A + B$. Then, DRR is calculated and obtained via (3.11). Note that Steps 1-4 only focus on rare-event-related instances.

The k-means++ based data processing method is similar to the one based on k-means. The only difference is that k-means++ chooses the initial centers with a probability according to (3.5) [10].

3.3.2 *k*-MWO based Data Processing Method

The detailed steps of k-MWO based data processing method are given as follows:

1. Initialize N_N mussels, i.e., $Y_i = (y_{i1}, y_{i2}, .., y_{ik}, ...y_{iK})$ where $i \in \{1, 2, , N_N\}$ and $k \in \{1, 2, ..., K\}$.

2. Compute the fitness of each mussel by using (3.7). α_{nk} is calculated via (3.8). Where x_n, $n \in \{1, 2, .., N\}$, is the n-th data point and C_k is the center of the k-th class and μ_k is its mean. Y_i corresponds to one set of centers $U = \{\mu_k\}$, where $k \in \{1, 2, ..., K\}$, in (3.7) and (3.8).

3. Obtain the best fitness and search the top $\eta\%$ mussels, and then compute the center y_g.

4. Update the position of mussels by calculating each mussel's Levy walk; and then

update the mussel's position by $y'_{ik} = y_{ik} + l_i(y_g - y_{ik})$.

5. Calculate the new mussels' fitness, search their top $\eta\%$ ones, and update y_g;

6. Check whether the termination criterion is reached or not. If yes, return the best one; otherwise, go back to Step 4 and continue to the next iteration.

Finally, when the clustering is complete, rare-event-unrelated ones are partitioned into K new centers via (3.10). After that, $Z = A + B$. Then, DRR is calculated and obtained via (3.11). Note that our Levy walk adopts $l_i = \gamma[1 - \lambda]^{-1/(\rho-1)}$, where $1.0 < \rho < 3.0$ is a shape parameter. The walk scale factor λ is a positive real number and is randomly generated from the uniform distribution $[0, 1]$. In fact, k-MWO generates some mussels and uses them as centers. Then, its evolutionary mechanism updates those mussels and searches the best centers that minimize the objective function.

3.3.3 Time Difference

The study of time difference plays a vital role in understanding and revealing the relationship between the virtual and real worlds in a time domain. It reflects the precedence order between the two worlds. Understanding the time difference is able to help broadcast warnings and predict the severity of an event in advance. Thus, a time difference is proposed and adopted to evaluate the approach regarding the hurricane in the time domain. In this work, the time difference is defined as the time point associated with the peak of DRR curve minus the time of the arrival of hurricane. If it is a negative value, it represents that the DRR reaches its peak a little later than the arrival of hurricane, namely, a lag time difference. Otherwise, it is called a lead time difference, which denotes that the DRR reaches its peak earlier than the arrival of hurricane. Note that the minimum air pressure and the maximum wind speed are assumed as the sign of the hurricane's arrival at a given region. Thus,

their corresponding time points, to be described in Section 3.4, denote the time of the hurricane's arrival.

3.3.4 Selection of the Number of Clusters

Even though, k-means, k-means++, and k-MWO are different, the selection of the Number of Clusters value is same. In the real world, users usually post more messages during the daytime and relatively fewer at deep night when very few activities are ongoing. The intensity of messages thus varies. By using clustering algorithms, the centers of clusters move towards the high intensity of messages. Hence, centers should be obtained during the daytime or at earlier night. Then, the cluster count corresponds to the number of days when the data are collected. However, because of the impact of rare events, the regularities may be broken, especially for those rare events that occur at deep night and last for a long time. Thus, determining a proper the Number of Clusters is difficult. Yet this value should be around the number of days during which the data are collected.

3.4 Dataset and Experimental Results

This section describes the experimental results, illustrates the feasibility of the proposed data processing method, and compares our results with the real data obtained from National Oceanic and Atmospheric Administration (NOAA). To evaluate the effectiveness of our proposed data processing method, our experimental results are compared with the real world meteorological data. Low air pressure and strong wind speed can represent the arrival of a hurricane [74]. The correlation coefficient, called Kendall's τ, is used to verify the effectiveness of our proposed method. First, we introduce our social media and meteorological data. Next, we analyze our experimental results.

3.4.1 Dataset

Twitter, created and launched in 2006, is a well-known online social media platform that enables users to post maximum 280-character messages called tweets. In 2016, there were over 319 million active users monthly. Thus, it provides sufficient social media data that is widely used in various research areas [17, 35, 96, 117]. The data used in this section focuses on three large cities in the United States as the concerned regions: the capital of the United States–Washington D.C.; the global power city–New York City (NYC), NY; and a large seaport–Baltimore, MD. Those tweets are filtered by specifying the spatial region and time range. In the spatial domain, the location of a weather station that is the nearest one to the geographical center of each city is specified as the center of each city. The buffer distances are set to 19.65, 8.72 and 7.51 km for New York City, Baltimore and Washington D.C., respectively, as same as that in Guan's work [39]. Note that the buffer distance represents the real geographical radius of a city. In the temporal domain, the time period spans from Oct. 27, 2012, when the storm warning was issued, to Nov. 7, 2012, a week after the hurricane landed in the specific region.

In total, more than 289,000 tweets were crawled via Twitter's Application Programming Interface (API). Each tweet has five columns as features: identifier, geographic coordinates (longitude and latitude), posted time, and contents. Then keywords are used to define whether a tweet is related to Hurricane Sandy or not. If a tweet contains at least one of the following keywords: "Sandy", "hurricane" and "storm" [39], then we regard it as related to Hurricane Sandy. This step returns about 27,000 rare-event-related tweets. In order to compute time points conveniently and consider the time zone, the tweets' posted time is converted into seconds and all time points are converted into Greenwich Mean Time (GMT). Since the dataset is filtered and retains the tweets posted from Oct. 27 to Nov. 7, 2012, the starting time is set at 00:00:00, Oct. 27. Two special dates, Oct. 29 and 30, are the dates

when Hurricane Sandy touched this selected area and a day right after its arrival, respectively. Then 190,800, 277,200 and 363,600 seconds are used to denote the two dates.

3.4.2 Experimental Results

The Kendall's τ measures the difference between two variables and is adopted here to evaluate the feasibility and effectiveness of proposed methods [16]. Mathematically, if τ approaching -1 or $+1$, there is a strong correlation between the two variables; otherwise, the two variables have less correlation if τ is close to 0. In addition, a p-value is companying with each τ value and associated with a hypothesis testing. This process indicates whether the two variables have a significant difference or not. It also means that even though a τ value is 1, if the p-value is greater than a significance level [6], we still need to accept that the two variables do not have any strong correlation. On the contrary, if a p-value is less than a significance level, the corresponding τ value is named as a satisfied τ value. Normally, the significance level is 0.05.

In our cases, depending on the posting time of all tweets, they are grouped into K classes. In the time domain, this helps us analyze the evolution of an event for each specific city. We select different K values and compare their results. Tables 3.1-3.9 give the average and variance of τ values regarding our three specific cities by using the proposed methods. K values are chosen as 5, 10, 15, 20, and 50. For each city and each K value, each method is executed 200 times. Then, if τ value is close to -1, it represents that the experimental results have correlation with the meteorological data. Then, in Tables I-IX, the satisfied τ values are put in a bold font. In order to keep these values simple in the tables, each value uses three numbers only after the decimal point. In other words, for example, if a value is 0.7001, it is written as 0.700. Note that if it is a value smaller than 0.001, it is written as 0.000. In other words, it is very small but not necessarily a real value 0. We now show the results

of experiments obtained versus the changes of the air pressure. For each specific city, there exists at least one satisfied τ value that is slightly greater than or less than -0.6. Even some of them are less than -0.7. Note the τ value greater than -0.6 and less than -0.8 indicates that two variables have moderate correlation. For each specific city and each clustering algorithm, all highest τ values are obtained when K equals 10, 15 or 20. For each city, among three clustering algorithm-based methods, the best satisfied τ values are obtained by the k-means++ based method, because its best τ values are less than those of other two methods. The k-MWO-based and k-means-based methods obtain roughly the same results. In other words, in some cases, k-MWO-based method performs better than k-means-based one, but in other cases, k-MWO-based method performs worse than k-means-based one. If we focus on the comparisons with the variant of wind speed, only a few of τ values are satisfied with the constraint that p-value is less than 0.05 and they are much less than the τ values obtained by air pressure. It concludes that the social media data from the virtual world has a relationship with meteorological data in a real world. It clearly means that social media activities are associated with the disaster in the real world. The k-means++ based method is the best among three compared ones. Due to the uncertain and random wind speed changes , the relative stable variant of air pressure is better than wind speed for the comparisons between the virtual world and the real one.

Table 3.1 Correlation of Experimental Results with STP and Wind Speed for Washington DC by Using k-means++

		K=5		K=10		K=15		K=20		K=50	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
STP	avg.	-0.800	0.083	**-0.732**	0.002	**-0.700**	0.000	**-0.683**	0.000	**-0.665**	0.000
	var	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000
Wind speed	avg.	0.527	0.333	0.288	0.364	0.354	0.107	0.278	0.105	**0.362**	0.001
	var	0.000	0.000	0.020	0.031	0.007	0.009	0.001	0.002	0.001	0.000

Table 3.2 Correlation of Experimental Results with STP and Wind Speed for NYC by Using k-means++

		K=5		K=10		K=15		K=20		K=50	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
STP	avg.	-0.800	0.083	**-0.674**	0.009	**-0.735**	0.000	**-0.742**	0.000	**-0.676**	0.000
	var	0.000	0.000	0.000	0.000	0.001	0.000	0.001	0.000	0.000	0.000
Wind speed	avg.	0.316	0.633	0.163	0.584	0.260	0.215	0.296	0.104	**0.290**	0.010
	var	0.000	0.000	0.000	0.001	0.002	0.005	0.005	0.004	0.002	0.000

Table 3.3 Correlation of Experimental Results with STP and Wind Speed for Baltimore by Using k-means++

		K=5		K=10		K=15		K=20		K=50	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
STP	avg.	-0.800	0.083	**-0.725**	0.003	**-0.657**	0.000	**-0.667**	0.000	**-0.597**	0.000
	var	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000
Wind speed	avg.	0.600	0.233	0.419	0.139	**0.421**	0.046	**0.444**	0.011	**0.449**	0.000
	var	0.000	0.000	0.008	0.005	0.003	0.001	0.002	0.000	0.001	0.000

Table 3.4 Correlation of Experimental Results with STP and Wind Speed for Washington DC by Using k-means

		K=5		K=10		K=15		K=20		K=50	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
Air pressure	avg.	-0.800	0.083	**-0.674**	0.007	**-0.630**	0.001	**-0.609**	0.000	**-0.540**	0.000
	var	0.000	0.000	0.001	0.000	0.000	0.000	0.001	0.000	0.001	0.000
Wind speed	avg.	0.748	0.125	0.524	0.068	**0.644**	0.004	**0.664**	0.000	**0.568**	0.000
	var	0.010	0.007	0.011	0.005	0.009	0.000	0.005	0.000	0.001	0.000

Table 3.5 Correlation of Experimental Results with STP and Wind Speed for NYC by Using k-means

		K=5		K=10		K=15		K=20		K=50	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
Air pressure	avg.	-0.800	0.083	**-0.688**	0.007	**-0.685**	0.000	**-0.695**	0.000	**-0.658**	0.000
	var	0.000	0.000	0.005	0.000	0.002	0.000	0.002	0.000	0.001	0.000
Wind speed	avg.	0.280	0.646	0.280	0.646	**0.445**	0.036	**0.498**	0.012	**0.506**	0.000
	var	0.017	0.002	0.017	0.002	0.004	0.001	0.009	0.001	0.002	0.000

Table 3.6 Correlation of Experimental Results with STP and Wind Speed for Baltimore by Using k-means

		K=5		K=10		K=15		K=20		K=50	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
Air pressure	avg.	-0.776	0.101	**-0.644**	0.010	**-0.580**	0.002	**-0.600**	0.000	**-0.464**	0.000
	var	0.004	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000
Wind speed	avg.	0.464	0.403	0.485	0.078	**0.457**	0.029	**0.536**	0.002	**0.446**	0.000
	var	0.009	0.014	0.005	0.006	0.004	0.001	0.001	0.000	0.002	0.000

Table 3.7 Correlation of Experimental Results with STP and Wind Speed for Washington DC by Using k-MWO

		K=5		K=10		K=15		K=20		K=50	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
Air pressure	avg.	-0.651	0.195	**-0.675**	0.008	**-0.674**	0.000	**-0.658**	0.000	**-0.579**	0.000
	var	0.008	0.004	0.002	0.000	0.001	0.000	0.000	0.000	0.001	0.000
Wind speed	avg.	0.458	0.416	0.334	0.271	0.387	0.095	0.341	0.067	**0.406**	0.000
	var	0.012	0.016	0.016	0.034	0.012	0.011	0.005	0.005	0.002	0.000

Table 3.8 Correlation of Experimental Results with STP and Wind Speed for NYC by Using k-MWO

		K=5		K=10		K=15		K=20		K=50	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
Air pressure	avg.	-0.597	0.240	**-0.551**	0.036	**-0.588**	0.003	**-0.583**	0.000	**-0.527**	0.000
	var	0.003	0.003	0.004	0.001	0.002	0.000	0.001	0.000	0.001	0.000
Wind speed	avg.	0.257	0.778	0.462	0.104	**0.425**	0.049	**0.449**	0.016	**0.517**	0.000
	var	0.058	0.114	0.009	0.008	0.005	0.002	0.005	0.001	0.002	0.000

Table 3.9 Correlation of Experimental Results with STP and Wind Speed for Baltimore by Using k-MWO

		K=5		K=10		K=15		K=20		K=50	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
Air pressure	avg.	-0.792	0.089	**-0.709**	0.005	**-0.695**	0.000	**-0.659**	0.000	**-0.583**	0.000
	var	0.002	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Wind speed	avg.	0.580	0.261	0.406	0.152	**0.518**	0.010	**0.456**	0.009	**0.426**	0.000
	var	0.006	0.013	0.007	0.007	0.000	0.000	0.002	0.000	0.002	0.000

Figures. 3.6 and 3.7 show the curves of DRR by using k-means-based method versus air pressure and wind speed for Washington D.C., respectively. Washington D.C. is shown here and regarded as an example. For other cities and methods, the figures are similar. Three black dotted vertical lines distinguish Oct. 29 and 30, 2012, as two specific dates. Figure 3.6 shows that the air pressure reaches its peak, i.e., the minimum value, on Oct. 29 when the hurricane touched these cities. Furthermore, around the time when the hurricane strikes the cities, the air pressure decreases sharply and then increases. After a short period, the air pressure gradually restores to a normal status. Its tendency of variation is similar to the curve of DRR we obtained before. In Figure 3.6, the curve of DRR increases sharply in the beginning, but then gradually decreases and approaches 0 in a few days after the arrival of the hurricane. Since many factors, such as the angle of wind, impact the measurement of wind speed, the speed changes more frequently and sharply than the air pressure. Figure 3.7 shows that the wind speed changes sharply. The maximum value of wind speed is found on Oct. 29. At the same time, the maximum DRR values appear on the same day as well. Clearly, the curves of DRRs and wind speed have the very similar tendency. In other words, both wind speed and DRR grow from a low value to its peak sharply, and drop back to a low one gradually.

If we have a close view of Figure 3.6, we discover that a short time difference exists between the peak of DRR and the peak of air pressure. Since many rare-event-related tweets are posted slightly earlier than the arrival of the hurricane, the short time difference is supposed to be derived.

Tables 3.10-3.18 concern the time differences. We can do the comparisons among our meteorological data and experimental results. If a τ value is a satisfied one in Tables 3.10-3.18 and is less than the corresponding value in Tables 3.1-3.9, they are put in a bold face. Let us use the comparisons in Tables 3.1 and 3.10 for Washington D.C. as an example. When $K = 15$ and the air pressure data is compared with, in

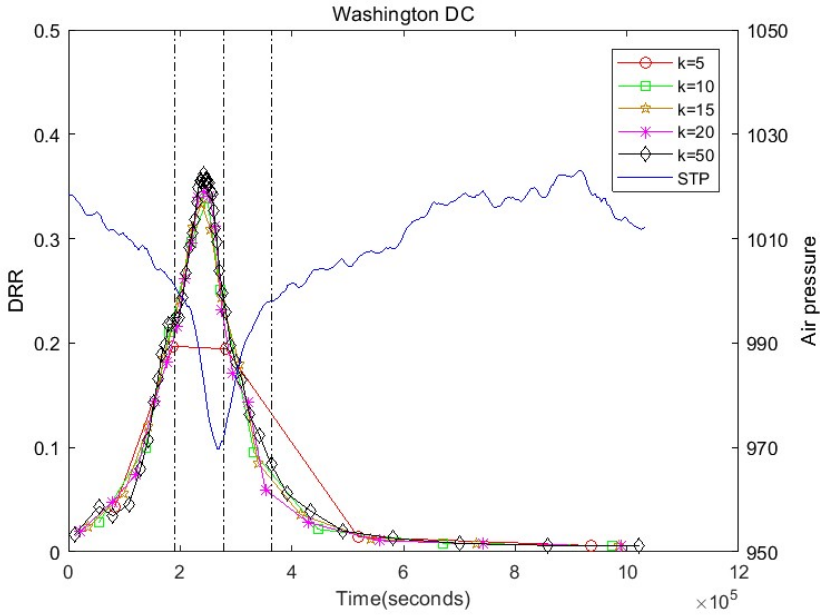

Figure 3.6 DRR curve vs. air pressure for Washington, D.C.

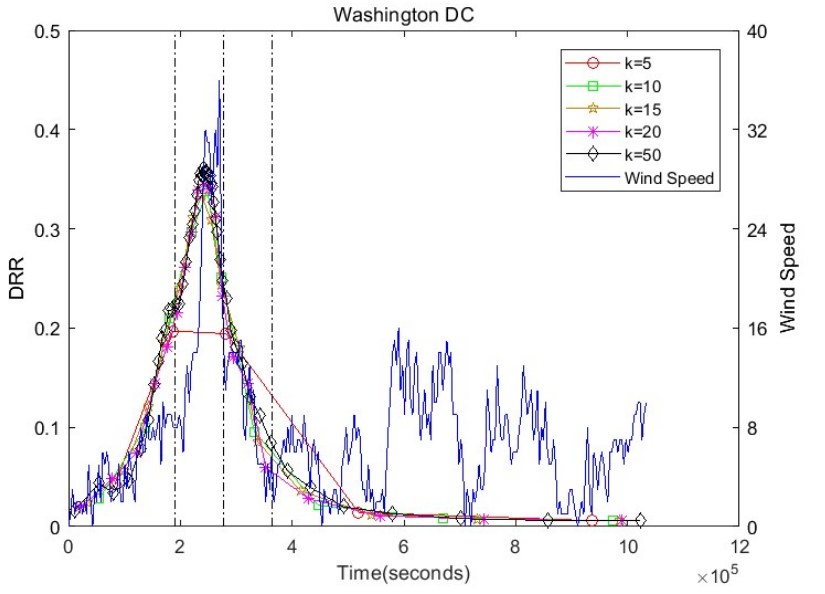

Figure 3.7 DRR curve vs. wind speed for Washington, D.C.

Table 3.10 Correlation of Time Difference Considered Experimental Results with STP and Wind Speed for Washington D.C. by Using k-means++

		K=5		K=10		K=15		K=20		K=50	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
STP	avg.	-0.600	0.233	**-0.827**	0.000	**-0.810**	0.000	**-0.785**	0.000	**-0.767**	0.000
	var	0.000	0.000	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000
	Time Difference	18428 s		**34477 s**		**24934 s**		**23197 s**		**27764 s**	

Table 3.11 Correlation of Time Difference Considered Experimental Results with STP and Wind Speed for NYC by Using k-means++

		K=5		K=10		K=15		K=20		K=50	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
STP	avg.	-0.600	0.233	**-0.688**	0.005	**-0.769**	0.000	**-0.745**	0.000	**-0.684**	0.000
	var	0.000	0.000	0.000	0.000	0.001	0.000	0.001	0.000	0.000	0.000
	Time Difference	-26504 s		**-2988 s**		**12575 s**		**3421 s**		**2820 s**	

Table 3.12 Correlation of Time Difference Considered Experimental Results with STP and Wind Speed for Baltimore by Using k-means++

		K=5		K=10		K=15		K=20		K=50	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
STP	avg.	-0.601	0.233	**-0.824**	0.001	**-0.758**	0.000	**-0.747**	0.000	**-0.661**	0.000
	var	0.000	0.000	0.002	0.000	0.000	0.000	0.001	0.000	0.001	0.000
	Time Difference	15900 s		**31602 s**		**20780 s**		**22456 s**		**15653 s**	

Table 3.13 Correlation of Time Difference Considered Experimental Results with STP and Wind Speed for Washington D.C. by Using k-means

		K = 5		K = 10		K = 15		K = 20		K = 50	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
Air pressure	avg.	-0.800	0.083	**-0.830**	0.000	**-0.832**	0.000	**-0.791**	0.000	**-0.722**	0.000
	var	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.003	0.000
	Time Difference	8121 s		**24011 s**		**28305 s**		**26316 s**		**25646 s**	

Table 3.14 Correlation of Time Difference Considered Experimental Results with STP and Wind Speed for NYC by Using k-means

		$K=5$		$K=10$		$K=15$		$K=20$		$K=50$	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
Air pressure	avg.	-0.614	0.223	**-0.732**	0.003	**-0.704**	0.000	**-0.704**	0.000	**-0.675**	0.000
	var	0.003	0.001	0.003	0.000	0.002	0.000	0.002	0.000	0.001	0.000
	Time Difference	-22051 s		**8356** s		**3244** s		**1855** s		**3889** s	

Table 3.15 Correlation of Time Difference Considered Experimental Results with STP and Wind Speed for Baltimore by Using k-means

		$K=5$		$K=10$		$K=15$		$K=20$		$K=50$	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
Air pressure	avg.	-0.800	0.083	**-0.738**	0.003	**-0.729**	0.000	**-0.685**	0.000	**-0.574**	0.000
	var	0.000	0.000	0.004	0.000	0.004	0.000	0.003	0.000	0.004	0.000
	Time Difference	15142 s		**19157** s		**24410** s		**15264** s		**16002** s	

Table 3.16 Correlation of Time Difference Considered Experimental Results with STP and Wind Speed for Washington D.C. by Using k-MWO

		$K=5$		$K=10$		$K=15$		$K=20$		$K=50$	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
Air pressure	avg.	-0.657	0.191	**-0.787**	0.001	**-0.735**	0.000	**-0.737**	0.000	**-0.681**	0.000
	var	0.008	0.005	0.001	0.000	0.001	0.000	0.001	0.000	0.003	0.000
	Time Difference	22592 s		**34622** s		**21917** s		**22017** s		**25935** s	

Table 3.17 Correlation of Time Difference Considered Experimental Results with STP and Wind Speed for NYC by Using k-MWO

		$K=5$		$K=10$		$K=15$		$K=20$		$K=50$	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
Air pressure	avg.	-0.613	0.224	**-0.642**	0.012	**-0.605**	0.002	**-0.598**	0.000	**-0.552**	0.000
	var	0.002	0.001	0.002	0.000	0.002	0.000	0.003	0.000	0.007	0.000
	Time Difference	2729 s		**14580** s		**2480** s		**2359** s		**7264** s	

Table 3.18 Correlation of Time Difference Considered Experimental Results with STP and Wind Speed for Baltimore by Using k-MWO

		$K = 5$		$K = 10$		$K = 15$		$K = 20$		$K = 50$	
		τ	p-value	τ	p-value	τ	p-value	τ	p-value	τ	p-value
Air pressure	avg.	-0.659	0.192	**-0.792**	0.001	**-0.714**	0.000	**-0.730**	0.000	**-0.656**	0.000
	var	0.011	0.005	0.001	0.000	0.000	0.000	0.001	0.000	0.003	0.000
	Time Difference	18500 s		**31119 s**		**11883 s**		**18449 s**		**19100 s**	

Table 3.1, τ is -0.7, and in Table 3.10, it is -0.810, which is less than -0.7. At the same time, the corresponding p-value of τ, 0.810, is less than 0.05. Then, in Table 3.10, 0.810 is put in a bold font.

