DENSITY-CONTOUR BASED FRAMEWORK FOR SPATIO-TEMPORAL CLUSTERING AND EVENT TRACKING IN TWITTER

A Dissertation Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Yongli Zhang

December 2018

DENSITY-CONTOUR BASED FRAMEWORK FOR SPATIO-TEMPORAL CLUSTERING AND EVENT TRACKING IN TWITTER

Yongli Zhang

APPROVED:

Christoph F. Eick, Chairman
Dept. of Computer Science

Ricardo Vilalta
Dept. of Computer Science

Guoning Chen
Dept. of Computer Science

Yunsoo Choi
Dept. of Earth and Atmospheric Sciences

Dean, College of Natural Sciences and Mathematics

Acknowledgment

This dissertation would not have been possible without the guidance of my advisor Prof. Christoph F. Eick. I would like to first thank him for being an awesome advisor, whom I have learned tons of research and time management skills from. The most important aspect that I learned from him is being confident that I can conduct better research. This journey was filled with both happy and frustrating times, but the best times I had at the University of Houston surrounded the motivating discussions on research ideas and the process that we collaborated in for scientific publications. Under Dr. Eick's supervision, I learned how to define a research problem, find a solution to it and finally, publish the results. To sum up, I would give Dr. Eick most of the credit for helping me to become the scientist I am today.

Aside from my advisor, I would also like to thank the other members of my dissertation committee: Guoning Chen, Ricardo Vilalta and Yunsoo Choi. Their expertise and invaluable comments helped to improve the dissertation significantly.

I would like to thank my collaborators from Data Analysis and Intelligent Systems Laboratory: Sujing Wang, Romita Banerjee and Karima Elgarroussi. Without them, much of the work in the dissertation would not have been possible. Sujing made a significant contribution to our paper submitted to ISMIS 2017; she implemented the SNN-based spatio-temporal clustering approach, which was used to compare with the serial approach we propose. Karima and Romita also contributed significantly to our tweet emotion mapping paper.

Last but not least, I would not be able to finish my dissertation without the support from my mom, dad, wife and friends. Thanks to them for always being my strongest support.

Previously Published Material

Chapter 1 revises a previous publication [77]: Y. Zhang and C. F. Eick. Novel clustering and analysis techniques for mining spatio-temporal data. In *Proc. ACM SIGSPATIAL PhD Workshop*, page 2, Dallas, TX, USA, November 4-7 2014.

Chapter 2 and Chapter 3.4 revise a previous publication [79]: Y. Zhang and C. F. Eick. St-copot: Spatio-temporal clustering with contour polygon trees. In *Proc. ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 84:1-84:4, Redondo Beach, CA, USA, November 7-10 2017.

Chapter 3.2 revises a previous publication [78]: Y. Zhang and C. F. Eick. St-dcontour: a serial, density-contour based spatiotemporal clustering approach to cluster location streams. In *Proc. ACM SIGSPATIAL International Workshop on GeoStreaming*, page 5, San Francisco, CA, USA, October 31 - November 3 2016.

Chapter 3.3 and Chapter 4 revise a previous publication [81]: Y. Zhang, S. Wang, A. M. Aryal, and C. F. Eick. "Serial" versus "parallel": a comparison of spatio-temporal clustering approaches. In *Proc. International Symposium on Methodologies for Intelligent Systems*, pages 396-403, Warsaw, Poland, June 26-29 2017.

Chapter 5 revises a previous publication [80]: Y. Zhang and C. F. Eick. A novel two-stage system for detecting and tracking events in twitter. In *Proc. IEEE International Conference on Artificial Intelligence and Knowledge Engineering*, pages 77-84, Laguna Hills, CA, USA, September 26-28 2018.

Chapter 5.5 briefly discusses a previous publication [6]: R. Banerjee, K. Elgarroussi, S. Wang, Y. Zhang, and C. F. Eick. Tweet emotion mapping: Understanding us emotions in time and space. In *Proc. IEEE International Conference on Artificial Intelligence and Knowledge Engineering*, pages 93-100, Laguna Hills, CA, USA, September 26-28 2018.