As the time difference defined in Section 3.3.3, it is acceptable only when the p-value is lower than 0.05. The time differences put in a bold face denote the acceptable ones in Tables 3.10-3.18. Note that only air pressure is expressed through consideration of the unreliability and uncertainties of wind speed. In Tables 3.10-3.18, most of time differences are put in the bold face. It represents that most of the time differences are acceptable and most of their corresponding τ values are less than -0.7. In other words, the τ values become low values when the time differences are considered. All satisfied τ values are increased when the time differences are considered. Then, we conclude that the time difference does exist, since it is able to increase the τ values. In other words, when the time difference is concerned, the relation between the virtual world and the real one is much more correlated.

For the three cities, all satisfied time differences are greater than 0. It also represents that the time differences are lead time ones. The only difference among three cities is that their lead time is different. The lead time of Baltimore and Washington D.C. is much greater than that of New York City. As studied in [35], the people located at different communities should have different responses. Let us use Indian Ocean tsunami in 2004 as an example. Due to the tragic memory during that time, South Asia is more sensitive to tsunami than other continents. Therefore,

for each specific city, the records of hurricanes in history are investigated. Because Baltimore is located in Maryland and Washington D.C. is adjacent to Maryland, the records of the region containing both Maryland and Washington D.C. are used. Two links, [3] and [4], from Wikipedia uncover the records of hurricanes in history. Since the year of 1950, the region of Washington D.C. and Maryland have already been attacked by hurricanes 111 times with 12 of them defined as deadly storms. New York State has been impacted 61 times since the year of 1950 and 13 of them were concerned as deadly storms. For both NY and the region of Washington D.C. and Maryland, the numbers of deadly storms are similar. However, in the region of Washington D.C. and Maryland, the number of deaths is 61 during the deadly storms. That number in New York State is 107. The death count in NY is 1.75 times more than that in the region of Maryland and Washington D.C., but the population of New York State is triple more than that in the latter. Also, the number of hurricanes in the region of Maryland and Washington D.C. is 1.82 times more than that in NY. From this perspective, we conclude that the region of Washington D.C. and Maryland is more sensitive to hurricanes than NY. It well explains the reason that both Washington D.C. and Baltimore have longer lead time differences than NY has. The reason is that the residents are more sensitive to and need more time in advance to cope with hurricanes. Thus, during the hurricane, residents in Washington D.C. and Baltimore intend to post more rare-event-related tweets or alerts much earlier than the hurricane's arrival. Meanwhile, there is a short lead time difference in New York City. Our experimental results reveal that the time differences between the virtual world and real one definitely exist. Three clustering algorithms adopted in our work can obtain the same conclusions and results.

3.4.3 Comparisons and Impact of the Number of Clusters

Because of uncertainty and rapid changes of wind speed, only the comparisons with the air pressure are discussed in this section. Tables 3.1-3.9 reflect the correlation that is computed without the concern of time difference by using three clustering algorithm-based methods. Overall, the best τ values among three cities are obtained via k-means++. The three best τ values for Washington D.C., NYC and Baltimore are given when the number of clusters, $K = 10, 20$ and 10, respectively. No matter which method is used, the best τ values are obtained when $K = 10$, 15 or 20 for three cities. Thus, the proper range of K is from 10 to 20. In addition, when selecting K in this range, the τ values change slightly only. However, the cases with $K = 5$ or 50 result in the unsatisfied τ values for each city and each method. This implies that the too few or too many clusters cannot lead to acceptable results for the problem in this work.

Tables 3.10-3.18 give the correlation that is computed with the consideration of time difference. Overall, the best τ values for NYC and Baltimore are obtained via k-means++. The best τ value for Washington D.C. is given by using k-means, but k-means++ based method only gives a slightly greater τ value than the k-means based one. Three best τ values for Washington D.C., NYC and Baltimore are given when $K = 15$, 15 and 10, respectively. It reflects that the proper range of K from 10 to 20 is acceptable with the consideration of time difference. In addition, in this case, no matter which city and which method are concerned, $K = 10$ or 15 for the best τ values. Meanwhile, $K = 5$ gives the unsatisfied τ values and $K = 50$ has the worst satisfied τ values for each city and each method. The k-means++ based data processing method performs well, since its τ values reach the smallest value for most cases. The k-means based method only has the best τ value for Washington D.C. with the consideration of time difference. Furthermore, this best τ value is just slightly less than the τ value obtained via k-means++ based method. The k-MWO

based method cannot reach the best τ value, and thus it is not good enough. In conclusion, the experimental results suggest that the number of clusters should be selected around the number of days during which data are collected. The proposed k-means++ based data processing method performs the best among all.

3.4.4 Discussion of Adopted Clustering Algorithms

The three clustering algorithms adopted in this work have some differences. First of all, k-means clustering algorithm is the basic one. In general, it randomly selects K initial points as centers, and then it stops when the objective function reaches the local or global minimum value. Initially, k-MWO randomly selects centers as k-means does. However, the former utilizes the global optimization ability of mussels wandering optimization and combines with k-means. That is the reason that k-MWO performs slightly better than k-means. The initial points selected by k-means++ differ from the previous two algorithms. It can start from better initial centers. We reveal that k-means++ is superior to the other two methods. In addition, the intensity of posted tweets should be high in the daytime, especially at noon or afternoon, and low at night due to human beings' common habits. Thus, starting from proper initial centers should be more important to the performance of k-means++. Furthermore, we study the posted time of tweets and cluster them in the time domain. It means that the data are clustered at a low dimension. Thus, we obtain a significant result that even though k-MWO combines the ability of global optimization and local search, it does not have superiority over k-means, implying that local search is suitable for our case. As a result, the selection of proper initial centers is more important than others like global optimization. k-means++ performs the best among the clustering algorithms.

CHAPTER 4

A NOVEL FUZZY LOGIC-BASED TEXT CLASSIFICATION APPROACH

In the previous chapter, the keyword search method helps one identify the rare-event-related and unrelated short texts. Its obvious disadvantage is that it is very sensitive to select keywords. If they are not well chosen, it is easy to skip many important rare-event-related instances and may extract some undesired/noisy ones that are not related. This chapter deals with this issue and contains two main aspects, fuzzy logic-based text feature extraction and text classification methods. The former one aims at extracting text features by using a fuzzy logic method. Then, each short-text is represented by a vector. The latter is to classify the short texts into binary classes.

4.1 Data and Feature Extraction

This section focuses on the research of text data. First of all, the data are introduced including data labeling. Next, noisy data, ambiguous words and redundant information, are pre-processed in the data pre-processing step. At last, a fuzzy-logic based feature extraction method is described and used to extract seven features for each short-text.

4.1.1 Dataset Description

Tweets, which are distinct from many other data, usually are short and without any context. Without such context, different people often have different interpretations regarding the meanings of tweets. In this work, we randomly select $2,000$ tweets from the initial dataset, and they are labeled manually as the ground-truth data. Therefore, we build a fuzzy logic-based model by using this labeled dataset. Note that even though there are some auto-labeling methods, such as auto-encoder, they

tend to produce a good number of errors. In order to guarantee the accuracy of labeling, we adopt manual labeling.

These 2,000 tweets are labeled by 15 volunteers. Each volunteer gives a score from 0 to 4. 0 represents that the tweet is extremely not related with our event, Hurricane Sandy. On the contrary, 4 represents that the tweet is extremely related with it. 1, 2 and 3 represent low, moderate and high, respectively. Then, the average of volunteers' scores for each instance represents its final score. We predefine four relevance classes which are regarded as four datasets, D_1, D_2, D_3, and D_4. They correspond to irrelevance, low relevance, moderate relevance and high relevance, respectively. The jth tweet belongs to one of the predefined four datasets: $j \in D_i$ if $i-1 \leq \Phi(j) < i$, where $j \in \{1, 2, ..., 2000\}$ denotes the jth tweet and $i \in \{1, 2, 3\}$; and $j \in D_4$, otherwise. Note that $\Phi(j)$ means the average score of the jth tweet that is given by volunteers.

4.1.2 Data Preprocessing

As a short text message posted online, a tweet has its own formats, structures and properties. In this subsection, data preprocessing should be done first. In fact, tweets are not clean enough for direct and efficient use. They contain Internet slang and "noise" such as a uniform resource locator (URL). Such information may disturb the performance of a classification approach and decrease the computational speed. For instance, a raw tweet, "All systems active! #BucksSandy #Sandy (@ Hurricane Bunker) http://t.co/y1U0FlYp", has such involved interference information. Pattern matching provides a way to solve such problem efficiently. This method checks a given sequence of expressions that match the presence of constituents of some patterns. Usually the matching identifies the correct patterns that are contained in a huge number of given texts [85]. For example, as we know, URL has a fixed format starting with "http://". When "http://" is found by the pattern matching method,

the information following it is automatically removed until a space is encountered. Furthermore, a hashtag usually provides some keywords regarding an event, but it is not fit for all messages since about only one-third of the messages contain hashtags and are often inconsistent [54]. For example, in a tweet, "#sandycantstopme Don't let her stop you. #floodproof #keepgoing", even if there is no space among words in hashtags, we can still understand the meanings of these words. But it is quite difficult and challenging for machines to understand, since, some phrases, e.g., "sandycantstopme" is not a correct English word despite people easily understanding it as "sandy can't stop me". For this situation, we remove all hashtags instead. Finally, stop words are filtered and each word is converted into the lower case. Usually these stop words are the most common words in a language. In fact, they are necessary for some sentences to be grammatically correct or meaningful, such as "the", "this", "a" and "on", but are rarely useful or meaningful. They may appear repeatedly in sentences and carry redundant and often no meaningful information. If we count the most common words in a corpus without removing stop words, "the", as an example, is obviously ranked as the top ones among all. Thus, we filter stop words. In addition, transforming words into the lower case helps us guarantee that all words are of the same format, since a computer may treat the same word, but under lower and upper cases, as two different ones. Hence, for example, "STORM" and "Storm" are regarded as a same word after every capital letter is converted into the lower-case one.

4.1.3 Feature Extraction

Computers cannot cognize the meanings of words and sentences directly. Thus, researchers aim at converting those words and sentences into numerical values for feature extraction. There are several methods such as the bag-of-word and TF-IDF [112]. Most of them focus on the word frequency. If some messages are talking about the same topics, they may contain common or similar semantic words. For instance,

in an airport, "time", "arrival" and "airline" are more repeatedly mentioned than "bids" and "price" that should appear in an auction scenario. Taking a word with frequent appearance into consideration, we pick up the tweets belonging to D_2, D_3 and D_4 from the training dataset and acquire top 50 most frequently used words. Note that D_2, D_3 and D_4 respectively describe low relevance, moderate relevance and high relevance. For word i, its word importance is denoted by α_i and is defined as:

$$\alpha_i = P_i/Q_i \times 100\% \tag{4.1}$$

where P_i is the number of word i that appears in the tweets that belong to D_2, D_3 and D_4; Q_i decides the number of word i that appears in all the tweets; α_i is a percentage that represents the importance of word i. Generally, the larger α_i, the more important word i. Then, we sort all the most frequently used words according to α_i from the highest to the lowest in a key list L. Then we split it equally into three subsets represented by L_1, L_2 and L_3 with different relevant weights θ_1, θ_2 and θ_3, respectively, where $L = L_1 \cup L_2 \cup L_3$. Then, the similarity between a word in a tweet and one in the key list L is calculated by using a similarity function in [5]. A similarity evaluation process is defined as a mathematical operator: $\otimes$. Given a tweet containing n words, T_i, $i \in \{1, 2, ..., n\}$, denotes the ith word in the tweet. W_k, $k \in \{1, 2, ..., 50\}$, denotes the kth word in the key list L. The highest one among the similarity scores is used to represent T_i's score. Thus, the similarity score of T_i is calculated as follows:

$$S_i = \underset{1 \le k \le 50}{max}(\omega_k \times T_i \otimes W_k), i \in \{1, 2, ..., n\} \tag{4.2}$$

where

$$\omega_k = \begin{cases} \theta_1 & if\ k \in \{1, 2, ...,\ 15\} \\ \theta_2 & if\ k \in \{16, 17, ...,\ 31\} \\ \theta_3 & otherwise \end{cases} \tag{4.3}$$

S_i is a basic value. Next, six features are extracted from each tweet individually and are used to build our proposed fuzzy logic-based model. The descriptions and definitions of feature extraction approaches are given next.

1. The highest word score in the jth tweet (H_j)

$$H_j = \max_{1 \leq i \leq n} S_i \tag{4.4}$$

where H_j represents the largest word's score in the jth tweet. It adopts the word with the highest score to represent the tweet's score. If a word has a higher score, it is more possibly related with the event. A tweet with such word is potentially related with the event.

2. The score of the jth tweet (F_j)

$$F_j = \sum_{i=1}^{n} S_i \tag{4.5}$$

where F_j denotes a score of the jth tweet that is an accumulation of all words' scores in a tweet. (4.5) uses the sum of the score of each word to represent the score of a tweet. If there are more words with high scores, this tweet should have a high score. Then, this tweet must be relevant to the event.

3. The number of frequently-used words in the jth tweet (I_j)

I_j, as the third feature, indicates that the number of words in the jth tweet is the same as those in the key list L. Note that L is derived from all training tweets and contains all the most frequently-used words. This feature counts the number of key words in a tweet. It is obvious that the more the key words in it, the more relevant this tweet is to the event.

4. The weight of the jth tweet (G_j)

$$G_j = \frac{F_j}{N_j} \tag{4.6}$$

where N_j is the number of words in the jth tweet and F_j denotes a score of the jth tweet.

5. The weight of frequently-used words in the jth tweet (E_j)

$$E_j = \frac{I_j}{N_j} \tag{4.7}$$

where E_j decides the proportion of the frequently-used words to all the words in a tweet. It takes the number of words, N_j, in a tweet into consideration. A tweet that is very long and has some keywords may not be more relevant to an event than a very short tweet with a few event-related key words. In other words, the larger E_j means that the tweet has more useful information.

6. The number of patterns in the jth tweet (V_j)

Some useful combinations of words may easily be ignored when each of them is concerned separately. For example, "no power", "power off" and "no school" are more informative than a single word like "no" or "power" alone. Then, V_j describes the corresponding pattern count in a tweet. It obviously denotes that the more patterns it has, the more relevant to the event it is.

4.2 Fuzzy Logic-Based Text Classification Method

In this section, a fuzzy logic-based model is proposed as shown in Figure 4.1. We use seven features, including the number of words in a tweet, extracted and defined in Section 4.1.3 as inputs for the proposed model. Fuzzification is a step that maps the crisp or real inputs to fuzzy sets by using membership functions. In this work, we utilize the simple and commonly used trapezoidal-shaped membership function. Inference is a process that is combined with multiple rules and maps a given input to an output. Here we use IF-THEN fuzzy rules to convert the fuzzy input to output. Rules are a set of linguistic statements derived from human expert knowledge and empirical rules. Many defuzzification methods are introduced in [42, 66, 71]. This

work uses five methods, centroid, bisector, mean of maximum (MOM), smallest of maximum (SOM) and largest of maximum (LOM) [42]. R, representing the output, is a single value defuzzified and obtained from an aggregate fuzzy set containing a group of output values.

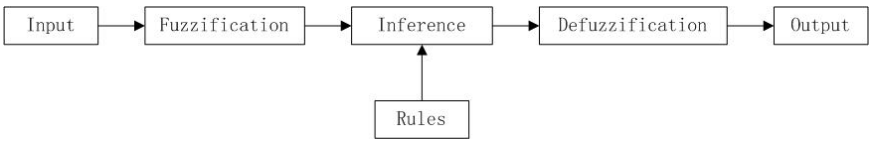

Figure 4.1 The framework of using a fuzzy logic-based model.

4.2.1 Parameters Selection

The parameter ranges of seven inputs and an output are shown in Table 4.1. Note that they are set by using empirical knowledge. For example, the highest word score, H, has five degrees as given in Table 4.1, i.e., very low, low, moderate, high and very high degrees when H falls into [0, 0.35], [0.15, 0.45], [0.25, 0.55], [0.4, 0.7] and [0.6, 0.1], respectively. H's high degree means that its corresponding tweet is highly relevant to the event.

4.2.2 Fuzzy Rules

In this work, we adopt IF-THEN statements to formulate our fuzzy rules. Each IF-THEN statement corresponds to a fuzzy rule and contains a condition or several conditions and a conclusion. The rules are designed and fall into four categories: high relevance, moderate relevance, low relevance, and irrelevance. In order to make a further illustration, some rules as examples are listed as follows:

R_1. *If H is high or very high, and I is high, then R is regarded as high relevance.*

A high word score and many frequently-used words in a tweet infer its high relevance to the event.

Table 4.1 Input and Output Parameters

Variable	Linguistic Variables	Range	Linguistic Value	Parameter
Input	H	0-1	Very Low	0-0.35
			Low	0.15-0.45
			Moderate	0.25-0.55
			High	0.4-0.7
			Very High	0.6-1
	F	0-20	Very Low	0-3.5
			Low	2-8
			Moderate	5-11
			High	8-14
			Very High	11-20
	N	0-20	Short	0-8
			Moderate	5-15
			Long	12-20
	I	0-8	Low	0-3
			Moderate	2-6
			High	4-8
	G	0-1	Very Low	0-0.25
			Low	0-0.38
			Moderate	0.3-0.6
			High	0.55-0.7
			Very High	0.65-1
	E	0-1	Low	0-4
			Moderate	3-7
			High	6-10
	V	0-10	Low	0-4
			Moderate	3-7
			High	6-10
Output	R	0-100	Irrelevance	0-40
			Low Relevance	30-65
			Moderate Relevance	50-85
			High Relevance	75-100

R_2. *If H is moderate and G is moderate, then R is regarded as moderate relevance.*

A tweet is regarded as the moderate relevance to the event when both its word score and weight are moderate.

R_3. *If E is low or moderate, and G is low, then R is regarded as low relevance.*

The low weight of a tweet and the low weight of the frequently-used words indicate that there are few important key words. Thus it is a low relevant tweet.

R_4. *If H is very low and M is long, then R is regarded as irrelevance.*

A tweet is an irrelevant one if there are no words that are closely related to the event in it.

The first category, high relevance, contains those tweets highly related to the event. It needs that the variables have higher value or shorter length, as required in, for example, R_1. The second category, moderate relevance, includes a tweet with moderate linguistic values, such as those satisfying the conditions in R_2. The third category, low relevance, may have some variables that are moderate, but some are low, such as those satisfying the conditions in R_3. These kinds of tweets cannot be regarded as moderately relevant, rather belong to the low relevance category. The last category, irrelevance, has those tweets that do not belong to any other three categories, such as those satisfying the conditions in R_4. Usually, tweets in the last category have either too low linguistic values or are relatively too long. Note that if a tweet is too long with low linguistic values, this tweet contains little or minimal information about the event. Thus, the tweet can be regarded as an irrelevant one.

Accordingly, we establish 25 rules. The four categories, i.e., high relevance, moderate relevance, low relevance and irrelevance, have seven, eight, three and seven rules, respectively.

4.2.3 Defuzzification Methods

We adopt centroid, bisector, MOM, SOM and LOM as our defuzzification methods, which are widely used [66, 71]. The centroid defuzzification method is defined as the center of gravity or center of a defuzzification area. The bisector method utilizes a vertical line that divides the region into two equal sub-regions. Note that the centroid and bisector can sometimes be coincident. It depends on the shape of an aggregate membership function [71]. MOM selects the mean value of the maximum membership function. SOM chooses the smallest value of the maximum membership function [71]. Similarly, LOM corresponds to the largest value of the maximum membership function. MOM, SOM and LOM focus on the maximum value assumed by the aggregate member function. Note that if the aggregate membership function has a unique maximum, then these three defuzzification methods all take the same value. The performance of these methods is discussed in the next section based on experimental results.

4.2.4 Evaluation Metrics

This section describes some evaluation metrics that are suitable to evaluate the effectiveness of our proposed method. A confusion matrix given in Table 4.2 is utilized to determine the performance of a binary classification method [30]. True positive (TP) represents that the number of instances that are positive and classified as positive; false negative (FN) denotes that the number of instances that are positive but classified as negative; false positive (FP) indicates that the number of instances that are negative but classified as positive; true negative (TN) represents that the number of instances that are negative and classified as negative.

Since our work aims at finding the instances that are correctly classified into the rare-event relevant class or irrelevant one separately, a precision value and a

Table 4.2 Confusion Matrix

	Actual Positive	Actual Negative
Predicted Positive	*TP*	*FP*
Predicted Negative	*FN*	*TN*

negative predictive value (NPV) are adopted. In this work, the former one reflects the correctness rate that the relevant tweets are correctly classified into the relevant class. If it is high, more relevant ones are properly classified into the right class, i.e., relevant class. Otherwise, some relevant ones are not properly classified into the right class. Similarly, NPV indicates the correctness rate that the irrelevant tweets that are correctly classified into the irrelevant class. If it is high, more irrelevant ones are correctly classified. Otherwise, some irrelevant ones are not correctly classified. precision and NPV are computed as follows:

$$precision = \frac{TP}{TP + FP} \tag{4.8}$$

$$NPV = \frac{TN}{TN + FN} \tag{4.9}$$

Additionally, in order to compare the effectiveness of two different methods, a change rate λ is introduced to describe that a method can exploit more or less information than another method, which is defined as:

$$\lambda = \frac{x_\alpha - x_\beta}{x_\beta} \tag{4.10}$$

where x_α represents the number of instances that are correctly classified into the right class by using a new method α, and x_β denotes the number of instances that are correctly classified into the right class through an older or baseline method β. In general, more correctly extracted instances are useful to obtain more rare-event-related information. For example, in [16, 20, 63], a keyword search is adopted to extract some instances that contain the keywords, such as hurricane and

Sandy. If some instances do not have any keywords as mentioned above, and talk about trees' falling down or power outage, they are not classified as Sandy-related ones. However, these instances are useful, because they provide some specific information about real phenomena caused by Sandy in this particular context. The change rate is an important metric, since the more event-related instances provide more information that can be further used in the studies for event analysis, such as [16, 19, 24, 39, 56, 59, 88]. If λ is positive, it represents that method α extracts more instances than method β. Otherwise, method α extracts fewer instances than method β.

4.3 Experimental Results

This section shows the experimental results and illustrates the feasibility and performance of the proposed classification method. The results are further compared with the keyword search method adopted in [16, 39, 63]. Five defuzzification methods are adopted and compared. Our case study focuses on Hurricane Sandy, 2012, which is investigated in [16, 39, 63] as well. We aim at determining whether a tweet is related to Hurricane Sandy or not. We convert this problem into a binary classification problem.

4.3.1 Dataset

Hurricane Sandy in the year of 2012 is among the deadliest and most destructive hurricanes. It landed in the northeast of the United States on Oct. 29, hit New York City and New Jersey with severe damages, and affected 24 states, including the entire eastern coast from Florida to Maine. Streets were flooded, power was cut off, and subway lines were suspended. The damages resulted in the loss of nearly 70 billion US dollars [2]. With the help of Twitter's API, we randomly pick up 2,000

tweets with their posted time during the period from Oct. 27, 2012 to Nov. 7, 2012. In this work, tweets related to Hurricane Sandy are called relevant ones for short. Similarly, those are not related to Hurricane Sandy are called irrelevant ones. The ratio between relevant and irrelevant ones is around 1:3 based on our subjective judgment. This operation is used to control and reduce the impact of large imbalance ratio. These 2,000 tweets are randomly divided into two parts, one for training and one for testing, with the consideration of the ratio between relevant and irrelevant tweets. The training data has 1,600 instances and the test data consists of 400. In order to avoid the occasionality of results, we randomly generate five datasets with the index from 1 to 5.

4.3.2 Comparisons of Different Defuzzification Methods

Table 4.3 gives the results obtained by different defuzzification methods for five randomly generated datasets. Table 4.3 gives the averages of their precision values, NPVs, and ROC AUCs by using five different defuzzification methods. Note that ROC AUC is abbreviated from the area under curve (AUC) of a receiver operating characteristic (ROC) curve. In fact, ROC is a graphical plot that reflects the performance of a binary classification approach. The ROC AUC, called as AUC for short, is a regularly used metric to verify the performance of a binary classification method. Our proposed method classifies the instances into two classes, i.e., a relevant one and an irrelevant one. In Table 4.3, the highest precision values and NPVs are put in a bold face. The centroid method is good at classifying the relevant ones, because its precision value is higher than the results obtained by other methods for both training and test data. Furthermore, using the centroid method, AUC is the highest among all defuzzification methods. Accordingly, the bisector method is the second best one, since it leads to the second highest precision value and AUC among

all defuzzification methods. The highest NPV is given by using the LOM method, but it has the worst performance for the relevant class. We thus conclude that the centroid method is fit for classifying the relevant class and the LOM method is good at classifying the irrelevant class. In fact, it is a tradeoff between the relevant class and irrelevant one. If we want to obtain higher precision value, NPV is sacrificed, because some equivocal instances are classified into irrelevant ones.

Table 4.3 Binary Classification Problem by Using the Fuzzy Logic Based Classification Method with Multiple Defuzzification Methods

Method	Evaluation metrics	Training data	Test data
Bisector	precision	0.8932	0.9020
	NPV	0.9478	0.9465
	AUC	0.9735	0.9774
Centroid	precision	**0.9011**	**0.9160**
	NPV	0.9342	0.9292
	AUC	**0.9746**	**0.9782**
Largest of Maximum (LOM)	precision	0.8004	0.7977
	NPV	**0.9898**	**0.9898**
	AUC	0.9622	0.9644
Mean of Maximum (MOM)	precision	0.8705	0.8824
	NPV	0.9421	0.9406
	AUC	0.9672	0.9688
Smallest of Maximum (SOM)	precision	0.8805	0.8962
	NPV	0.9287	0.9288
	AUC	0.9001	0.8998

4.3.3 Comparison with Keyword Search Method

A keyword search method is widely adopted in [25,39,63] to extract Hurricane Sandy relevant tweets from the original dataset. Its advantage is that it obtains highly relevant tweets effectively and accurately. However, because of the limitation of the

keyword list, its disadvantages are clear, i.e., a) it is unable to extract all relevant tweets completely and b) it is highly sensitive to the selection of keywords.

With the consideration of both quantity, precision value and NPV of obtained relevant tweets, comparisons between the proposed fuzzy logic-based method and the keyword search one are conducted. In the experiment, the latter uses the keywords as same as that in [63]. Table 4.4 shows the comparative analysis of quantity, precision value and NPV. The same five datasets adopted in the previous section are employed. In Table 4.4, the first column corresponds to the index of the five datasets. All the evaluation metrics given in the table rely on the test data.

In Table 4.3, because the centroid has a good performance on AUCs and the classification for relevant class, and LOM is good at distinguishing the irrelevant class, we adopt both to compare with the keyword search method. Then, the defuzzification methods adopted are LOM and centroid in Table 4.4. In the table, the keyword search method is good at dealing with relevant tweets because it selects the relevant ones properly. It makes sense since the keyword search method specifies those keywords that are obviously related with Hurricane Sandy.

The proposed fuzzy logic-based with LOM defuzzification method performs well for the classification of irrelevant ones. However, the keyword search method is the worst one for this case. In fact, the keyword search method has a limit keyword list. The keywords that are not included in some tweets are regarded as irrelevant ones. As we aforementioned, some tweets do describe the phenomena during Hurricane Sandy, but they do not have the keywords in the list. This reason leads a few relevant tweets to an incorrect class.

In Table 4.4, for all five datasets, the fuzzy logic-based method with the centroid defuzzification gives the best AUC and the keyword search method is the worst. It concludes that our proposed method performs better on the binary classification problem than the keyword search method.

In addition, the more relevant tweets that are precisely extracted, the more information we can obtain. There are tweets that are associated with the phenomena and reflect real statuses of human life. They cannot be treated as irrelevant and are thus ignored, especially for studies [19] and [25] that analyze the changes and evolution of awareness and moods when an event and its related activities are ongoing. Thus, extracting and mining more relevant tweets is useful and necessary. Note that those relevant tweets mentioned here are not only classified into the relevant class, but also are true relevant ones that are given by labeling. In Table 4.4, no matter which the defuzzification method is adopted, all methods are compared with the keyword search method. In other words, λ is obtained between a fuzzy logic-based method with different defuzzification methods and the keyword search one. In Table 4.4, the fuzzy logic-based method with LOM defuzzification is the best one. It can extract about 30% more relevant tweets than the keyword search method. Except the second dataset, the fuzzy logic-based method with the centroid defuzzification performs better than the keyword search method. Note that for the second dataset, λ is -0.0241, which is negative. It means that the fuzzy logic-based method with the centroid defuzzification extracts 2.41% less relevant tweets than the keyword search method does.

Table 4.5 gives more experimental results about precision change versus λ. Positive and negative changes denote a precision value of our method is increased and decreased in comparison of the keyword search method, respectively. Clearly, a positive change means that our method has higher precision than the keyword search one and vice versa. If $\lambda>0$, it represents that the fuzzy-logic-based method extracts more valuable tweets than the keyword-based one. Otherwise, the former extracts less valuable tweets than the latter. If λ is higher than the precision deterioration, it means that our method extracts more valuable tweets with a small precision deterioration

Table 4.4 Comparison Results of Keyword Search Method and Fuzzy Logic-Based Method with the Centroid or LOM

Data	Method	Defuzzification method	precision	NPV	AUC	λ
1	Keyword search	/	**0.9324**	0.8926	0.8558	/
	Fuzzy logic-based	Centroid	0.8817	0.9283	**0.9666**	0.1081
		LOM	0.7615	**0.9815**	0.9435	**0.3378**
2	Keyword search	/	**0.9639**	0.9243	0.8990	/
	Fuzzy logic-based	Centroid	0.8804	0.9253	**0.9822**	-0.0241
		LOM	0.7803	**0.9963**	0.9649	**0.2410**
3	Keyword search	/	**0.9733**	0.9046	0.8606	/
	Fuzzy logic-based	Centroid	0.9362	0.9477	**0.9829**	0.1733
		LOM	0.7907	**0.9926**	0.9745	**0.3600**
4	Keyword search	/	**0.9863**	0.9021	0.8510	/
	Fuzzy logic-based	Centroid	0.9412	0.9238	**0.9772**	0.0959
		LOM	0.8417	**0.9893**	0.9690	**0.3836**
5	Keyword search	/	**0.9730**	0.9018	0.8558	/
	Fuzzy logic-based	Centroid	0.9405	0.9209	**0.9819**	0.0676
		LOM	0.8145	**0.9891**	0.9698	**0.3649**

and is put in a bold font. In Table 4.5, it is obvious that more valuable tweets are extracted with a small precision deterioration with the our method.

Generally, when comparing all five datasets, our proposed method can extract more relevant tweets than the keyword search method. Both the proposed method and the keyword search are implemented in Python. The average execution time of our proposed method, including training and testing processes, is 45.56 seconds. The keyword search method adopts the keyword list which does not need a training process. Its average execution time is 0.024 seconds. The adopted CUP is Intel Core i7-5500U @2.4GHz with an 8 GB RAM.

In conclusion, the keyword search method gives a better precision value. However, the proposed fuzzy logic-based method obtains greater AUC and a higher NPV than that by the keyword search method. This concludes that the proposed fuzzy logic-based method performs better than a keyword search method when dealing

Table 4.5 Comparison Results of Keyword Search Method and Fuzzy Logic-Based Method with the Centroid or LOM in Percentage

Data	Defuzzification method	precision change	λ
1	Centroid	**-5.07**%	**10.81**%
	LOM	**-17.09**%	**33.78**%
2	Centroid	-8.35%	-2.41%
	LOM	**-18.36**%	**24.10**%
3	Centroid	**-3.71**%	**17.33**%
	LOM	**-18.26**%	**36.00**%
4	Centroid	**-4.51**%	**9.59**%
	LOM	**-14.46**%	**38.36**%
5	Centroid	**-3.25**%	**6.76**%
	LOM	**-15.85**%	**36.49**%

with the binary-classification problem. We claim that our method performs well on a research context, where a high number of relevant tweets are highly desired for the analysis stage, such as [16, 25, 39, 64]. High quantity, precision value and NPV, can guarantee more informative and useful data. However, the keyword list is predefined as a study in [39] and performs well. Those words adopted in the list quite frequently appear for this specific rare event. The keyword search method still has a space for improvement, if more specific words are added. But finding the specific proper words is a challenging issue as they tend to fit a particular case only.

4.3.4 Feature Extraction Comparisons with Word2Vec

As aforementioned, our proposed fuzzy logic-based method contains two steps: extracting features and classification. In this subsection, we compare our fuzzy logic-based feature extraction method with Word2Vec, which has been widely studied in recent years. By using Word2Vec, each word is given a vector with a high dimension. Then, we use the same way in [79] to generate the vector for each tweet. Thus, each tweet is represented by a vector. A classic k-means++ clustering algorithm

is used to classify them into two classes. Google's pre-trained word2vec model contains 100 billion words from a Google news dataset and each word corresponds to 300 features. It provides a vector for each word. The comparison is given in Table 4.6. Our fuzzy logic-based feature extraction method combined with k-means++ is superior to the combination of word2vec and k-means++. It denotes that our feature extraction method is more effective.

Table 4.7 shows the comparison among multiple methods. Because the proposed fuzzy logic-based method with the centroid defuzzification method performs the best in terms of AUC from Table 4.4, we choose the centroid defuzzification as the defuzzification method in Table 4.7 as well. The best AUCs are put in the bold font in Table 4.7. It is clear that our fuzzy logic-based method outperforms others in both training and test data.

Table 4.6 ROC AUC Comparisons between Word2Vec+k-means and Fuzzy-based Feature Extraction Method+k-means

	Word2Vec+k-means++				Fuzzy-based feature extraction method +k-means++			
	Training data		Test data		Training data		Test data	
Data	Accuracy	ROC AUC	Accuracy	ROC AUC	Accuracy	ROC AUC	Accuracy	ROC AUC
1	0.5031	0.6328	0.5025	0.6420	**0.9419**	**0.9008**	**0.9275**	**0.8793**
2	0.5243	0.6134	0.5575	0.6823	**0.9363**	**0.8931**	**0.95**	**0.9101**
3	0.5131	0.6403	0.515	0.6318	**0.9356**	**0.8903**	**0.9475**	**0.9084**
4	0.5125	0.6343	0.505	0.6437	**0.9369**	**0.8935**	**0.9475**	**0.9084**
5	0.5144	0.6395	0.5175	0.6192	**0.9375**	**0.8963**	**0.945**	**0.8973**

Table 4.7 ROC AUC Comparisons among Multiple Methods

	Word2Vec+k-means++		Fuzzy-based feature extraction method + k-means++		Fuzzy logic-based method (Centroid)		Keyword search method	
	Training data	Test data	Training data	Test data	Training data	Test data	Training data	Test data
1	0.6328	0.6420	0.9008	0.8793	**0.9772**	**0.9666**	0.8729	0.8558
2	0.6134	0.6823	0.8931	0.9101	**0.9704**	**0.9822**	0.8620	0.8990
3	0.6403	0.6318	0.8903	0.9084	**0.9779**	**0.9829**	0.8717	0.8606
4	0.6343	0.6437	0.8935	0.9084	**0.9747**	**0.9772**	0.8741	0.8510
5	0.6395	0.6192	0.8963	0.8973	**0.9729**	**0.9819**	0.8729	0.8558

CHAPTER 5

GROUND TRUTH INFERENCE ALGORITHMS BASED ON MANUALLY LABELED SOCIAL MEDIA DATA

In the previous chapter, we assume that the manually given labels from the labelers represent the ground truth. However, they may not be entirely correct, since the labels are created with their subjective judgment. Thus, this is a no-ground-truth problem. In order to conquer this issue, we adopt ground truth inference algorithms to deduce the ground-truth of short-texts in this chapter. Based on the comparative study of four algorithms, the simplicity and high execution speed are the advantages of majority voting (MV). Then, we propose an adaptive majority voting (Adaptive MV) extended from MV. The rest of section is organized as follows. First of all, the no ground truth problem is stated. Then, four algorithms, MV [111], generative models of labels, abilities and difficulties (GLAD) [99], positive label frequency threshold (PLAT) [110], and ground-truth inference using clustering (GTIC) [109], are selected and compared. Next, the evaluation metrics and experimental results show the performance of the algorithms. Lastly, the Adaptive MV algorithm is proposed and is compared with the conventional MV algorithm and the best among GLAD, PLAT, and GTIC on real world datasets.

5.1 Problem Statement

For a crowdsourcing system, a sample set is defined as $E = \{e_i\}_{i=1}^{N}$, where N denotes the number of tasks. Each example is given as $e_i =< \boldsymbol{l}_i, \widehat{y}_i >$, where $\boldsymbol{l}_i$ is the feature vector and $\widehat{y}_i \in \{0, 1\}$ is the estimated label associated with the ith task. A vector $\widehat{Y} = \{\widehat{y}_i\}_{i=1}^{N}$ corresponds to the estimated ground truth for E. The feature vector $\boldsymbol{l}_i$ is defined as $\boldsymbol{l}_i = \{l_{i,j}\}_{j=1}^{R}$, where R represents the number of labelers and $l_{i,j} \in \{0, 1\}$. Note that 0 and 1 indicate that a task belongs to two different classes, i.e., negative

and positive classes. In our case, 0 and 1 identify whether a short-text is unrelated or related to a rare event, respectively. Matrix $L = \{\boldsymbol{l}_i\}_{i=1}^{N}$ with dimension $N \times R$ contains all labels, i.e., $N \times R$ labels, which are given by R labelers. In order to compare the performance of algorithms, the ground truth or true label is represented as y_i for each task. The ground truth vector is defined as $Y = \{y_i\}_{i=1}^{N}$ for N tasks. Then, our objective is to obtain an estimated $\widehat{y_i}$ for each task i as its estimated ground truth, and to minimize the empirical risk as follows:

$$\Gamma = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\widehat{y_i} \neq y_i) \tag{5.1}$$

where $\mathbb{I}$ is an indicator function whose output is 1 if the test condition is true, or satisfied. Otherwise, its output is 0. In other words, for each task, if $\widehat{y_i} \neq y_i$, $\mathbb{I} = 1$; otherwise, $\mathbb{I} = 0$.

5.2 Ground Truth Inference Algorithms

In this section, four adopted ground truth inference algorithms, MV, GLAD, PLAT, and GTIC, are described in detail. The reasons for selecting them are discussed as follows.

1) MV is a basic algorithm, and its simplicity and effectiveness are its clear advantages. It has been adopted in many studies [26, 57, 99] as a baseline method.

2) GLAD was proposed in 2009 and is a classical EM-based algorithm. With both the reliability of the labeler and the difficulty of the example considered, it is superior to the other EM-based algorithms.

3) Both PLAT and GTIC are two novel algorithms and are verified to perform better than MV and GLAD on noisy labels.

4) Both PLAT and GTIC take the biased labeling issue into consideration. In addition, our datasets are imbalanced, i.e., there are more event-unrelated messages than related ones.

5.2.1 Majority Voting (MV)

For binary classification problems, a common majority voting strategy is as follows:

$$\widehat{y_i} = \begin{cases} 1 & \text{if } \frac{1}{R}\sum_{j=1}^{R} l_{i,j} \geq 0.5, \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

For each instance, MV counts the proportion of positive and negative labels, i.e., 1 and 0. In other words, if the number of positive labels is greater than that of negative ones, MV returns a positive label; otherwise, a negative one. MV is simple, and it is effective in many cases. The studies [110] and [84] define a labeling quality for a labeler and the labeling quality for the j-th labeler is given as follows:

$$p_j = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(l_{i,j} = y_i) \tag{5.3}$$

It represents the percentage that the labels given by the j-th labeler, $\boldsymbol{l}_j = \{l_{i,j}\}_{i=1}^{N}$, match the ground truth, Y. If it is high, the corresponding labeler is a good labeler and thus provides good quality labels for tweets. Otherwise, this labeler does not label well. An integrated labeling quality indicates the percentage of integrated labels that match the ground truth. Every labeler is assumed to have the same labeling quality. The integrated labeling quality can be computed by using the Bernoulli model as follows:

$$q = \sum_{i=N+1}^{2N+1} \binom{2N+1}{i} p^i (1-p)^{2N+1-i} \tag{5.4}$$

where q represents the integrated labeling quality and p is the labeling quality of a labeler. If p is not less than 0.5, then q approaches to 1 as the number of labelers increases. Figure 5.1 shows the integrated labeling quality versus the number of labelers when different p's are given.

However, the disadvantage of the Bernoulli model is that not all labelers are equally good so their labeling qualities are different. In some particular cases, if there

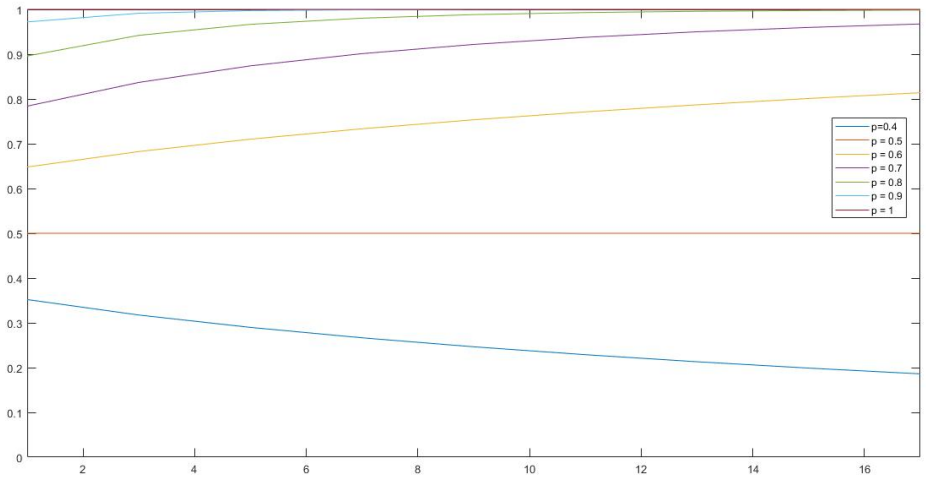

Figure 5.1 Bernoulli model when the labeling quality varies.

is only one qualified labeler and many novices, then the integrated labels are partial to the novices and may gain unexpected/undesired results. In general, the integrated labeling quality when all labeling qualities of labelers are not the same is given as follows:

$$q = \sum_{k=1}^{2^{R-1}} \prod_{j=1}^{R} {p_j}^{\sigma_j} (1 - p_j)^{1-\sigma_j} \tag{5.5}$$

where p_j is the labeling quality of the j-th labeler. σ_j is an indicator. If the j-th labeler gives a correct label, it is 1. Otherwise, it is 0. In the cases with R labelers and different labeling qualities, the probability of giving a correct label is $\prod_{j=1}^{R} {p_j}^{\sigma_j} (1 - p_j)^{1-\sigma_j}$. In addition, there are two results that a labeler for a task, either correctly giving a label, $\sigma_j = 1$ or incorrectly giving a label, $\sigma_j = 0$. Thus, there are 2^R possible cases given by R labelers. If MV is followed, we need to have no less than a half number of labelers that provide correct labels. Thus, there are $2^{(R-1)}$ cases that satisfy the requirement, $\sum_{j=1}^{R} \sigma_j > \frac{R}{2}$. Figures 5.2 and 5.3 show the changes of integrated labeling quality versus the number of noisy labelers. Note that noisy labelers represent those that have low labeling qualities. Examples are given in Figures 5.2 and 5.3, which these figures show the scenarios with different labeling qualities.