DENSITY-CONTOUR BASED FRAMEWORK FOR SPATIO-TEMPORAL CLUSTERING AND EVENT TRACKING IN TWITTER

An Abstract of a Dissertation

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Yongli Zhang

December 2018

Abstract

Due to the advances in remote sensors and sensor networks, different types of spatio-temporal datasets have become increasingly available. Revealing interesting spatio-temporal patterns from such datasets is very important, as it has broad applications, such as understanding climate change, epidemics detection, and earthquake analysis. The main focus of this research is the development of spatio-temporal clustering frameworks.

In this dissertation, we introduce a density-contour based framework for spatio-temporal clustering including several novel serial, density-contour based spatio-temporal clustering algorithms: ST-DCONTOUR, ST-DPOLY, and ST-COPOT. They all rely on a three-phase clustering approach, which takes the point cloud stream as input and divides it into batches based on fixed-size time windows. Next, a density estimation approach and contouring algorithms are employed to obtain spatial clusters as polygon models. Finally, spatio-temporal clusters are formed by identifying continuing relationships between spatial clusters in consecutive batches. The framework was successfully applied to New York City (NYC) taxi trips data. The experimental results show that all the algorithms can effectively discover interesting spatio-temporal patterns in taxi-pickup-location streams.

Recently, Twitter, one of the fastest-growing microblogging services, induced lots of research; one hot topic was event detection from tweets. Since geo-tagged tweets can be viewed as location streams with time tags and the content of tweets, we propose a novel two-stage system to detect and track events from Twitter by integrating an LDA-based approach with the density-contour based spatio-temporal clustering approach we introduced earlier. In the proposed system, events were identified as

topics in tweets using an LDA-based (Latent Dirichlet Allocation) topic discovery step. Next, each tweet was assigned an event label. After all locations were extracted from each event, the spatio-temporal approach was employed to obtain event clusters and track their temporal continuity. Through some case studies, we demonstrated the effectiveness of the proposed system. In summary, we aimed to acquire not only the semantic aspect of the events, but also the geographic distribution of the events and their continuity along time. Such information can be used to help individuals, corporations, or government organizations to stay informed of "what is happening now" and to acquire actionable knowledge.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Due to the advances in remote sensing and sensor networks, different types of spatio-temporal datasets become increasingly available. Revealing interesting spatio-temporal patterns from such datasets is very important, as it has broad applications, such as understanding climate change [65], identifying crime patterns [33], epidemics detection [48], flood risk analysis [29], geo-targeting, environment protection, and earthquake analysis [56]. Spatio-temporal clustering is particularly useful in analyzing large amounts of data since it allows domain experts to consider groups of objects rather than individual objects and to focus on a higher-level and more summarized representation of the data.

Consequently, spatio-temporal clustering has become a major research field of GIS-related knowledge discovery, which aims to detect groups of similar spatio-temporal entities. These clusters frequently identify trends, patterns related to geographical phenomena. Practically speaking, spatio-temporal clusters capture a

relationship between the spatial and temporal dimensions for a given phenomenon. Therefore, identifying spatio-temporal clusters may provide valuable insight beyond the determination of exclusively spatial or temporal clusters. In general, spatio-temporal clustering can reveal interesting distribution patterns that serve as valuable inputs for other data mining techniques, such as classification and association analysis [77].

Moreover, geospatial applications, such as location-based services (LBS) and Intelligent Transportation System (ITS) have been widely used these days. In general, geostreaming applications are growing both in quantity and scale due to recent advancements in sensing technology and the increased popularity of social media and smartphones [43]. As a result, there is an exponential growth in data generation and querying rates for these data, highlighting the importance of efficient techniques for geostreaming. When it comes to processing geo-tagged data streams, we face the following major stream processing challenges: querying, analysis and integration, scalability, extensibility, one-time access to data, volume, and real-time analysis.

1.1 Dissertation Overview

In order to identify spatio-temporal clusters, one major challenge that needs to be addressed is determining how spatial and temporal information are combined. Almost all existing approaches treat time and space in a parallel fashion. These approaches pass over the data several times and cannot deal with the case when the dataset is too large and the memory is limited, which raises a limitation on the scalability. For

example, ST-DBSCAN proposed by Briant [10] is an extension of DBSCAN, which introduces a temporal neighborhood radius in addition to the spatial neighborhood radius. It then looks for dense regions both temporally and spatially at the same time. Consequently, one of our research objectives is the development of novel spatio-temporal clustering frameworks that use time and space in serial, which is capable of clustering large-scale spatio-temporal data streams.

In this research, we propose a novel serial, three-phase density-contour based framework for spatio-temporal clustering of point cloud streams (e.g., location streams). It operates directly on density functions and applies contouring algorithms to extract spatial clusters from density contours. A family of novel distance functions are subsequently proposed to extract spatio-temporal clusters. While contouring algorithms are not very popular in machine learning and the data mining field, approaches that directly operate on density functions are rarely investigated as well. Moreover, based on the framework, we propose three spatio-temporal clustering algorithms: ST-DCONTOUR, ST-DPOLY, and ST-COPOT. They all rely on a three-phase clustering approach, which takes the point cloud stream as input and divides it into batches based on fixed-size time windows; next, a density estimation approach and contouring algorithms are employed to obtain spatial clusters as polygon models; finally, spatio-temporal clusters are formed by identifying continuing relationships between spatial clusters in consecutive batches. The framework was successfully applied to NYC taxi trips data [66]. The experimental results show that all the algorithms can effectively discover interesting spatio-temporal patterns in taxi pick-up location streams. This allows us to have a look into how citizens commute through taxis and

helps taxicab companies to allocate resources using the identified demanding change pattern.

To further utilize the serial, density-contour based framework, we propose a novel system to detect and track events in Twitter streams. Nowadays, Twitter has become one of the fastest-growing microblogging services, with around 328 million monthly active Twitter users producing over 5 million tweets per day in 2017 [1]. People have been using it to report everything from facts and experiences in their lives to the latest local and global news and events. Monitoring and analyzing this rich and continuously user-generated content can reveal unprecedentedly valuable knowledge that can be used to help individuals, corporations, or government organizations to stay informed of "what is happening now" and to acquire actionable knowledge. For example, people are interested in getting updates, facts, opinions, or advice on news and events [42, 82]; companies are increasingly using Twitter to recommend products, brands, and services, to maintain reputations and to improve decision-making among other things [41, 55]. Moreover, the government uses information from Twitter for disaster and emergency management [67, 54].

Compared to traditional media, social media platforms are a valuable knowledge source for event detection as well. First, as they are online services, real-world happenings can be revealed in a quicker fashion. Second, a more complete and detailed picture of a real-world event can be obtained in large-scale from different angles on social media as well [23]. These advantages have triggered a significant amount of research in event detection from social media. Moreover, detecting and

[1] https://www.omnicoreagency.com/twitter-statistics/

tracking events from tweets allows us to analyze and summarize the most important aspects of an event, such as peoples opinion about a particular event, when and where the event is taking place and how long it lasts. In summary, applying the event detection techniques to social media provides actionable knowledge, thereby enabling better, data-driven decision-making.

When it comes to the geo-tagged tweets specifically, they represent a spatio-temporal signal (geolocation and timestamp of the tweet) with a semantic information layer (content of tweet) [64]. According to the survey paper [64] published in 2015, among 92 papers concerning twitter event detection, only 33 percent of papers use all information layers, including message content, geotag and the timestamp. Therefore, the study of event detection by using both spatio-temporal information and semantic analysis of content from location-based social networks represents a promising but, still, underexplored field.

In this research, we propose a novel two-stage system to detect and track events from tweets by integrating natural language processing techniques and machine learning techniques; specifically, it integrates an LDA-based approach [58] and an efficient density-contour based spatio-temporal clustering approach. The density-contour based approach extends a spatio-temporal clustering framework called ST-COPOT [79], which is capable of processing very large data streams in approximately linear time. The major modifications include: supporting "relative" density, including edge correction and introducing distance functions to establish the spatial and temporal continuity of events. In the proposed system, we first divide the geo-tagged tweet stream into temporal time windows. Next, events are identified as topics in tweets

using an LDA-based topic discovery step. Subsequently, each tweet is assigned an event label and after locations for each event are extracted in each batch, a density-contour based spatio-temporal approach is employed to identify spatio-temporal clusters of each event. Moreover, event continuity is established through two perspectives, topic continuity is established by calculating KL-divergence (Kullback-Leibler) [46] between topics and spatio-temporal continuity is established by a family of newly formulated distance functions that assess the similarity of spatial clusters obtained for different density thresholds.