In order to simplify the scenarios, we assume that there are two kinds of labeling qualities for labelers, p_h and p_l. They denote the high and low labeling qualities, respectively. When there are more noisy labelers, the integrated labeling quality is reduced due to the incorrect labels given by the noisy labelers. In Figure 5.2, we see that the smaller p_l, the more sharp the curve. In Figure 5.3, we see that as p_h decreases, the curve quickly declines. Many studies, such as [50], investigate the strategy that allocates different weights to different labels.

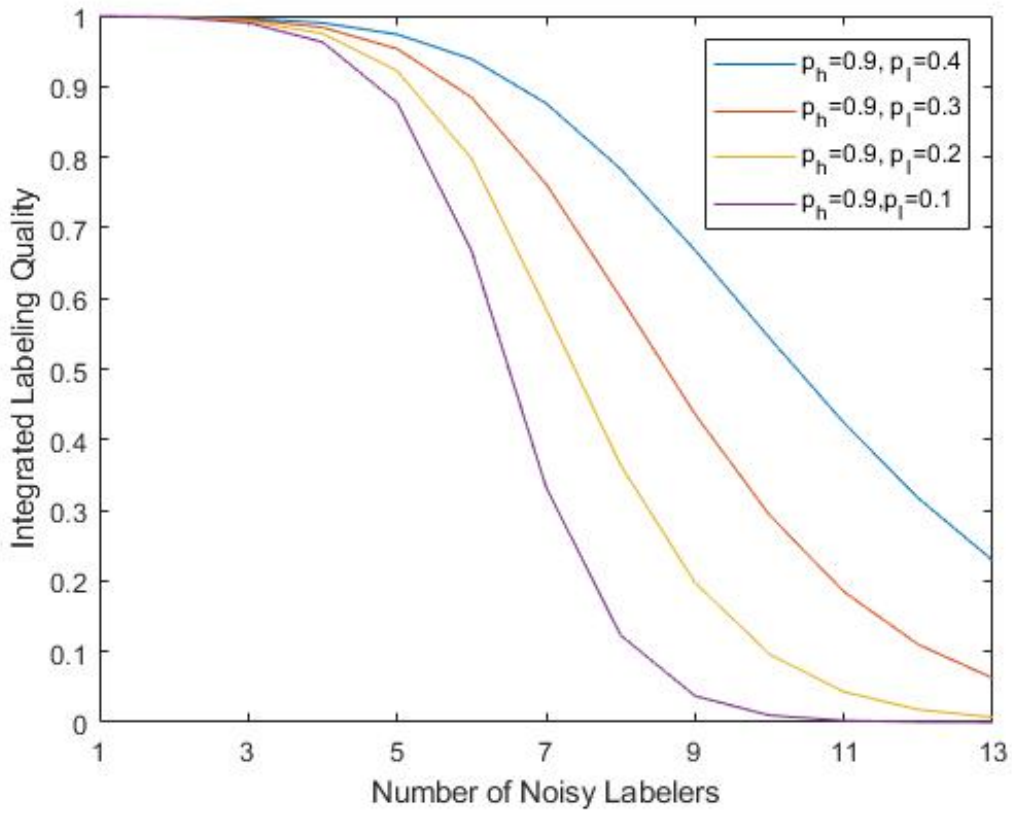

Figure 5.2 Integrated labeling quality versus the number of noisy labelers when $p_h = 0.9$ and $p_l \in \{0.4, 0.3, 0.2, 0.1\}$.

5.2.2 Generative Models of Labels, Abilities and Difficulties (GLAD)

Whitehill *et al.* [99] formulate a probabilistic model of the labeling process with two parameters, the difficulty of task and the expertise of labeler, thus leading to GLAD.

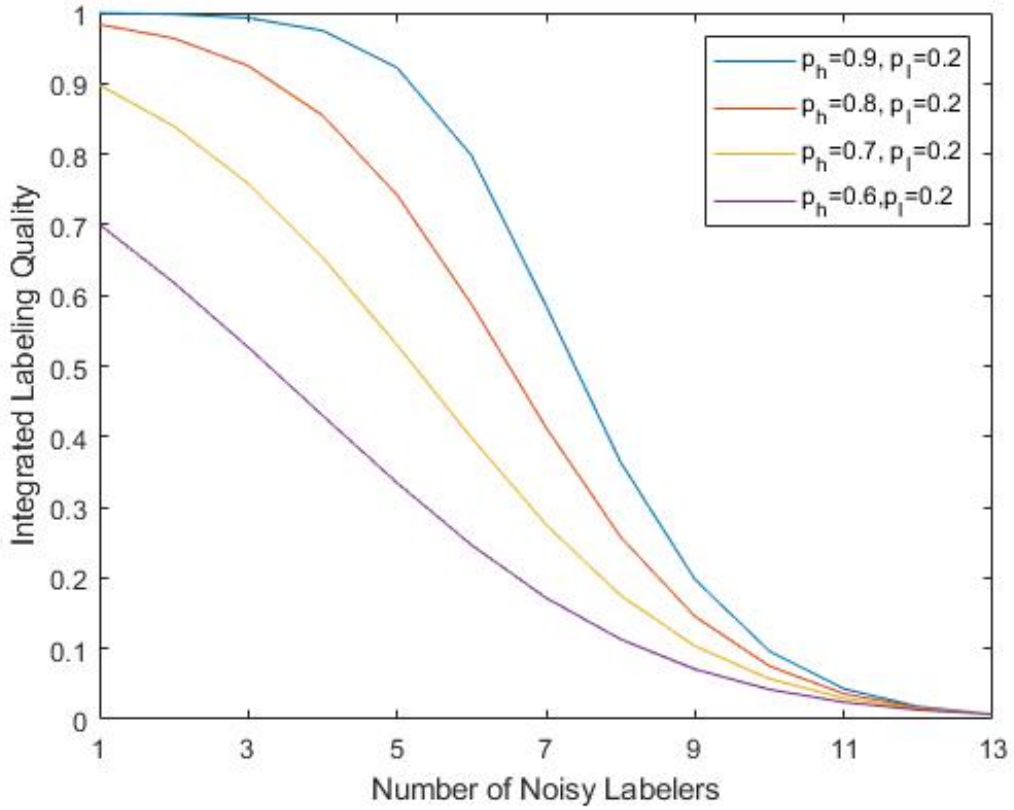

Figure 5.3 Integrated labeling quality versus the number of noisy labelers when $p_h \in \{0.9, 0.8, 0.7, 0.6\}$ varies and $p_l = 0.2$.

The former represents the difficulty level while labeling a task. The difficulty here does not mean that a task is difficult. Instead, because some texts are ambiguous and labelers may be confused, the difficulty of a task is extended to whether identifying a task is hard. The difficulty of task i is defined as $1/\beta_i \in (0, +\infty)$. $1/\beta_i \to +\infty$ means that the task is very ambiguous and even the most proficient labeler has a 50% chance of labeling it incorrectly. On the contrary, $1/\beta_i \to 0$ represents that the task is very easy to label, i.e., an obtuse labeler can label it 100% correctly. The expertise of labeler j is modeled by the parameter $\alpha_j \in (-\infty, +\infty)$. If α_j approaches $+\infty$, it means that the labeler always labels tasks correctly. If α_j decreases to $-\infty$, the labeler always makes incorrect decisions. Finally, $\alpha_j = 0$ means that the labeler cannot determine two classes.

The label given by labeler j to task i is denoted as l_{ij} and is generated by a sigmoid function as follows:

$$p(l_{ij} = \widehat{y}_i | \alpha_j, \beta_i) = \frac{1}{1 + e^{-\alpha_j \beta_i}} \tag{5.6}$$

In this model, the observed labels are sampled from random variables $\{l_{ij}\}$. The unobserved variables are the estimated labels $\widehat{y}_i$, the labeler's ability α_j, and the difficulty of task $1/\beta_i$. The goal is to search for the most probable values of unobserved variable $\widehat{Y}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ via the given observed data. Note that $\widehat{Y} = \{\widehat{y}_i\}_{i=1}^{N}$, $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1}^{N}$, and $\boldsymbol{\beta} = \{\beta_j\}_{j=1}^{R}$. Then, an Expectation-Maximization approach (EM) is adopted to obtain the maximum likelihood and to estimate the parameters. In the E-step, the posterior probability of all $\widehat{y}_i \in \{0, 1\}$ given $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is as follows:

$$\begin{aligned} p(\widehat{y}_i | \boldsymbol{l}, \boldsymbol{\alpha}, \beta_i) &= p(\widehat{y}_i | \boldsymbol{l}_i, \boldsymbol{\alpha}, \beta_i) \\ &\propto p(\widehat{y}_i) \prod_j p(l_{ij} | \widehat{y}_i, \alpha_j, \beta_i) \end{aligned} \tag{5.7}$$

By using the conditional independence assumptions, i.e. $p(\widehat{y}_i | \boldsymbol{\alpha}, \beta_i) = p(\widehat{y}_i)$, we can simplify $p(\widehat{y}_i | \boldsymbol{\alpha}, \beta_i)$ as $p(\widehat{y}_i)$ in (5.7). In the M-step, the auxiliary function Q is defined as the expectation of joint log-likelihood of the observed and hidden variables $\boldsymbol{l}$ and $\widehat{Y}$, respectively, given $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The function Q is given as follows:

$$\begin{aligned} Q(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= E[\ln p(\boldsymbol{l}, \widehat{Y} | \boldsymbol{\alpha}, \boldsymbol{\beta})] \\ &= E\Big[\ln \prod_i \Big(p(\widehat{y}_i) \prod_j p(l_{ij} | \widehat{y}_i, \alpha_j, \beta_i) \Big) \Big] \\ &= \sum_i E[\ln p(\widehat{y}_i)] + \sum_{ij} E[\ln p(l_{ij} | \widehat{y}_i, \alpha_j, \beta_i)] \end{aligned} \tag{5.8}$$

In the M-step, Q is maximized by tuning $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ values. By using (5.6), (5.8), and the gradient ascent, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ can be found to locally maximize Q. Then, the E-step and M-step are repeated until Q is stabilized.

5.2.3 Ground Truth Inference using Clustering (GTIC)

A ground truth inference that uses a clustering algorithm is called GTIC for short. This algorithm pays attention to a multiple classification problem, and thus is able to work on a binary one as well. It contains two steps: feature generation and clustering. An instance, e_i, is associated with a multiple noisy label set, $\boldsymbol{l}_i$, and consists of labels belonging to classes from c_1 to c_k. N_k is the number of labelers that give c_k as its label for instance i, i.e., $N_k = \sum_{j=1}^{R} \mathbb{I}(l_{ij} = c_k)$, where R represents the number of labelers. Then, the probability of this instance being a member of class k is given by a parameter θ_k. Then, we have

$$\theta = [\theta_1, \theta_2, ..., \theta_k, ..., \theta_K], \text{where } 0 \leq \theta_k \leq 1 \text{ and} \sum_{k=1}^{K} \theta_k = 1 \tag{5.9}$$

where K denotes the number of classes. The Bayesian statistics model is adopted to estimate the parameter. For each label $l_{ij} \in \{c_1, c_2, ..., c_K\}$, there exists a K-dimension random vector $x_k^{(j)} \in \{0, 1\}^K$, where $x_k^{(j)} = 1$ indicates that $l_{ij} = c_k$. The probability mass function of a random vector $x^{(j)}$ obeys a multinomial distribution. The likelihood of all labels provided for instance i is as follows:

$$p(l_i|\theta) = \prod_{j=1}^{J} \mu(x^{(j)}|\theta) = \prod_{k=1}^{K} \theta_k^{N_k} \tag{5.10}$$

where $\mu(x^{(j)}|\theta)$ is a multinomial distribution given θ. The conjugate prior of a multinomial distribution is a Dirichlet distribution that results in the posterior in the same form. Thus, the posterior of parameter θ is as follows:

$$\begin{aligned} p(\theta|\boldsymbol{l}_i) &\propto p(\boldsymbol{l}_i|\theta)p(\theta) \propto \prod_{k=1}^{K} {\theta_k}^{N_k} \theta_k^{\alpha_k - 1} \\ &= Dir(\theta|\alpha_1 + N_1, ..., \alpha_k + N_k) \end{aligned} \tag{5.11}$$

where $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_k, ..., \alpha_K\}$, and $\alpha_k \geq 0$ is a hyper parameter of a Dirichlet distribution. By using a Lagrange multiplier with the constraint in (5.9), the

maximum value of $p(\theta|\boldsymbol{l}_i)$ can be calculated. Note that $p(\theta|\boldsymbol{l}_i)$ is the posterior distribution. Then, a constrained objective function is given by taking the logarithm on (5.10), log prior, and constraint-related item, as follows:

$$\ell(\theta, \lambda) = \sum_{k=1}^{K} N_k \log\theta_k + \sum_{k=1}^{K} (\alpha_k - 1)\log\theta_k + \lambda(1 - \sum_{k=1}^{K} \theta_k) \tag{5.12}$$

By taking derivatives with respect to λ and θ_k, respectively, and the sum-to-one constraint in (5.9), the maximum a posteriori (MAP) estimation of θ_k is given as:

$$\widehat{\theta_k} = \frac{N_k + \alpha_k - 1}{N + \sum_{k=1}^{K} \alpha_k - K} \tag{5.13}$$

Usually, a crowdsourcing system is deployed in an agnostic environment. In other words, there is no priori knowledge. Then, we use a non-informative uniform priori, i.e., $\alpha_1 = ... = \alpha_K = 1$. At this point, θ_k simply becomes the frequency of label c_k. Thus, for each instance, θ_k is treated as its kth feature, and then a clustering algorithm is adopted to cluster similar instances. In addition, the $(K+1)$-th feature is obtained by calculating the average variety of every "phase" against its previous "phase" in a histogram, i.e.,

$$\theta_{K+1} = \frac{1}{K} \sum_{k=1}^{K-1} (\theta_{k+1} - \theta_k) \tag{5.14}$$

The main steps of GTIC are as follows:

1. For each e_i in E, use (5.13) and (5.14) to generate its $K+1$ features, i.e., $\theta_i = (\widehat{\theta_{i1}}, ..., \widehat{\theta_{iK}}, \theta_{i(K+1)})$.

2. Use the k-means clustering algorithm by computing Euclidean distance.

3. For each cluster s with the size of $M^{(s)}$ obtained from k-means, create vector $\tau^{(s)} = \sum_{i=1}^{M^{(s)}} \theta_k^{(i)}$, and $s \in \{1, 2, ..., K\}$.

4. For each cluster, based on its vector $\tau^{(s)}$, assign this cluster with the class $k^{(s)} = \text{argmax}_k\{\tau_k^{(s)}\}$ under the constraint that a cluster is mapped to one and only one class.

5. Assign each e_i, an inferred label according to the label of each cluster and return E. $E = \{e_i\}_{i=1}^{N}$ and N denotes the number of tasks.

Step 1 generates $K + 1$ features, and Step 2 runs the k-means clustering algorithm and returns K clusters. Steps 4-6 are executed from the cluster with the maximum and the minimum size to map one cluster into one and only one class.

5.2.4 Positive Label Frequency Threshold (PLAT)

PLAT aims at dealing with the problem that has two constraints: binary classification and imbalanced labeling. A class with fewer instances than the other is called the positive class and the other is the negative one. In this algorithm, a dataset is defined as $L = \{\boldsymbol{l}_i\}_{i=1}^{N}$ with an $N \times R$ dimension containing all labels, i.e., $N \times R$ labels. N and R represent the number of instances and number of labelers, respectively. $\boldsymbol{l}_i = \{l_{i,j}\}_{j=1}^{R}$ denotes a feature vector, where $l_{i,j} \in \{0, 1\}$. Note that 0 and 1 indicate that an instance belongs to two different classes, i.e., negative and positive classes. In our case, 0 and 1 identify whether a short text is unrelated or related to a rare event, respectively. The positive frequency of instance i is denoted as $f_i^+ = r_i/R_i$, where r_i is the number of positive labels and R_i is the total number of labels. Because all R labelers give labels to every instance, $R_i = R$ represents the number of labelers of instance i. $f_k^+ \in \{0, 1/R, 2/R, ..., (R-1)/R, 1\}$ denotes the positive frequency given by R labelers, where $k \in \{0, 1, 2, ..., R\}$ is an index. 0 and 1 correspond to the two cases that no labeler gives a positive label and all labelers give positive labels. F_k^+ is a set associated with index k and corresponds to f_k^+. For the instance i, if $f_i^+ = f_k^+$ where $i \in \{1, 2, ..., N\}$, then f_i^+ is assigned into set k, i.e., $f_i^+ \in F_k^+$. By counting the number of instances in F_k^+, a frequency table is obtained. Note that this table has two coordinates, frequencies, f_k^+, and the number of instances in F_k^+. In addition, it records f_k^+ in the ascending order versus the number of corresponding instances.

Zhang *et al.* [110] draw a positive frequency distribution (PFD) graph that is associated with the frequency table. The positive frequency and the number of corresponding instances are its horizontal and vertical coordinates, respectively. By assumption, for $\boldsymbol{l}_i$ associated with the i-th instance, the probability p_k having k positive labels obeys a binomial distribution. In addition, they prove that if the imbalance ratio is not too big, there are two peaks and one valley in a PFD graph. If so, then there are not two distinct peaks, but only one peak with the maximum number of instances instead. In other words, there are two kinds of PFD graphs with two cases. Then, the binary classification problem is converted into estimating the best threshold. In the first case, the graph has two peaks and one valley. In the second case, it has only one peak. For the first case, the positive frequency that is associated with the valley is the threshold. For the second case, the positive frequency that corresponds to the peak is the threshold. Because the frequency is sorted in the ascending order, the frequencies of instances that are less than the threshold are classified into the negative class. Otherwise, they belong to the positive class. This algorithm fits the imbalanced labeling. There is a constraint stating that more than half of the instances should have frequencies that are not greater than the threshold. If the constraint is not satisfied, the threshold should be increased to satisfy it. In [110], it assumes that if the labeling qualities of labelers are equal, and thus the probability having k positive labels obeys the binomial distribution. However, in real world, so labelers are not the same, then the labeling qualities are not same; the probability having k positive labels cannot obey the binomial distribution any more. Thus, the cases, two peaks, and one valley and one peak, cannot be strictly followed.

Figures 5.4 and 5.5 are shown as two examples of PFD regarding Hurricane Maria and Sandy, respectively. Their horizontal and vertical coordinates denote the values of positive frequency and the number of instances corresponding to them. Both of them have one valley and two peaks and are relatively flat in the middle.

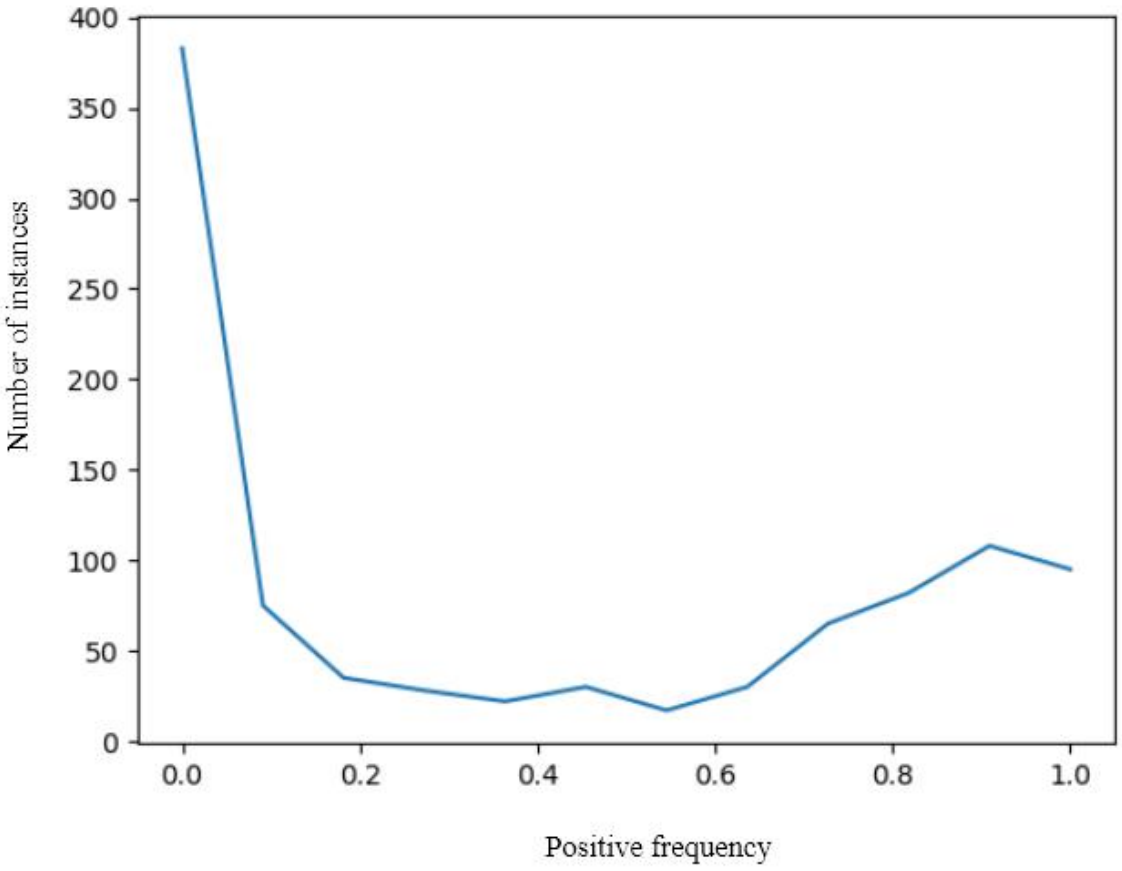

Figure 5.4 PFD for Hurricane Maria when the number of labelers is 11.