Moreover, the density-contour based approach considers two types of densities: absolute density and relative density, used to identify regions with a high percentage of tweets or a high density of tweets related to a particular event. By tracking events from tweets, our proposed system can locate the target users that are interested in the particular event and infer the most affected region by an event. For example, we can locate a large number of sports fans in order to push advertisements to a target region through social media, or we can identify regions where a high percentage of the population is affected by a snowstorm.

1.2 Dissertation Contribution

Key contributions of the proposed research:

- We propose a serial, density-contour based framework for spatio-temporal clustering.

1. Our approach treats time and space in a serial fashion, which creates spatial clusters first and then spatio-temporal clusters are constructed as continuing spatial clusters in consecutive batches.
2. To the best of our knowledge, our proposed approach is the first approach that directly operates on density functions and uses density-contour and contour analysis to create spatio-temporal clusters from spatio-temporal data streams.
3. We propose a novel data structure called *contour polygon tree* as our spatial cluster model.
4. We propose a family of novel distance functions that operate on contour polygon trees to establish continuity between spatial clusters in consecutive batches; spatio-temporal clusters are created at different levels of granularity, e.g., continuing polygons, continuing trees, and continuing forests.
5. The proposed framework is time efficient and can achieve approximately linear time complexity for large-scale point cloud stream.
6. We propose three spatio-temporal clustering algorithms based on the serial, density-contour based framework.
7. We evaluate the proposed framework and these algorithms in a challenging real-world case study involving NYC taxi trips data. The experimental results show that all of them can effectively discover interesting spatio-temporal patterns in taxi pick-up location streams.

- We propose a novel two-stage system by integrating an LDA-based approach and an efficient density-contour based approach for event detection and tracking in Twitter.

    1. In contrast to other spatial-temporal clustering approaches, our approach operates directly on density functions and uses contouring algorithms to identify event clusters.
    2. Our approach employs "relative" and "absolute" density to obtain spatio-temporal clusters.
    3. We propose a drill down operation—that operates on the identified event clusters that have been obtained by our system—which summarizes the spatial variation of tweet locations that are related to an event at a finer granularity. This is accomplished by rerunning the spatial clustering algorithms with a different set of parameters, such as different density thresholds, kernel bandwidth, and grid cell sizes.
    4. We demonstrate our approach using real-world data collected from Twitter. The experimental results show that the proposed system can effectively detect and track events from tweets.
    5. Moreover, we discuss practical experiences in using LDA for topic discovery in Twitter.
    6. The proposed system is successfully applied in an emotion mapping approach that operates on tweets.

1.3 Dissertation Organization

The rest of the dissertation is organized as follows. Chapter 2 introduces the related work about spatio-temporal clustering and presents the serial, three-phase density-contour based clustering framework we propose. In Chapter 3, we discuss three proposed serial algorithms that employ the three-phase density-contour based framework. Chapter 4 discusses the comparison of "serial" and "parallel" spatio-temporal clustering approaches. Chapter 5 introduces the two-stage system we propose for event detection in Twitter. Chapter 6 concludes the dissertation.

Chapter 2

Serial, Density-contour based Framework for Spatio-temporal Clustering

2.1 Inputs and Overview of the Framework

We propose multiple spatio-temporal clustering algorithms, for point cloud stream specifically, the inputs are all the same.

One input of our framework is a *spatio-temporal point cloud stream*; for example, taxi pick-up location cloud streams that are described by the location, e.g., using longitude, latitude, and the pick-up/drop-off time. Moreover, we assume that data originated from a *data collection area*, e.g., the New York Metropolitan Area and we

assume that this data collection area is given in the form of a rectangle or a polygon.

Moreover, prior to applying the framework, we subdivided the collected data into temporal batches associated with a particular time window. Our approach assumes that spatio-temporal point clouds are processed in batches and that each batch is collected at a fixed interval, e.g., every hour. For many applications, it is challenging to determine how to partition streams into batches. In this framework, we use a simple approach that employs equal-time intervals as batch sizes and the goal is to obtain spatial clusters and find a continuing relationship between spatial clusters in consecutive batches.

Our proposed serial, density-contour based framework mainly consists of the following three phases:

1. A spatial density function is obtained for spatial point cloud collected in each batch.

2. Spatial clusters are identified for each batch as polygons that are created from density-contour lines of the spatial density function.

3. Continuing relationships between spatial clusters are identified and spatio-temporal clusters are constructed as continuing spatial clusters in consecutive batches.

2.2 Phase 1: Obtain a Spatial Density Function

For phase 1, we use non-parametric kernel density estimation (KDE) [24] to obtain a 2-dimensional spatial density function, f. For a bivariate random sample $X_1, X_2, \ldots, X_n$ drawn from an unknown density, f, the kernel density estimator is defined as follows:

$$\widehat{f}(x; H) = \frac{1}{n} \sum_{i=1}^{n} K_H(x - X_i), \tag{2.1}$$

where $x = (x_1, x_2)^T$, $X_i = (X_{i1}, X_{i2})^T$ $(i = 1, 2, \ldots, n)$, $K((x_1, x_2); \sigma) = \frac{1}{2\sigma\pi} e^{\frac{-(x_1^2 + x_2^2)}{2\sigma^2}}$ is the Gaussian kernel which is a symmetric, non-negative probability density function; H is bandwidth matrix which is symmetric and positive-definite, $K_H(x) = |H|^{-1/2} K H^{-1/2} x$. Our implementation uses the KernSmooth package [69] in R to estimate the spatial density distribution for given spatial points in each batch.

2.3 Phase 2: Spatial Cluster Extraction

For the second phase of our framework, the goal is to identify dense spatial regions in the data collection area as spatial clusters using the spatial density functions that have been created for each batch in phase 1. A spatial cluster is defined as a region which is enclosed by a polygon and whose probability density of data points is above a given threshold. Our approach uses the contouring algorithm—marching square and post-processing to get such polygons.

The process of extracting spatial cluster from a spatial density function mainly consists of the following six steps:

1. The data collection area is gridded using a two-dimensional grid.

2. Probability density values for all grid intersection points are calculated using the spatial density function and a density matrix is obtained. A table T is created to store locations of all grid intersection points and corresponding density matrix.

3. Table T, along with a pair of density threshold θ_1, $\hat{\theta}_1$ are passed to marching square, which returns two sets of contour lines.

4. Open contour lines are closed.

5. The obtained contour lines are classified into holes and spatial clusters.

6. The step 3 to step 5 are iterated for density threshold pairs: θ_2,$\hat{\theta}_2$, θ_3, $\hat{\theta}_3$,, ..., θ_N,$\hat{\theta}_N$, respectively.

In step 6, we use multiple density thresholds to extract polygons that are embedded into each other for each batch [1]. As a compact representation of the clustering results, we defined a novel data structure called *Contour Polygon Tree (CPT)* whose nodes store polygons. The contour polygon tree satisfies the containment relationship between a node and its children and polygons of children of the same parent in the tree have to be non-overlapping, as is specified below.

Definition 1. *Let T be a contour polygon tree, child(n) denotes the set of successor nodes of n in the tree, and n.polygon be the polygon associated with n:*

[1] Using solely one density threshold, sometimes we might obtain a poor clustering result, and more importantly, it does not allow one to view a dataset at different density granularities.

$$Contour\ Polygon\ Tree(T) \Leftrightarrow$$

$$\forall n \in T\ (not(leaf(n))) \Rightarrow \forall \hat{n}(child(n) \Rightarrow (\hat{n}.polygon \subseteq n.polygon)) \wedge$$

$$\forall \hat{n} \in T\ \forall \hat{\hat{n}} \in T\ \exists n(\hat{n} \in child(n) \wedge \hat{\hat{n}} \in child(n) \Rightarrow \hat{n}.polygon \cap \hat{\hat{n}}.polygon = \varnothing)$$

where, "$\subseteq$" represents the polygon containment relationship: $p \subseteq \hat{p} \Longleftrightarrow p \cap \hat{p} = p$, with "$\cap$" represents the polygon intersection operator.