5.2.5 Dataset

Two datasets are crawled and collected via Twitter's API during the two destructive disasters, i.e., Hurricane Sandy 2012 and Hurricane Maria 2017. During the hurricanes, there were numerous tweets posted, most of which are not related to them, but most of them are not. In order to keep a low imbalanced ratio, we carefully choose 887 and 970 tweets for Hurricane Sandy and Hurricane Maria, respectively. If a tweet is related to the hurricane, it is called a rare-event-related tweet. Otherwise, it is an unrelated one. 13 labelers are requested to label all of the tweets individually and independently. Each labeler gives either 0 (unrelated) or 1 (related) to a tweet. In reality for most cases, nobody knows the exact meaning of a tweet other than the user himself/herself. Then, identifying whether a tweet is related to the hurricane

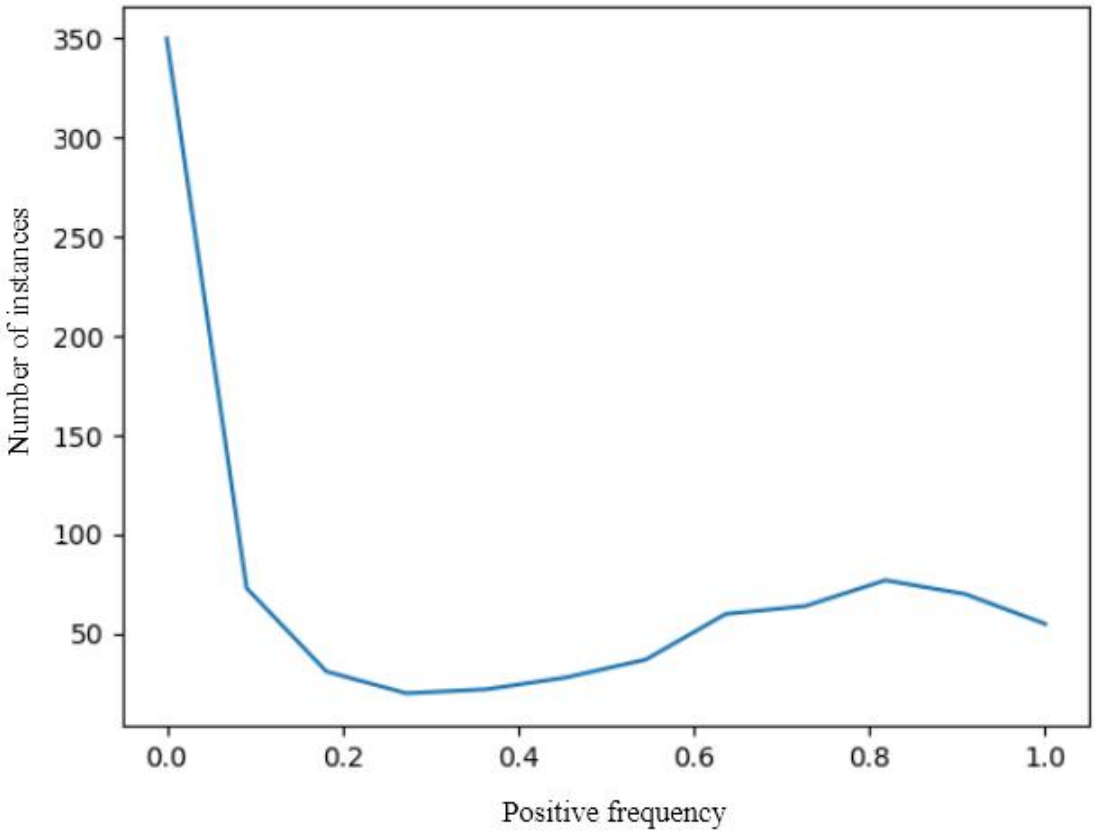

Figure 5.5 PFD for Hurricane Sandy when the number of labelers is 11.

is difficult. It is clear that we are studying a no-ground-truth problem. In order to compare the performance among four algorithms in our dataset, we select some short texts from newspapers, street interviews, and tweets with strong correlated hashtags. In other words, they have ground truth.

5.2.6 Evaluation Metrics

In the experiment, we adopt five evaluation metrics that are able to show the performance of algorithms. The corresponding p-value of hypothesis testing is further described along with them.

Accuracy The accuracy is a basic metric that calculates the percentage of correctly classified instances.

McNemar test The McNemar test is a statistical test used on paired data. It is adopted in order to test whether any two algorithms can reach the same accuracy. The McNemar test is applied to a 2×2 contingency table given in Table 5.1 and is defined in (5.15):

Table 5.1 Contingency Table

Algorithm B	Algorithm A	
	Positive	Negative
Positive	e_{00}	e_{01}
Negative	e_{10}	e_{11}

$$\chi^2 = \frac{(e_{01} - e_{10})^2}{e_{01} + e_{10}} \tag{5.15}$$

If either e_{01} or e_{10} is small, then χ^2 cannot approximate the chi-square distribution well. An exact binomial test can then be used. Edwards proposes the following continuity that corrects the version of the McNemar test and is given as follows [29]:

$$\chi^2 = \frac{(|e_{01} - e_{10}| - 1)^2}{e_{01} + e_{10}} \tag{5.16}$$

ROC AUC ROC AUC is short for the receiver operating characteristic curves and area under the curve. ROC is a two-dimensional curve that models the trade-off between the true positive rate (TPR) and false positive rate (FPR) [30] [62]. AUC corresponds to the area that is under the curve of ROC. This metric is popularly used to verify the performance of methods when dealing with imbalanced data.

F-measure F-measure is another metric that evaluates the classification results for imbalanced datasets [62]. It is computed by using precision and recall, and given in 5.17.

$$F = \frac{2 \bullet precision \bullet recall}{precision + recall} \tag{5.17}$$

Note that the precision compares correctly classified positive instances to all instances that are classified as positive. The recall compares the correctly classified positive instances to the instances that their true labels are positive. The larger F-measure, the better performance.

Average Execution time The average execution time records the average speed of an algorithm that is executed multiple times. It reflects how fast an algorithm can be executed.

***p*-value** In statistical hypothesis testing, the result has statistical significance if it is impossible to reach given the null hypothesis. Then, a study defines the significance level, ρ, which is the probability for the study to reject the null hypothesis. The p-value of a result is the probability of obtaining a result that occurs. By the standards of the study, when p-value is less than ρ, the result is statistically significant or the null hypothesis is rejected. ρ is pre-given and denotes the probability of the occurrence of a small probability event. Commonly, ρ can be 0.01, 0.05 and 0.1 [31]. In our study, we define the ρ value as $\rho_0 = 0.05$.

5.2.7 Experimental Results

In this section, we work with two datasets and investigate the effectiveness of algorithms if the labelers are randomly selected. Note that we have 13 labelers to label the datasets. In order to verify whether an algorithm is impacted by the quality of the labelers and the number of labelers, we randomly select the labelers. For each case, the number of randomly selected labelers is denoted as x. The results below explore the changes of labelers among multiple algorithms.

Accuracy In this section, we compare the accuracy values among four algorithms. Figures 5.6 and 5.7 show the changes of accuracy values versus the changes of labeler counts among four algorithms.

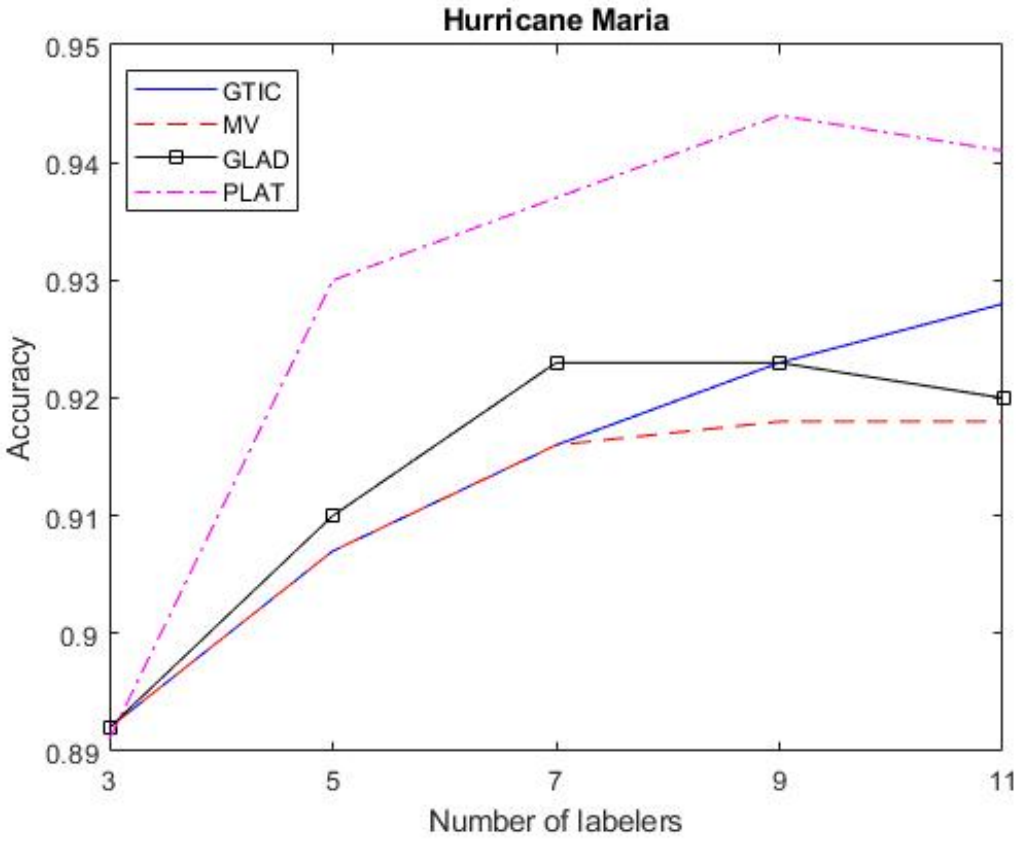

Figure 5.6 Accuracy comparisons among four algorithms for Hurricane Maria.

In both Figures 5.6 and 5.7, the horizontal axis represents the number of labelers. Since there are 13 labelers and the labelers are randomly selected, we choose 3, 5, 7, 9 and 11 as the horizontal axis. Note that number of labelers, x, is odd, in order to avoid a tie case. Each algorithm is executed 30 times and the vertical axis corresponds to the average of accuracy values for each algorithm. There are two datasets. Each of them contains 5 cases that are associated with labeler counts, i.e., 3, 5, 7, 9 and 11. In general, there are 10 cases per labeler count. PLAT performs the best among all

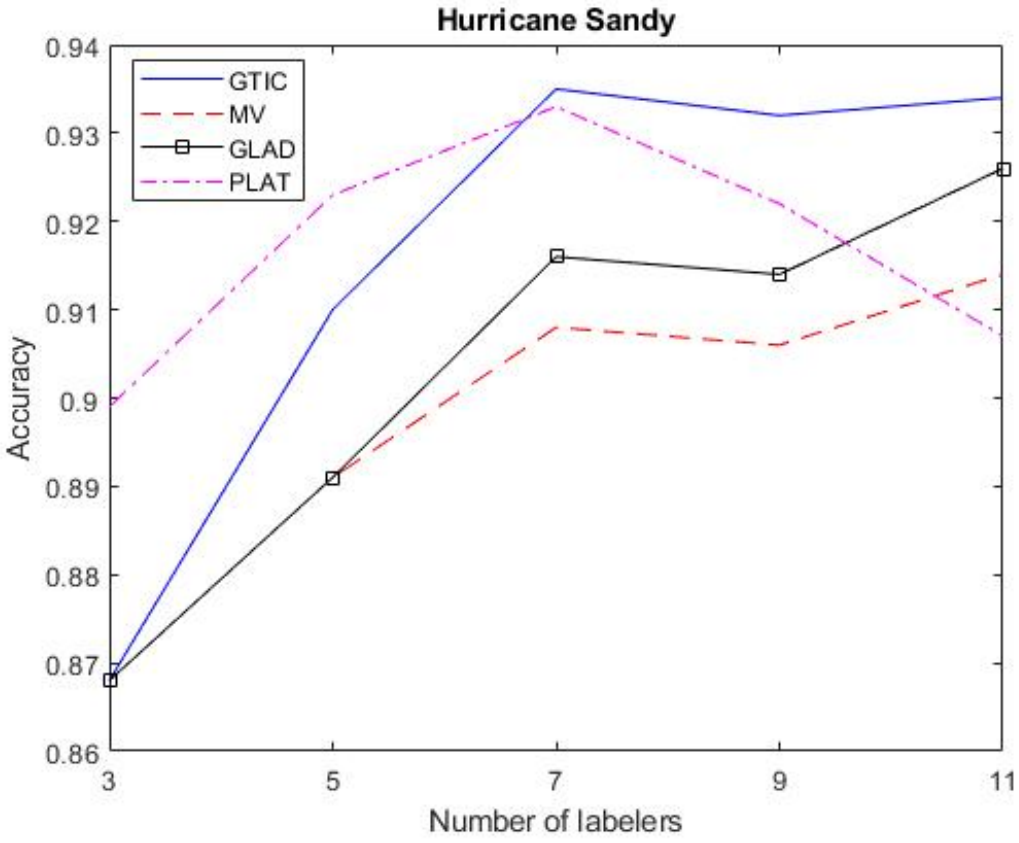

Figure 5.7 Accuracy comparisons among four algorithms for Hurricane Sandy.

algorithms because it is the best one for 7 cases and it is the second best for 2 cases. GTIC is the second best since it achieves the best for 4 cases and the second best for 3 cases. On the contrary, MV is the worst because it is the worst for almost all cases. In addition, the accuracy values of MV, GLAD and GTIC increase with the number of labelers. This trend is obvious because their basic strategy follows the majority labels. However, PLAT is different from them. It needs to analyze the PFD and is sensitive to labeler quality and noisy labels. Even though PLAT performs better than MV and GLAD on noisy labels [110], it is still possible to obtain extremely bad cases when the labelers are not selected well. This explains the reason that PLAT sometimes does not work well. In addition, Figures 5.8 and 5.9 show the box plot of accuracy values when the number of labelers are 9 and 11, respectively. We use them

as an example to analyze the performance of PLAT in-depth. Note that the number of labelers are 9 and 11, which correspond to the cases that PLAT does not perform the best. The bar inside the box represents the median of accuracy values for each algorithm. The up arrow represents the mean of accuracy values. In both Figures 5.8 and 5.9, the median values obtained from PLAT are the best. It means that half of the accuracy values obtained from PLAT are above the median value. In other words, in most cases, PLAT obtains a good performance. However, since there are some accuracy values that are very low, it hurts PLAT's average accuracy values.

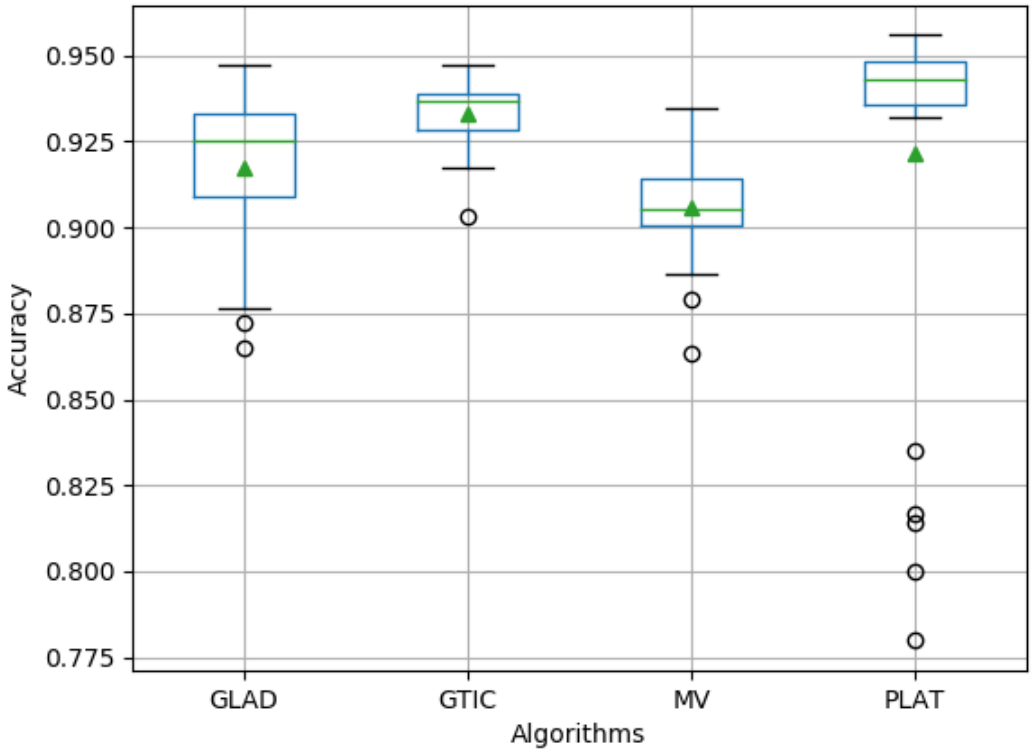

Figure 5.8 Box plot of accuracy values among four algorithms for Hurricane Sandy when $x = 9$.

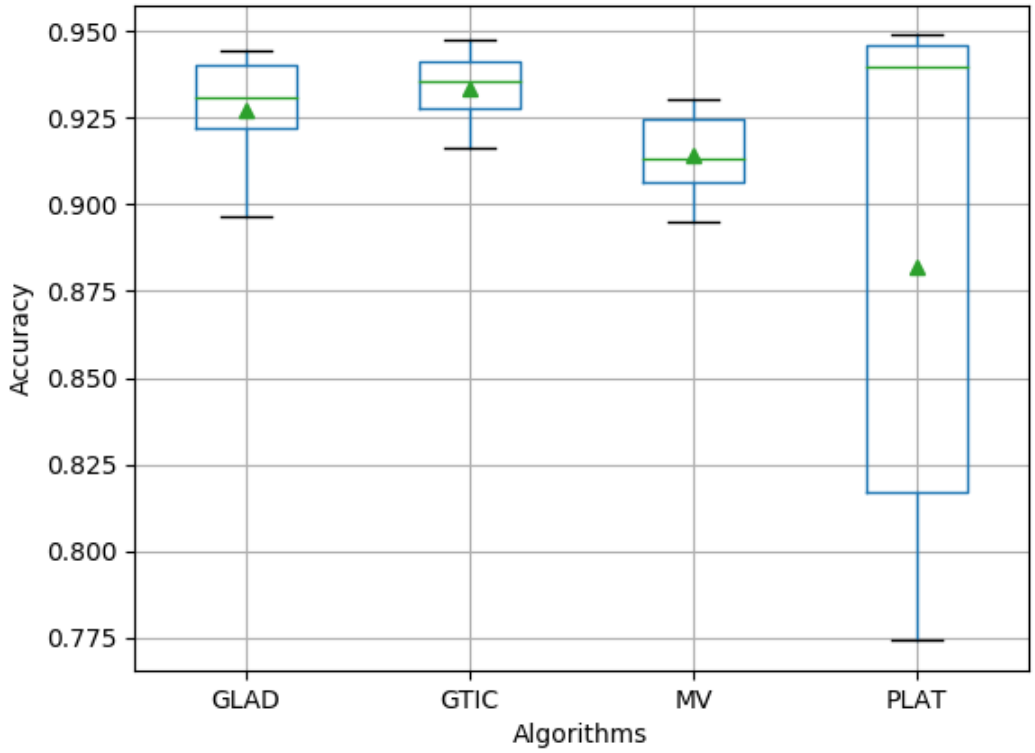

Figure 5.9 Box plot of accuracy values among four algorithms for Hurricane Sandy when $x = 11$.

McNemar test In order to test whether the accuracy values obtained from different algorithms have significant differences or not, we choose to use the McNemar test. Each algorithm is executed 30 times, which returns 30 accuracy values accordingly. Then, the McNear test is able to verify whether the accuracy obtained from any pair has a significant difference. If the p-value obtained by the McNemar test is not greater than the confident coefficient, then the algorithms have a significant difference in accuracy. Note that taking 30 samples is the minimum requirement to run the McNemar test to create its common knowledge, and thus, each algorithm is executed 30 times. Tables 5.2 and 5.3 show comparisons of p-values between two algorithms. Note that the p-values in the tables are average values.

Table 5.2 p-value Comparisons between Two Algorithms for Hurricane Maria Data with McNemar Test

	3	5	7	9	11
GTIC vs MV	1.0000	1.0000	1.0000	0.7443	0.3921
GTIC vs GLAD	1.0000	0.8377	0.3519	0.4295	0.3825
GTIC vs PLAT	1.0000	0.2242	0.0827	0.1037	0.1519
MV vs GLAD	1.0000	0.8377	0.3519	0.4183	0.5825
MV vs PLAT	1.0000	0.2242	0.0827	0.0194	0.0956
GLAD vs PLAT	1.0000	0.2283	0.1927	0.0791	0.0915

Table 5.3 p-value Comparisons between Two Algorithms for Hurricane Sandy Data with McNemar Test

	3	5	7	9	11
GTIC vs MV	1.0000	0.7096	0.2405	0.0622	0.0092
GTIC vs GLAD	1.0000	0.5532	0.1584	0.3197	0.5069
GTIC vs PLAT	0.8000	0.3776	0.4686	0.2422	0.2175
MV vs GLAD	1.0000	0.8437	0.2437	0.1373	0.2453
MV vs PLAT	0.8000	0.0872	0.1226	0.0969	0.0150
GLAD vs PLAT	0.8000	0.0991	0.2067	0.1729	0.2010

For each pair, there are 5 cases that are associated with the changes of labeler counts for each dataset. Then, each pair corresponds to 10 cases regarding two datasets that have 6 pairs; we have 60 cases in total. If ρ_0 is adopted, we find that most of the results obtained by these algorithms do not have significant differences. Only three cases exhibit significant differences. They correspond to the pair of MV and PLAT when $x = 9$ for Hurricane Maria and $x = 11$ for Hurricane Sandy. In other words, only 5% of cases have significant differences.

Even though in Figures 5.6 and 5.7, PLAT performs the best for 7 cases among all algorithms, these four algorithms do not have significant differences. In other words, these four algorithms still have the similar accuracy values and their performances are similar.

ROC AUC Since our data is imbalanced, ROC AUC is adopted here to compare the performances among algorithms. The higher ROC AUC value, the better performance of an algorithm.

Figures 5.10 and 5.11 show the changes of ROC AUC values versus labeler counts among four algorithms. The horizontal and vertical axes show the labeler counts and ROC AUC values, respectively. In the figures, the ROC AUC values are averaged based on 30 times execution of each algorithm. With the same scenario as seen in the subsection of accuracy, we have 10 cases that are associated with two datasets. PLAT has the best ROC AUC for 6 cases, when $x \in \{5, 7, 9, 11\}$ for the Hurricane Maria dataset when $x \in \{3, 5\}$ for the Hurricane Sandy dataset. It also has the second best for 2 cases, when $x \in \{7, 9\}$ for Hurricane Sandy. GTIC has the best ROC AUC for 3 cases, when $x \in \{7, 9, 11\}$ for Hurricane Sandy, and the second best for 2 cases, when $x \in \{9, 11\}$ for Hurricane Maria. MV is the worst, since its ROC AUC values are always below the others as the labeler count changes. However,

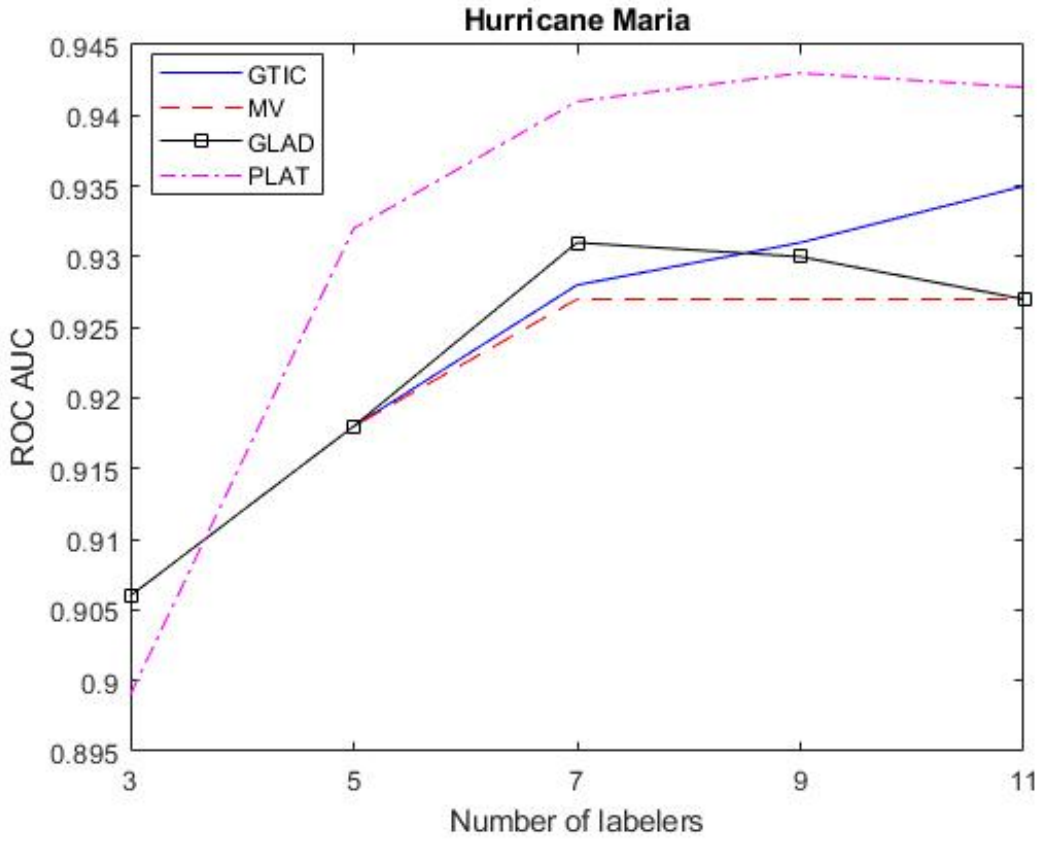

Figure 5.10 ROC AUC comparisons among four algorithms for Hurricane Maria.

PLAT still has some of the worst ROC AUC values compared to the others, such as when $x = 11$ for Hurricane Sandy.

F-measure F-measure is another metric that is adopted to validate the performance of an algorithm.

Figures 5.12 and 5.13 show the changes of F-measure values versus labeler counts among the four algorithms. The average of F-measure values based on 30 executions of each algorithm are shown. As mentioned in the previous sub-sections, there are 10 cases in total. PLAT has the best F-measure values for 6 cases, when $x \in \{5, 7, 9, 11\}$ and $x \in \{3, 5\}$ for Hurricane Maria and Hurricane Sandy, respectively. GTIC has the best F-measure values for 4 cases, when $x = 11$ and $x \in \{7, 9, 11\}$ for Hurricane Maria and Hurricane Sandy, respectively. Also, it has the second best F-measure

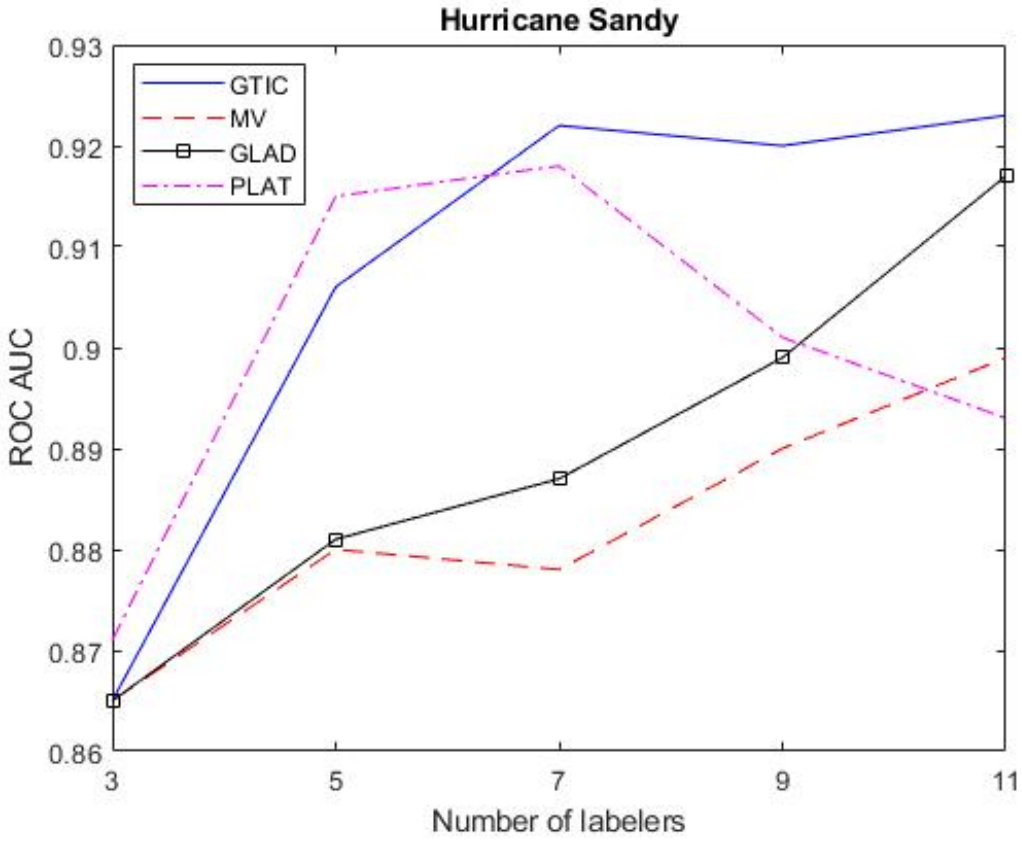

Figure 5.11 ROC AUC comparisons among four algorithms for Hurricane Sandy.

values for 2 cases, when $x = 11$ and $x = 5$ for Hurricane Maria and Hurricane Sandy, respectively. The GLAD method does not have any best F-measure value. MV is always the worst since its F-measure curve is below the others'. However, PLAT is not always good because it has some of the worst cases, such as when $x = 11$ for Hurricane Sandy.

Execution time The execution time is compared to test the execution speed of each algorithm. Table 5.4 shows the average execution time of each algorithm versus the number of labelers for both Hurricane Maria and Hurricane Sandy datasets.

It is obvious that GITC, MV, and PLAT have much less execution time than GLAD. Because GLAD gives parameters for both labelers and tasks, and needs to maximize the maximum likelihood function by using EM, it costs much time to obtain

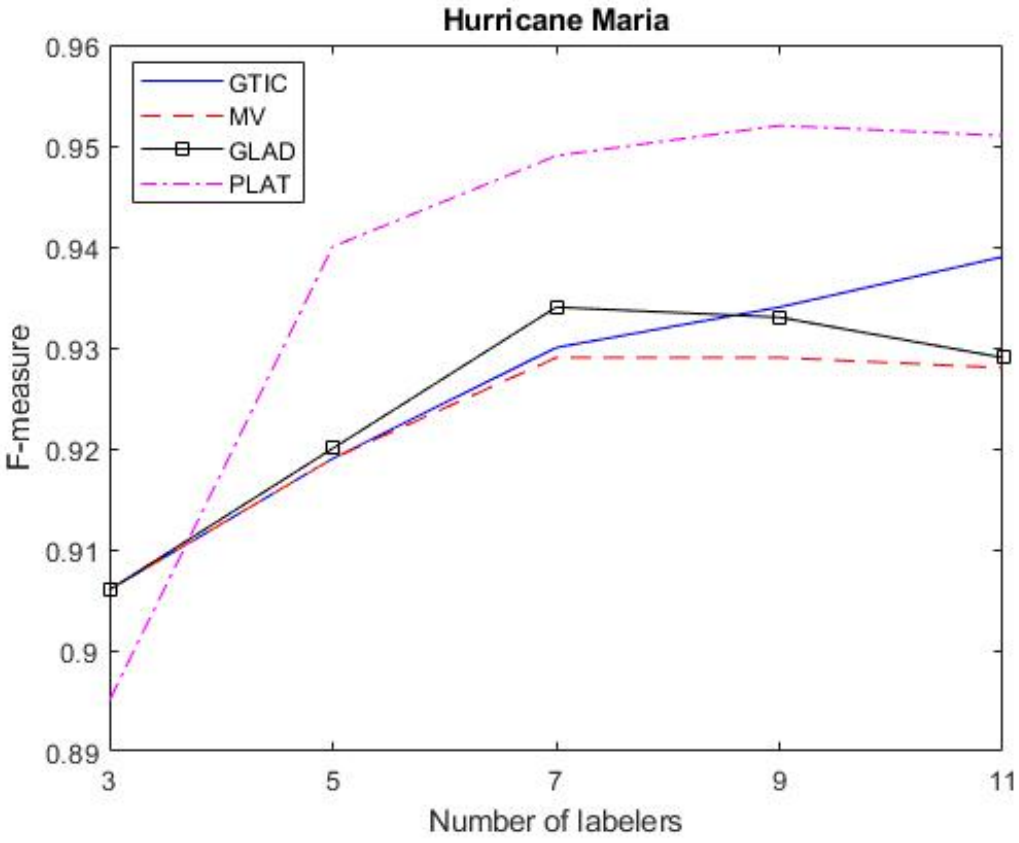

Figure 5.12 F-measure comparisons among four algorithms for Hurricane Maria.

its result. Compared with GLAD, the other algorithms, GTIC, MV, and PLAT, are much faster. GTIC takes a little longer time since it adopts a clustering process. In contrast, MV directly makes decisions on the majority labels and does not consider other factors, such as the quality of labelers and other instances. PLAT analyzes the PFD and estimates the threshold that splits the positive and negative portions. Thus, GTIC takes a little more time than MV and PLAT. The execution time of MV and PLAT are close, so it is hard to identify the faster method.

We compare these four algorithms using five evaluation metrics. The accuracy reflects the performance of classification performed from different algorithms. The McNemar test tells whether any two algorithms have any significant differences in accuracy. The ROC AUC and F-measure compare the performance of algorithms

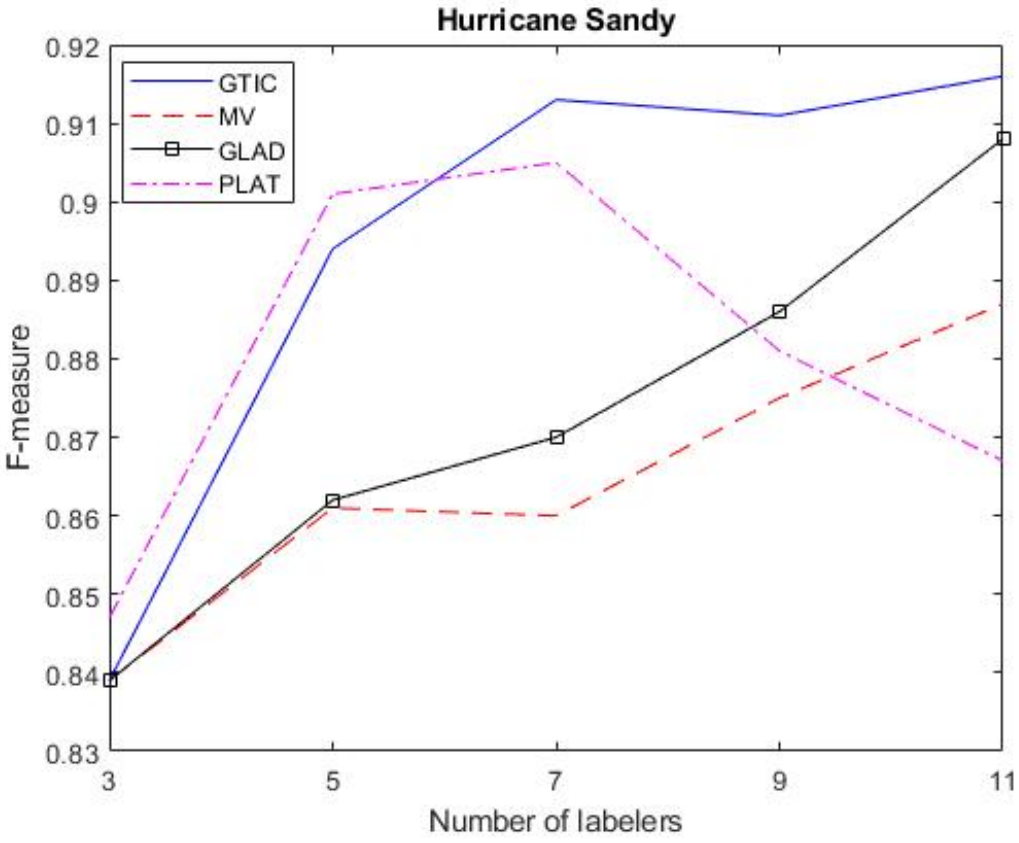

Figure 5.13 F-measure comparisons among four algorithms for Hurricane Sandy.

when there is imbalanced data. The execution time indicates the execution speed of algorithms. Overall, although PLAT has more cases that have better performances in accuracy, ROC AUC, and F-measure, it performs sometimes the worst, such as when $x = 11$ for Hurricane Sandy and when $x = 3$ for Hurricane Maria. Thus, it does not have the dominant advantage. Even though the curves of MV are below the others, it is not a significant difference in accuracy. Also, in Figures 5.6-5.7 and 5.10-5.13, the tendencies of GTIC and MV increase as labeler count increases. GLAD drops slightly sometimes. In contrast, PLAT drops significantly in Figures 5.7, 5.11 and 5.13 when the labeler count is large. Thus, the performance of PLAT cannot be guaranteed. Its robustness is worth a further study. In addition, PLAT and MV have less execution

Table 5.4 Execution Time for Hurricane Maria and Hurricane Sandy Data

Maria	3	5	7	9	11
GTIC	0.0548	0.0557	0.0601	0.0741	0.0850
MV	0.0191	0.0266	0.0348	0.0369	0.0358
GLAD	10.2266	10.7419	10.9184	11.1916	10.4113
PLAT	0.0220	0.0264	0.0305	0.0352	0.0392
Sandy	3	5	7	9	11
GTIC	0.0599	0.0547	0.0543	0.0544	0.0577
MV	0.0185	0.0217	0.0273	0.0272	0.0307
GLAD	9.7190	8.4106	9.0955	8.1751	8.1592
PLAT	0.0204	0.0246	0.0286	0.0311	0.0349

time than GTIC and GLAD. Overall, none of the four algorithms have a dominant advantage over the others.

5.3 Adaptive Majority Voting

With the explosion of social media data, their labeling task becomes a bottleneck for the machine learning and data mining community as they do not have their ground truth in general. Fortunately, crowdsourcing labeling systems, such as Amazon Mechanical Turk, provide cheap and fast ways to obtain a large quantity of labeled data [77]. However, for such a low price, the quality of labeled data is not guaranteed in general. Because many labelers want to earn more payment in the least time possible, the labels they produce tend to be inaccurate and sometimes wrong. In other words, the labels obtained via such systems are not always of desired quality [91].

5.3.1 Description of Adaptive Majority Voting

MV is a popular method, due to its simple implementation and high execution speed. As discussed in Section 5.2.1, if the labeling quality of labelers is greater than 0.5, then the integrated labeling quality tends to be high. If the labeling quality of labelers is high, the integrated labeling quality quickly approaches a high one as the number of labelers increases. If it is not that high, it approaches a final value slowly. To determine the labeling quality of labelers is not an easy task. Also, incapability of dealing with a tied case is a big disadvantage of MV. If a tied case occurs, MV randomly provides the label as 0 or 1. Even though we adopt an odd number of labelers, there still is a change to get a case close to a tied case. For example, given there are twenty-one people labeling a task; ten of them give a 0 as their labels, and the other eleven labelers give a 1 as their labels. Intuitively, this task is assigned as 1 by MV. Although the difference of labels between eleven 1s and ten 0s is only one, the integrated label becomes 1 since the majority vote is 1. It is important to note that this close difference of voters relies entirely on one labeler, and suppose he or she has a low labeling quality and has incorrectly marked the task, and if this is the case, that one labeler corrupts the label and it is not labeled correctly with the actual ground truth. We denote this situation as the close-to-tied case as the root of this error depends on a very close consensus between 0 and 1.

Unlike some studies, such as [83], that presume prior distribution for some parameters, which is normally unknown, the newly proposed Adaptive MV directly uses labels given by the labelers and does not require any other prior knowledge about labelers. Also, some work, such as [43], that assigns weights to each labeler, but the weights are only rough estimates and are inaccurate due to the limited observations in determining each weight. On the contrary, our method adaptively updates the weights by performing some iterations and to precisely assign proper weights to each labeler. In general, the purpose of Adaptive MV is to assign a weight to each labeler

based on his or her quality, which is determined from the number of values that the labeler's inputs and the majority's opinions, or majority voting results, are identical. Then, we construct a new MV model, compiled with all the updated weights, to and determine new integrated labels. The new integrated labels are used to adjust the previous weight of each labeler to ensure that the labeler is given the most accurate weight possible. We repeat this process of acclimating the weights until the difference between the present weight and new one is agreed to be negligible. Furthermore, for the labelers that have many instances of labeling along the lines of the majority, in our method, we are able to apply extra emphasis on them by commissioning a greater weight. Generally, the labelers with higher weights have the greatest probabilities to match their labels with the majority such that their labels are given more of an impact on modifying and creating the next MV model. Also, the labelers that are more inconsistent with the majority voting results are updated with lower weights, such that their labels carry less impact in the next majority voting round. As discussed in [46], [84] and [110], as the number of labelers, who have a labeling quality is greater than 0.5, increase, the integrated labeling quality tends to yield a good performance. Even if the labeling quality of labeler is slightly greater than 0.5, it still has a relatively higher weight.

Using the labeling quality given in (5.3), the weight for the j-th labeler is obtained as follows:

$$w_j = \frac{p_j}{\sum_{j=1}^{R} p_j} \tag{5.18}$$

When the weights are considered, the estimated integrated label for the i-th task, $\widehat{y_i}$, is given as follows:

$$\widehat{y_i} = \begin{cases} 1 & \text{if } \dfrac{1}{R}\sum_{j=1}^{R}(w_j \cdot l_{i,j}) \geq 0.5, \\ 0 & \text{otherwise.} \end{cases} \tag{5.19}$$

The Adaptive MV method is shown in Algorithm 1.

Algorithm 1: Adaptive Majority Voting

Input : The labels matrix given as L;

Output: The estimated integrated labels, $\widehat{y}$;

Initialize weights as $w = [1, 1, ...1]_{1\times R}$ and $\Delta w = [1, 1, ...1]_{1\times R}$;

Set the threshold $\psi = [0.0001, 0.0001, ...0.0001]_{1\times R}$;

repeat

    $\widehat{y}$ is updated by using (5.19);

    w^{next} is updated by using (5.3) and (5.18);

    $\Delta w = w - w^{next}$;

    $w^{next} = w$;

until $|\Delta w_j| < \psi_j$;

The input is an $N \times R$ matrix L which is associated with the labels given by R labelers for N tasks. The algorithm output is the estimated integrated label vector, $\widehat{y}$. Step 1 initializes the weights as $w = [1, 1, ...1]_{1\times R}$ which treats every labeler has the same weight, and the weight difference is set to $\Delta w = [1, 1, ...1]_{1\times R}$. Once the changes of weights are less than the threshold, the weights have converged to the agreed endpoint. The estimated integrated labels, $\widehat{y}$, are updated with (5.19) in Step 4. The weights are updated with (5.3) and (5.18) in Step 5. The updated weight vector is denoted as w^{next}. The weight difference is computed in Step 6, i.e., $\Delta w = w - w^{next}$. This algorithm stops when the absolute value of weight difference, $|\Delta w|$, is less than the threshold.

5.3.2 Experimental Results

In this section, we adopt the Adaptive MV method on the two real datasets that are associated with Hurricane Sandy and Hurricane Maria and compare it with the conventional MV method. In order to verify the performance of Adaptive MV, we randomly replace some original labelers with noisy labelers in the raw datasets; the original labels are substituted by the noisy labels given by noisy labelers. Then, the data is randomly split into training and testing datasets. Note that there are 13 labelers in our datasets. Because PLAT has more best cases in the previous comparative study and MV is a basic and simple method, we choose MV and PLAT to be our adaptive MV's peers. All three algorithms are executed ten times. Two metrics, accuracy and ROC AUC, are utilized to verify the performance of algorithms. The t-test verifies whether they have significant difference or not. Figures 5.14 - 5.17 describe the performance of algorithms. Their x-axes scale is the number of noisy labelers and their y-axes correspond to the metric. Note that we set the labeling quality of noisy labeler to 0.2 and the values given in the figures are average values that are calculated from the ten executions. Since there are 13 labelers, if the number

of noisy labelers is greater than the half of number of labelers, then the majority becomes the noisy labelers. This is not our case since we assume that the majority of labelers have good labeling qualities.

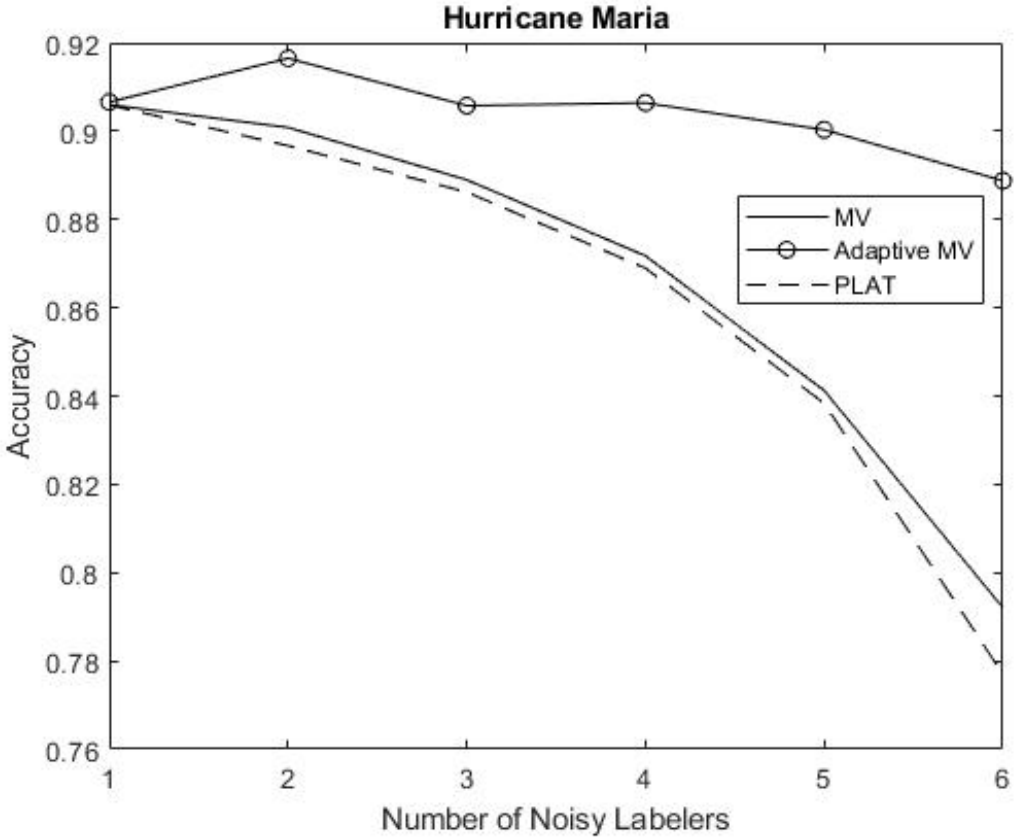

Figure 5.14 Accuracy comparisons between MV and Adaptive MV with different numbers of noisy labelers on Hurricane Maria data.

Figures 5.14 and 5.15 show the comparisons based on accuracy for Hurricane Maria and Sandy, respectively. In Figure 5.14, Adaptive MV is the best out of the three algorithms. All three algorithms begin relatively close when the number of noisy labelers, $x = 1$. Note that x represents the number of noisy labelers. As x increases, $x \in \{2, 3, 4, 5, 6\}$, the accuracies of each algorithm diverge with Adaptive MV performing the best, MV performing the second best, and PLAT performing the worst. MV and PLAT are very close, but MV runs consistently a little better than

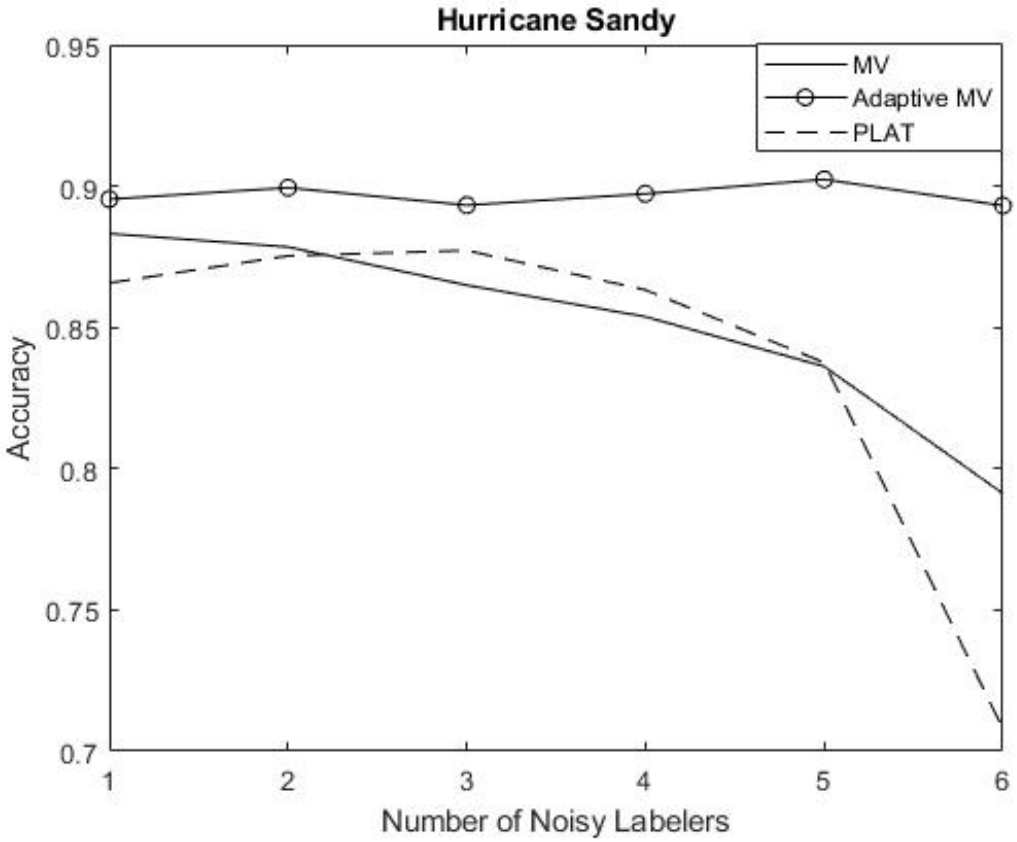

Figure 5.15 Accuracy comparisons between MV and Adaptive MV with different numbers of noisy labelers on Hurricane Sandy data.

PLAT in all cases. Thus, in this set, we conclude that our proposed method, Adaptive MV, is the best, MV comes in second place, and PLAT is the third but very close to MV. In Figure 5.15, Adaptive MV is clearly the best out of the three algorithms as it consistently performs the best in all cases, $x \in \{1, 2, 3, 4, 5, 6\}$. When $x = 1$, MV performs better than PLAT, but when $x \in \{3, 4, 5\}$, PLAT runs slightly better than MV. In the transition between five and six noisy labelers, PLAT's accuracy drops relatively more quickly than MV's accuracy, creating a sharp division between their accuracies when $x = 6$. Note that as the accuracies of PLAT and MV decline, the accuracy of Adaptive MV stabilizes at around 0.89.

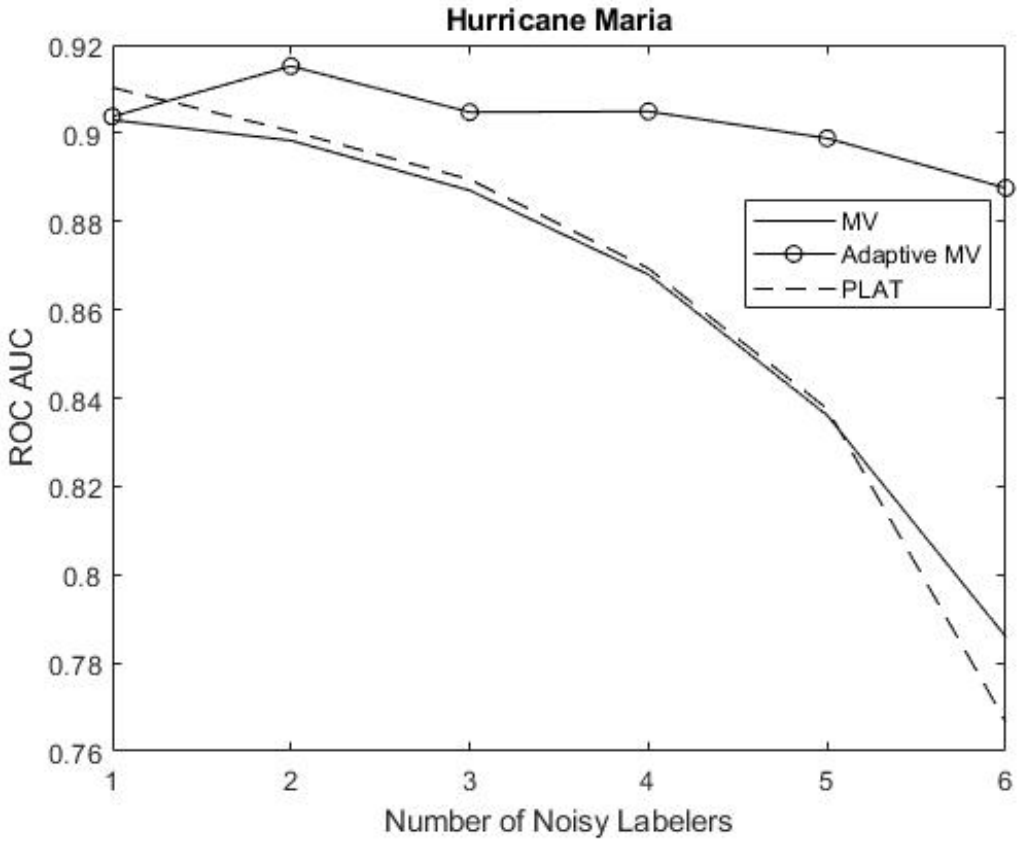

Figure 5.16 ROC AUC comparisons between MV and Adaptive MV with different numbers of noisy labelers on Hurricane Maria data.

Figures 5.16 and 5.17 describe the comparisons based on ROC AUC. In Figure 5.16, Adaptive MV proves to be the best out of the three algorithms. When $x = 1$, PLAT is the best, Adaptive MV is the second best, and MV is the worst. However, as x increases, the compared performances of each algorithm are as follows: Adaptive MV the best, PLAT the second best, and MV the worst. MV and PLAT are numerically close, but PLAT is slightly better than MV in four cases, when $x \in \{1, 2, 3, 4\}$, out of the six. In Figure 5.17, Adaptive MV is clearly the best out of the three algorithms since it consistently performs the best in all cases. When $x = 1$, MV performs better than PLAT, but when $x \in \{3, 4, 5\}$, PLAT runs slightly better

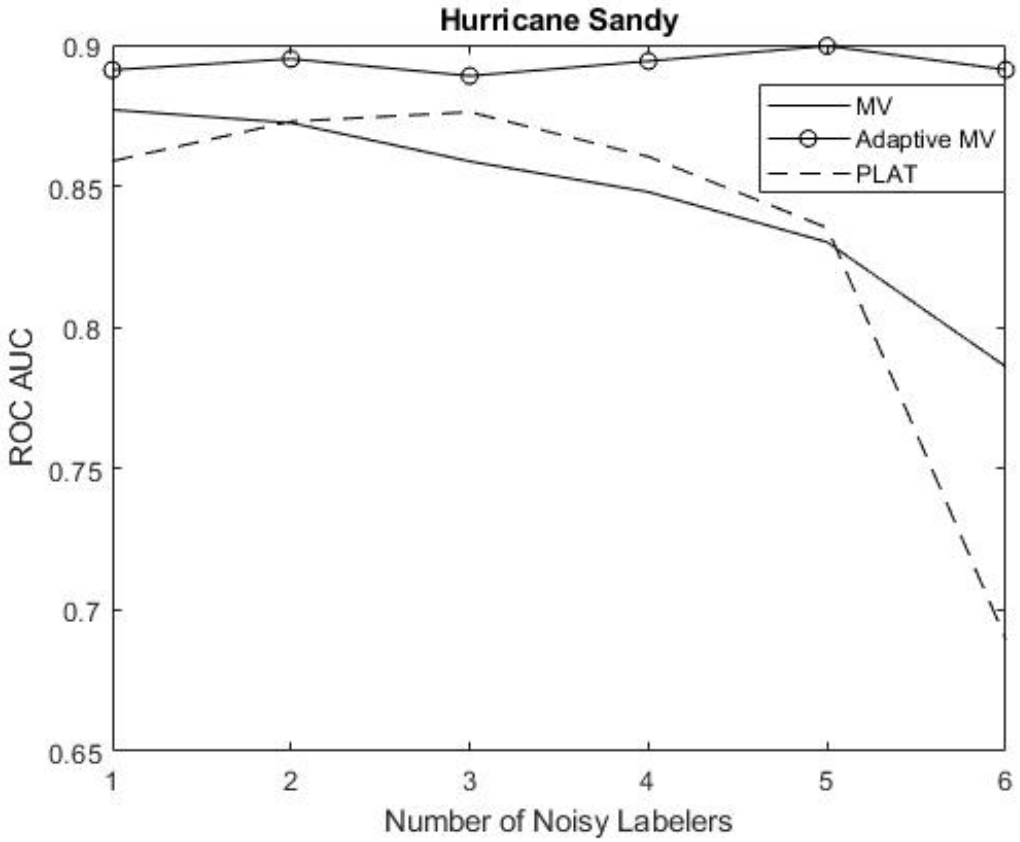

Figure 5.17 ROC AUC comparisons between MV and Adaptive MV with different numbers of noisy labelers on Hurricane Sandy data.

than MV. In the transition between five and six noisy labelers, PLAT's ROC AUC values drop relatively more quickly than MV's ROC AUC values.

In Tables 5.5-5.8, the t-test is adopted to verify if there is any significant difference that exists among algorithms based on accuracy and ROC AUC. Note that if a p-value is less than 0.05, it is put in bold font in the tables. The first row shows the number of noisy labelers, and the first column represents a pair of algorithms.

In Table 5.5, each pair is compared with six cases corresponding to the number of noisy labelers from 1 to 6 for Hurricane Maria. The pair, MV and Adaptive MV, have significant differences for five cases in accuracy when $x \in \{2, 3, 4, 5, 6\}$. The

Table 5.5 p-value Comparisons based on Accuracy between Two Algorithms for Hurricane Maria Data with t-test

	1	2	3	4	5	6
MV vs Adaptive MV	0.2172	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
Adaptive MV vs PLAT	0.7487	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
MV vs PLAT	0.9636	**0.0056**	0.2169	0.5178	0.0732	0.1683

Table 5.6 p-value Comparisons based on Accuracy between Two Algorithms for Hurricane Sandy Data with t-test

	1	2	3	4	5	6
MV vs Adaptive MV	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
Adaptive MV vs PLAT	0.0919	0.1226	0.2171	**0.0012**	**0.0000**	**0.0008**
MV vs PLAT	0.2969	0.8297	0.3258	0.2224	0.4274	0.0505

Table 5.7 p-value Comparisons based on ROC AUC between Two Algorithms for Hurricane Maria Data with t-test

	1	2	3	4	5	6
MV vs Adaptive MV	0.1841	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
Adaptive MV vs PLAT	**0.0045**	**0.0000**	**0.0000**	**0.0001**	**0.0000**	**0.0000**
MV vs PLAT	**0.0021**	**0.0105**	0.2108	0.8170	0.2585	0.1680

Table 5.8 p-value Comparisons based on ROC AUC between Two Algorithms for Hurricane Sandy Data with t-test

	1	2	3	4	5	6
MV vs Adaptive MV	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
Adaptive MV vs PLAT	0.1118	0.2190	0.3863	**0.0033**	**0.0000**	**0.0012**
MV vs PLAT	0.3523	0.9773	0.2374	0.1758	**0.0420**	0.0507

pair, Adaptive MV and PLAT, have significant differences for five cases when $x \in \{2, 3, 4, 5, 6\}$. The pair, MV and PLAT, only have one significant difference when $x = 2$. In Table 5.6 for Hurricane Sandy, MV and Adaptive MV have significant differences for all six cases. The pair, Adaptive MV and PLAT, have significant differences for three cases when $x \in \{4, 5, 6\}$. However, there is no significant difference for the pair of MV and PLAT. In Tables 5.5 and 5.6, there are twelve cases for each pair of algorithms by noting that either dataset contains six cases corresponding to the number of noisy labelers from one to six. The pair, MV and Adaptive MV, have eleven out of twelve cases that show significant differences. The pair, Adaptive MV and PLAT, have significant differences in eight out of twelve cases. In contrast, the pair, MV and PLAT, only have one case that has significant difference. Overall, using the t-test, Tables 5.5 and 5.6, and Figures 5.14 and 5.15, we conclude that our proposed method, Adaptive MV, is much better than MV and PLAT since its low p-values reject the null hypothesis. Note that rejecting the null hypothesis means that a significant difference exists. However, between MV and PLAT, we are unable to declare which method is better because almost all of the p-values are greater than 0.05 and it is also not clear in any of the figures since they show similar performances.

The results are similar when the ROC AUC is adopted. In total, there are eleven out of twelve cases with significant differences for the pair, MV and Adaptive MV. The pair, Adaptive MV and PLAT, have nine out of twelve cases with significant differences. However, there are only three cases with significant differences for the pair of MV and PLAT. In general, taking Tables 5.7 and 5.8, and Figures 5.16 and 5.17 into consideration, we conclude that our proposed method, Adaptive MV, is much better than MV and PLAT based on ROC AUC. However, between MV and PLAT, we are unable to declare which method is better because almost all of the p-values are greater than 0.05. It is unclear in any of the figures since they show similar performances.

In conclusion, the Adaptive MV method outperforms other methods, MV and PLAT as the number of noisy labelers increases. MV and PLAT do not have significant differences. Thus, their performance is similar. Note that when there are no noisy labelers, MV, PLAT and Adaptive MV show no significant differences.

CHAPTER 6

CONCLUSION

6.1 Summary of Contributions

This work aims at three core directions of rare event analysis: 1) exploring the temporal-spatial pattern of social activities; 2) classifying short-texts; and 3) dealing with no-ground-truth problem of short texts. Finding the temporal-spatial pattern of social activities can help us understand the real impacts that people have suffered and can be used to evaluate impacts during the arrival of a rare event. The second direction identifies rare-event-related and unrelated short texts by using a fuzzy logic-based feature extraction and classification methods. The last direction focuses on a no-ground-truth problem, since many short texts do not have ground truth.

For the first direction, a reliable and robust temporal-spatial pattern of social media activities can reflect real impacts that people have suffered, and be used to evaluate impacts during the arrival of a rare event. Regularities between virtual and real worlds are explored in this work. In Chapter 3, using the proposed clustering-algorithm-based data processing methods and analyzing the social media data in the virtual world, more precise and accurate temporal information can be obtained regarding a rare event. First, it verifies that there is a strong connection between the variations of social media activities and the evolution of a rare event in a time domain. Second, it provides a more precise and believable impacted time point of a rare event like a hurricane. Furthermore, it reveals that time differences exist and are different for varying cities. Investigating and revealing the differences are helpful in building the temporal pattern of the virtual world or social media activities during the occurrence of a rare event. Since social media activities provide timely information,

they can accurately reflect the human's behaviors, mood, and awareness in real time. The study of time differences is one important component of temporal patterns. It provides an approach to track, understand, analyze and evaluate the evolution of a rare event precisely and rapidly in a time domain. Then, relevant departments and organizations, and even individuals can start to better prepare for extreme events in advance.

Next, this work aims at a short-text classification problem, since Chapter 3 only uses the keyword search method that may filter important content and information out. Chapter 4 deals with the issue that is faced in Chapter 3. It mainly contains two parts. First, a novel feature extraction approach is provided to extract features from short texts. Second, a fuzzy logic-based text classification method is proposed to deal with the binary classification problem of short texts. The fuzzy rules and membership functions are given. The dataset crawled during Hurricane Sandy is used to verify the effectiveness of the proposed methods. With comparisons among five commonly used defuzzification methods, we draw a conclusion that centroid defuzzification is more effective and efficient than bisector, LOM, MOM and SOM. In addition, a comparison with the widely used keyword search method is conducted. The experimental results reveal that the proposed feature extraction and fuzzy logic-based classification methods are more suitable to find rare event-relevant messages. Fuzzy-logic methods are able to easily beat the keyword search method in NPV, AUC, and change rate at some small sacrifice of precision value for our case study. In addition, we compare our feature extraction method with word2vec by using the same classification methods. The results reflect that the proposed fuzzy logic-based feature extraction method is superior to the word2vec-based one.

Lastly, since the ground-truth labels of numerous social media data do not exist and are hard to determine, a no-ground-truth problem is an important research issue to be addressed. Automatically finding ground-truth labels for short texts is

a key to uncover users' real meanings. Improper labels lead the analysis of social media data into dilemmas. It is difficult to explore humans' activities and predict the possible influences on human beings' lives when a rare event takes place. Two real social media data sets, Hurricane Sandy and Maria, collected from Twitter are used to verify the performance of four existing algorithms. Overall, none of the four algorithms, i.e., PLAT, GLAD, GTIC, and MV, have a dominant advantage over the others. In addition, the proposed method, Adaptive MV, is compared with MV and PLAT methods. It outperforms MV and PLAT for the dataset containing poor labelers.

6.2 Limitations and Future Research

Even though this study deals with many issues facing in the rare event analysis, it still has some limitations. First of all, currently, we only deal with an event that lasts a relatively long time. The rare event may last for several days, such as hurricanes, and people may obtain weather forecast and alerts before their arrival. Amid the aftermath of their arrival, people may still be impacted by them. They belong to long-term events and have a pre-defined name. However, some short-term events are not concerned, such as earthquakes. There is no alert or extremely short (like a few minutes) before its arrival, and it does not have a specific name in general before they happen. Time differences, temporal-spatial patterns of social activities, and short texts are distinct from our current scenarios. Secondly, since different events may cause different impacts, people's feelings, attitudes, and behaviors are entirely different. Their short-text posts are distinct, so the feature extraction of short texts may be different from our current case. Thirdly, in this study, the completeness of data collected during rare events are limited, because only Twitter data are adopted and focus on the short texts using English. Multiple sources, such as videos, multiple social media platforms, such as Facebook, and multiple languages could be taken into

consideration. Lastly, the ground-truth inference methods only utilize labels obtained from labelers. However, combining with the feature extracted from texts is able to improve the performance of short-texts classification.

In order to conquer the limitations aforementioned, the future work should focus on four aspects. Firstly, multiple types of rare events can be well investigated. Deep learning based methods, such as [13, 70, 78, 90, 104], will be studied to analyze such events. Secondly, when dealing with the short-texts classification problem for different rare events, the fuzzy rules and membership function are planned to be adjusted and selected by using some intelligent optimization algorithms [23,37,38,40,41,67,94,115]. Thirdly, the data must be collected from multiple sources, social media platforms, and languages, such as the data adopted in [90] and [36]. Lastly, for the no-ground-truth problem, we should focus on analyzing the reliability of Adaptive MV, including theoretical proofs, and extending it to other real-world datasets. Also, the features extracted from short texts should be combined and taken into consideration. When the extracted features are combined with labelers' intelligence, the accuracy of short-texts classification is expected to be improved.

BIBLIOGRAPHY

[1] Federal emergency management agency. [Online]. Available: http://fema.maps.arcgis.com/home/webmap/viewer.html?webmap=307dd522499d4a44a33d7296a5da5ea0. Accessed: July 20, 2013.

[2] Hurricane Sandy. [Online]. Available: https://en.wikipedia.org/wiki/Hurricane_Sandy. Accessed: July 20, 2019.

[3] List of Maryland hurricanes. [Online]. Available: https://en.wikipedia.org/wiki/List_of_Maryland_hurricanes_(1950-present). Accessed: July 21, 2019.

[4] List of New York hurricanes. [Online]. Available: https://en.wikipedia.org/wiki/List_of_New_York_hurricanes. Accessed: July 20, 2019.

[5] Natural language toolkit. [Online]. Available: http://www.nltk.org/. Accessed: July 20, 2019.

[6] H. Abdi. The Kendall rank correlation coefficient. *Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, CA*, pages 508–510, 2007.

[7] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. Finding high-quality content in social media. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 183–194. ACM, Palo Alto, USA, Feb. 11–12, 2008.

[8] M. Allahbakhsh, B. Benatallah, A. Ignjatovic, H. R. Motahari-Nezhad, E. Bertino, and S. Dustdar. Quality control in crowdsourcing systems: Issues and directions. *IEEE Internet Computing*, 17(2):76–81, 2013.

[9] J. An, Q. Kang, L. Wang, and Q. D. Wu. Mussels wandering optimization: an ecologically inspired algorithm for global optimization. *Cognitive Computation*, 5(2):188–199, 2013.

[10] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[11] E. Aslanian, M. Radmanesh, and M. Jalili. Hybrid recommender systems based on content feature relationship. *IEEE Transactions on Industrial Informatics*, 2016, DOI: 10.1109/TII.2016.2631138, Early Access.

[12] S. Banerjee, K. Ramanathan, and A. Gupta. Clustering short texts using wikipedia. In *Proceedings of the 30th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 787–788. ACM, Amsterdam, Netherlands, July 23–27, 2007.

[13] D. P. Bertsekas. Feature-based aggregation and deep reinforcement learning: A survey and some new implementations. *IEEE/CAA Journal of Automatica Sinica*, 6(1):1–31, 2018.

[14] W. J. Bi, M. L. Cai, M. Q. Liu, and G. Li. A big data clustering algorithm for mitigating the risk of customer churn. *IEEE Transactions on Industrial Informatics*, 12(3):1270–1281, 2016.

[15] N. Bidi and Z. Elberrichi. Feature selection for text classification using genetic algorithms. In *Proceedings of 2016 8th International Conference on Modelling, Identification and Control (ICMIC)*, pages 806–810. IEEE, Algiers, Algeria, Nov. 15–17, 2016.

[16] S. R. Bishop, T. Preis, H. S. Moat, P. Treleaven, and H. E. Stanley. Quantifying the digital traces of hurricane Sandy on flickr. *Scientific Reports*, 3, 2013.

[17] J. Bollen, H. N. Mao, and X. J. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[18] S. Bouix, M. Martin-Fernandez, L. Ungar, M. Nakamura, M. S. Koo, R. W. McCarley, and M. E. Shenton. On evaluating brain tissue classifiers without a ground truth. *Neuroimage*, 36(4):1207–1224, 2007.

[19] C. Caragea, A. C. Squicciarini, S. Stehle, K. Neppalli, and A. H. Tapia. Mapping moods: Geo-mapped sentiment analysis during hurricane Sandy. In *Proceedings of ISCRAM*, pages 1–10, University Park, Pennsylvania, USA, May, 2014.

[20] C. Chen, D. Neal, and M. C. Zhou. Understanding the evolution of a disaster: a framework for assessing crisis in a system environment (facse). *Natural Hazards*, 65(1):407–422, 2013.

[21] S. L. Cutter. Are we asking the right question. *What is a disaster*, pages 39–48, 2005.

[22] B. Dhingra, Z. Zhou, D. Fitzpatrick, M. Muehl, and W. W. Cohen. Tweet2vec: Character-based distributed representations for social media. *arXiv preprint arXiv:1605.03481*, 2016.

[23] Z. H. Ding, Y. Zhou, and M. C. Zhou. Modeling self-adaptive software systems by fuzzy rules and Petri nets. *IEEE Transactions on Fuzzy Systems*, 26(2):967–984, 2018.

[24] P. S. Dodds, K. D. Harris, I. M. Kloumann, C. A. Bliss, and C. M. Danforth. Temporal patterns of happiness and information in a global social network: Hedonometrics and twitter. *PloS one*, 6(12):e26752, 2011.

[25] H. Dong, M. Halem, and S. J. Zhou. Social media data analytics applied to hurricane Sandy. In *Proceedings of 2013 International Conference on Social Computing*, pages 963–966. IEEE, Alexandria, VA, USA, Sep. 8–14, 2013.

[26] P. Donmez, J. G. Carbonell, and J. Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268. ACM, Paris, France, June 28–July 1, 2009.

[27] S. Dow, A. Kulkarni, S. Klemmer, and B. Hartmann. Shepherding the crowd yields better work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, pages 1013–1022. ACM, Seattle, Washington, USA, Feb. 11–15, 2012.

[28] D. Ediger, S. Appling, E. Briscoe, R. McColl, and J. Poovey. Real-time streaming intelligence: Integrating graph and nlp analytics. In *Proceedings of 2014 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–6. IEEE, Waltham, MA, USA, Sep. 9–11, 2014.

[29] A. L. Edwards. Note on the "correction for continuity" in testing the significance of the difference between correlated proportions. *Psychometrika*, 13(3):185–187, 1948.

[30] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[31] D. B. Figueiredo Filho, R. Paranhos, E. C. Rocha, M. Batista, Silva J. A., M. W. Santos, and J. G. Marino. When is statistical significance not significant? *Brazilian Political Science Review*, 7(1):31–55, 2013.

[32] M. R. Frank, L. Mitchell, P. S. Dodds, and C. M. Danforth. Happiness and the patterns of life: A study of geolocated tweets. *Scientific Reports*, 3:2625, 2013.

[33] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of IJcAI*, volume 7, pages 1606–1611, 2007.

[34] E. Gabrilovich and S. Markovitch. Feature generation for text categorization using world knowledge. In *Proceedings of the 19th international joint conference on Artificial intelligence(IJCAI)*, volume 5, pages 1048–1053, Edinburgh, Scotland, July 30–Aug. 5, 2005.

[35] C. Gao and J. M. Liu. Network-based modeling for characterizing human collective behaviors during extreme events. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):171–183, 2017.

[36] H. J. Gao, G. Barbier, and R. Goolsby. Harnessing the crowdsourcing power of social media for disaster relief. *IEEE Intelligent Systems*, 26(3):10–14, 2011.

[37] K. Z. Gao, Z. G. Cao, L. Zhang, Z. H. Chen, Y. Y. Han, and Q. K. Pan. A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems. *IEEE/CAA Journal of Automatica Sinica*, 6(4):904–916, 2019.

[38] S. C. Gao, M. C. Zhou, Y. R. Wang, J. J. Cheng, H. Yachi, and J. H. Wang. Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction. *IEEE transactions on neural networks and learning systems*, 30(2):601–614, 2018.

[39] X. Y. Guan and C. Chen. Using social media data to understand and assess disasters. *Natural Hazards*, 74(2):837–850, 2014.

[40] X. W. Guo, S. X. Liu, M. C. Zhou, and G. D. Tian. Dual-objective program and scatter search for the optimization of disassembly sequences subject to multiresource constraints. *IEEE Transactions on Automation Science and Engineering*, 15(3):1091–1103, 2017.

[41] L. H. He, D. Hu, M. Wan, Y. Wen, K. M. Von Deneen, and M. C. Zhou. Common bayesian network for classification of eeg-based multiclass motor imagery bci. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(6):843–854, 2016.

[42] H. Hellendoorn and C. Thomas. Defuzzification in fuzzy controllers. *Journal of Intelligent & Fuzzy Systems*, 1(2):109–123, 1993.

[43] J. Hernández-González, I. Inza, and J. A. Lozano. A note on the behavior of majority voting in multi-class domains with biased annotators. *IEEE Transactions on Knowledge and Data Engineering*, 31(1):195–200, 2018.

[44] X. Hu, N. Sun, C. Zhang, and T. S. Chua. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 919–928. ACM, Hong Kong, China, Nov. 2–6, 2009.

[45] Y. Huang, H. Dong, Ye. Yesha, and S. J. Zhou. A scalable system for community discovery in twitter during hurricane Sandy. In *Proceedings of 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pages 893–899. IEEE, Chicago, IL, USA, May 26–29, 2014.

[46] P. G. Ipeirotis, F. Provost, V. S. Sheng, and J. Wang. Repeated labeling using multiple noisy labelers. *Data Mining and Knowledge Discovery*, 28(2):402–441, 2014.

[47] A. K. Jain and R. C. Dubes. Algorithms for clustering data. 1988.

[48] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[49] Q. W. Jiang, W. Wang, X. Han, S. S. Zhang, X. Y. Wang, and C. Wang. Deep feature weighting in naive bayes for chinese text classification. In *Proceedings of 2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pages 160–164. IEEE, Beijing, China, Aug. 17–19, 2016.

[50] H. J. Jung and M. Lease. Improving consensus accuracy via z-score and weighted voting. In *Human Computation*, 2011.

[51] Q. Kang, S. Y. Liu, M. C. Zhou, and S. S. Li. A weight-incorporated similarity-based clustering ensemble method based on swarm intelligence. *Knowledge-Based Systems*, 104:156–164, 2016.

[52] D. R. Karger, S. Oh, and D. Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2014.

[53] D. R. Karger, S. Oh, and D. Shah. Budget-optimal crowdsourcing using low-rank matrix approximations. In *Proceedings of 2011 IEEE 49th Annual Allerton Conference on Communication, Control and Computing (Allerton)*, pages 284–291. IEEE, Monticello, IL, USA, Sep. 28–30, 2011.

[54] R. Korolov, D. Lu, J. J. Wang, G. Y. Zhou, C. Bonial, C. Voss, L. Kaplan, W. Wallace, J. W. Han, and H. Ji. On predicting social unrest using social media. In *Proceedings of 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 89–95. IEEE, San Francisco, CA, USA, Aug. 18–21, 2016.

[55] B. Kosko and M. Toms. *Fuzzy thinking: The new science of fuzzy logic*. Hyperion New York, 1993.

[56] V. Kulkarni, R. Al-Rfou, B. Perozzi, and S. Skiena. Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web*, pages 625–635. International World Wide Web Conferences Steering Committee, Florence, Italy, May 18–22, 2015.

[57] A. Kumar and M. Lease. Modeling annotator accuracies for supervised learning. In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, pages 19–22, Hong Kong, China, Feb. 9, 2011.

[58] M. Laituri and K. Kodrich. On line disaster response community: People as sensors of high magnitude disasters using internet gis. *Sensors*, 8(5):3037–3055, 2008.

[59] T. Lansdall-Welfare, S. Sudhahar, G. A. Veltri, and N. Cristianini. On the coverage of science in the media: A big data study on the impact of the fukushima disaster. In *Proceedings of 2014 IEEE International Conference on Big Data (Big Data)*, pages 60–66. IEEE, Washington, DC, USA, Oct. 27–30, 2014.

[60] A. Likas, N. Vlassis, and J. J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36(2):451–461, 2003.

[61] J. Lilleberg, Y. Zhu, and Y. Q. Zhang. Support vector machines and word2vec for text classification with semantic features. In *Proceedings of 2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pages 136–140. IEEE, Beijing, China, July 6–8, 2015.

[62] H. Y. Liu, M. C. Zhou, and Q. Liu. An embedded feature selection method for imbalanced data classification. *IEEE/CAA Journal of Automatica Sinica*, 6(3):703–715, 2019.

[63] X. S. Lu and M. C. Zhou. Analyzing the evolution of rare events via social media data and k-means clustering algorithm. In *Proceedings of 2016 IEEE 13th International Conference on Networking, Sensing, and Control (ICNSC)*, pages 1–6. IEEE, Mexico City, Mexico, Apr. 28–30, 2016.

[64] X. S. Lu, M. C. Zhou, and L. Qi. Analyzing temporal-spatial evolution of rare events by using social media data. In *Proceedings of 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2684–2689. IEEE, Banff, AB, Canada, Oct. 5–8, 2017.

[65] W. J. Luan, G. J. Liu, and C. J. Jiang. Collaborative tensor factorization and its application in poi recommendation. In *Proceedings of 2016 IEEE 13th International Conference on Networking, Sensing, and Control (ICNSC)*, pages 1–6. IEEE, Mexico City, Mexico, Apr. 28–30, 2016.

[66] C. N. Manikopoulos, M. C. Zhou, and S. S. Nerurkar. Design and implementation of fuzzy logic controllers for a heat exchanger in a water-for-injection system. *Journal of Intelligent & Fuzzy Systems*, 3(1):43–57, 1995.

[67] X. H. Meng, J. Li, M. C. Zhou, X. Z. Dai, and J. P. Dou. Population-based incremental learning algorithm for a serial colored traveling salesman problem. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(2):277–288, 2018.

[68] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[69] G. Miller. *WordNet: An electronic lexical database*. MIT Press, 1998.

[70] Khan Muhammad, Jamil Ahmad, and Sung Wook Baik. Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing*, 288:30–42, 2018.

[71] S. Naaz, A. Alam, and R. Biswas. Effect of different defuzzification methods in a fuzzy based load balancing application. *International Journal of Computer Science Issues (IJCSI)*, 8(5):261, 2011.

[72] R. Narayanam and Y. Narahari. A shapley value-based approach to discover influential nodes in social networks. *IEEE Transactions on Automation Science and Engineering*, 8(1):130–147, 2011.

[73] O. J. Oyelade, O. O. Oladipupo, and I. C. Obagbuwa. Application of k means clustering algorithm for prediction of students academic performance. *arXiv preprint arXiv:1002.2425*, 2010.

[74] T. Preis, H. S. Moat, S. R. Bishop, P. Treleaven, and H. E. Stanley. Quantifying the digital traces of hurricane Sandy on flickr. *Scientific Reports*, 3:3141, 2013.

[75] J. D. Prusa and T. M. Khoshgoftaar. Designing a better data representation for deep neural networks and text classification. In *Proceedings of 2016 IEEE 17th International Conference on Information Reuse and Integration (IRI)*, pages 411–416. IEEE, Pittsburgh, PA, USA, July 28–30, 2016.

[76] E. L. Quarantelli and R. R. Dynes. Response to social crisis and disaster. *Annual Review of Sociology*, 3(1):23–49, 1977.

[77] V. C. Raykar, S. P. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322, 2010.

[78] Z. Ren, K. Qian, Y. B. Wang, Z. X. Zhang, V. Pandit, A. Baird, and B. Schuller. Deep scalogram representations for acoustic scene classification. *IEEE/CAA Journal of Automatica Sinica*, 5(3):662–669, 2018.

[79] A. Rexha, M. Kröll, M. Dragoni, and R. Kern. Polarity classification for target phrases in tweets: a word2vec approach. In *Proceedings of European Semantic Web Conference*, pages 217–223. Springer, Heraklion, Greece, May 29–Jun. 2, 2016.

[80] M. Sahami and T. D. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th International Conference on World Wide Web*, pages 377–386. ACM, Edinburgh, Scotland, May 23–26, 2006.

[81] J. B. Sathe and M. P. Mali. A hybrid sentiment classification method using neural network and fuzzy logic. In *Proceedings of 2017 11th International Conference on Intelligent Systems and Control (ISCO)*, pages 93–96. IEEE, Coimbatore, India, Jan. 5–6, 2017.

[82] D. Shen, R. Pan, J. T. Sun, J. J. Pan, K. H. Wu, J. Yin, and Q. Yang. Query enrichment for web-query classification. *ACM Transactions on Information Systems (TOIS)*, 24(3):320–352, 2006.

[83] V. S. Sheng. Simple multiple noisy label utilization strategies. In *Proceedings of 2011 IEEE 11th International Conference on Data Mining*, pages 635–644. IEEE, Vancouver,BC, Canada, Dec. 11–14, 2011.

[84] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 614–622. ACM, Las Vegas, Nevada, USA, Aug. 24–27, 2008.

[85] A. Sheshasayee and G. Thailambal. A comparitive analysis of single pattern matching algorithms in text mining. In *Proceedings of 2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, pages 720–725. IEEE, Noida, India, Oct. 8–10, 2015.

[86] A. Sheth. Citizen sensing, social signals, and enriching human experience. *IEEE Internet Computing*, 13(4):87–92, 2009.

[87] G. Song, Y. Ye, X. L. Du, X. H. Huang, and S. F. Bie. Short text classification: A survey. *Journal of Multimedia*, 9(5):635–644, 2014.

[88] T. Spielhofer, R. Greenlaw, D. Markham, and A. Hahne. Data mining twitter during the uk floods: Investigating the potential use of social media in emergency management. In *Proceedings of 2016 3rd International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pages 1–6. IEEE, Vienna, Austria, Dec. 13–15, 2016.

[89] L. Suanmali, M. S. Binwahlan, and N. Salim. Sentence features fusion for text summarization using fuzzy logic. In *Proceedings of 2009 Ninth International Conference on Hybrid Intelligent Systems*, volume 1, pages 142–146. IEEE, Shenyang, China, Aug. 12–14, 2009.

[90] S. Subramani, H. Wang, H. Q. Vu, and G. Li. Domestic violence crisis identification from facebook posts based on deep learning. *IEEE access*, 6:54075–54085, 2018.

[91] D. P. Tao, J. Cheng, Z. T. Yu, K. Yue, and L. Z. Wang. Domain-weighted majority voting for crowdsourcing. *IEEE Transactions on Neural Networks and Learning Systems*, 30(1):163–174, 2018.

[92] S. Tatiraju and A. Mehta. Image segmentation using k-means clustering, em and normalized cuts. *Department of EECS*, 1:1–7, 2008.

[93] K. J. Tierney. From the margins to the mainstream? disaster research at the crossroads. *Annu. Rev. Sociol.*, 33:503–525, 2007.

[94] J. J. Wang and T. Kumbasar. Parameter optimization of interval type-2 fuzzy neural networks based on pso and bbbc methods. *IEEE/CAA Journal of Automatica Sinica*, 6(1):247–257, 2019.

[95] X. Wang and J. B. Bi. Bi-convex optimization to learn classifiers from multiple biomedical annotations. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14(3):564–575, 2017.

[96] X. F. Wang, M. S. Gerber, and D. E. Brown. Automatic crime prediction using events extracted from twitter posts. In *Proceedings of International Conference on Social Computing, Behavioral-cultural Modeling, and Prediction*, pages 231–238. Springer, College Park, MD, USA, Apr. 2–5, 2012.

[97] X. L. Wang, F. R. Wei, X. H. Liu, M. Zhou, and M. Zhang. Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 1031–1040. ACM, Glasgow, Scotland, UK, Oct. 24–28, 2011.

[98] P. Welinder and P. Perona. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *Proceedings of 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 25–32. IEEE, San Francisco, CA, USA, June 13–18, 2010.

[99] J. Whitehill, T. F. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems*, pages 2035–2043, 2009.

[100] S. A. Wood, A. D. Guerry, J. M. Silver, and M. Lacayo. Using social media to quantify nature-based tourism and recreation. *Scientific Reports*, 3:2976, 2013.

[101] Z. Xu, Y. H. Liu, N. Yen, L. Mei, X. F. Luo, X. Wei, and C. P. Hu. Crowdsourcing based description of urban emergency events using social media big data. *IEEE Transactions on Cloud Computing*, 2016.

[102] S. Yardi, D. Romero, G. Schoenebeck, et al. Detecting spam in a twitter network. *First Monday*, 15(1), 2010.

[103] W. T. Yih and C. Meek. Improving similarity measures for short segments of text. In *Proceedings of AAAI*, volume 7, pages 1489–1494, 2007.

[104] D. Yu and J. Y. Li. Recent progresses in deep learning based acoustic models. *IEEE/CAA Journal of Automatica Sinica*, 4(3):396–409, 2017.

[105] Z. Yu, H. X. Wang, X. M. Lin, and M. Wang. Understanding short texts through semantic enrichment and hashing. *IEEE Transactions on Knowledge and Data Engineering*, 28(2):566–579, 2016.

[106] L. A. Zadeh. Fuzzy logic= computing with words. In *Computing with Words in Information/Intelligent Systems 1*, pages 3–23. Springer, 1999.

[107] S. Zelikovitz and H. Hirsh. Transductive lsi for short text classification problems. In *Proceedings of Seventeenth International Florida Artificial Intelligence Research Symposium Conference (FLAIRS)*, pages 556–561. AAAI Press, Miami Beach, FL, USA, 2004.

[108] S. Zelikovitz and F. Marquez. Transductive learning for short-text classification problems using latent semantic indexing. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(02):143–163, 2005.

[109] J. Zhang, V. S. Sheng, J. Wu, and X. D. Wu. Multi-class ground truth inference in crowdsourcing with clustering. *IEEE Transactions on Knowledge and Data Engineering*, 28(4):1080–1085, 2016.

[110] J. Zhang, X. D. Wu, and V. S. Sheng. Imbalanced multiple noisy labeling. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):489–503, 2015.

[111] J. Zhang, X. D. Wu, and V. S. Sheng. Learning from crowdsourced labeled data: a survey. *Artificial Intelligence Review*, 46(4):543–576, 2016.

[112] W. Zhang, T. Yoshida, and X. J. Tang. A comparative study of tf* idf, lsi and multi-words for text classification. *Expert Systems with Applications*, 38(3):2758–2765, 2011.

[113] Y. C. Zhang, X. Chen, D. Y. Zhou, and M. I. Jordan. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 1260–1268, 2014.

[114] Y. T. Zhang, G. Ling, and Y. C. Wang. An improved tf-idf approach for text classification. *Journal of Zhejiang University-Science A*, 6(1):49–55, 2005.

[115] J. Zhao, S. X. Liu, M. C. Zhou, X. W. Guo, and L. Qi. Modified cuckoo search algorithm to solve economic power dispatch optimization problems. *IEEE/CAA Journal of Automatica Sinica*, 5(4):794–806, 2018.

[116] L. Zhao, F. Chen, C. T. Lu, and N. Ramakrishnan. Multi-resolution spatial event forecasting in social media. In *Proceedings of 2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 689–698. IEEE, Barcelona, Spain, Dec. 12–15, 2016.

[117] L. Zhao, J. Z. Chen, F. Chen, W. Wang, C. T. Lu, and N. Ramakrishnan. Simnest: Social media nested epidemic simulation via online semi-supervised deep learning. In *Proceedings of 2015 IEEE International Conference on Data Mining*, pages 639–648. IEEE, Atlantic City, NJ, USA, Nov. 14–17, 2015.