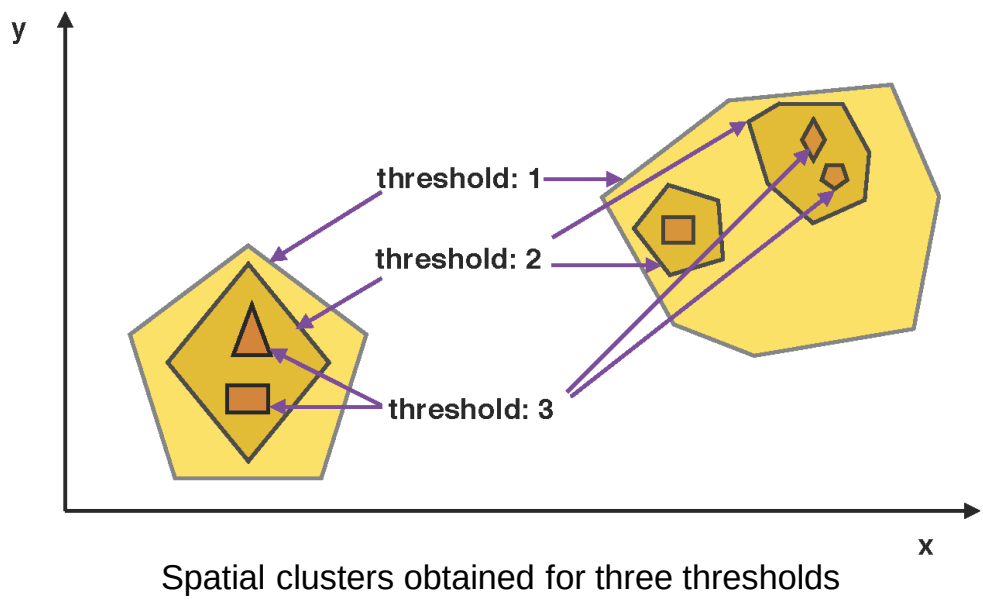


Figure 2.1: Example and visualization of contour polygon trees

Contour polygon tree represents a hierarchical structure. The root of each tree stores a polygon that has been generated using the lowest density threshold and the polygons of lower levels of the tree are always contained in the polygon of the

higher levels. In cases when get more than one contour polygon tree for each batch, we call the clustering result a *contour polygon forest*. Figure 2.1 gives an example of two contour polygon trees that have been generated and depicts their associated polygons. A similar data structure called *density contour tree* was first introduced and briefly discussed in [35]. However, it has not been further investigated in other methodologies or applications.

2.4 Phase 3: Spatio-temporal Cluster Extraction

We need to establish the temporal continuity to extract spatio-temporal clusters. In order to identify continuing relationship from two sets of contour polygon trees obtained for two consecutive batches, we propose a set of novel distance functions for contour polygons, contour polygon trees, and contour polygon forests. Using those density functions, spatio-temporal clusters are defined and obtained at different granularities continuing contour polygons, continuing contour polygon trees, and continuing contour polygon forests.

There exist many approaches calculating the distance between two density functions [15]. However, these approaches track the global change, that is, they analyze change over the whole data collection area. The approach that we present in this section emphasizes how the spatial clusters change over time, focusing on how dense areas in space change over time; that is, if our task is crime analysis, we are interested in where the main crime hotspots are and how they change over time. To support this kind of analysis, we introduce novel distance functions in this section.

Before we define those distance functions, let us introduce the following notations: let p and $\overline{p}$ be contour polygons that correspond to the same density threshold, t and $\bar{t}$ be contour polygon trees, S and $\overline{S}$ be sets of contour polygons at the same level for polygon trees t and $\bar{t}$, respectively, F and $\overline{F}$ be contour polygon forests, and N is the number of density thresholds. Moreover, we assume that p and $\overline{p}$, S and $\overline{S}$, t and $\bar{t}$, F and $\overline{F}$ are created for two consecutive batches, respectively.

We first define a distance function for polygons to identify continuing relationships between contour polygons as follows:

$$d_P(p, \overline{p}) = 1 - \frac{area(p \cap \overline{p})}{area(p \cup \overline{p})}, \tag{2.2}$$

where $area(p \cap \overline{p})$ is the intersection area of p and $\overline{p}$, and $area(p \cup \overline{p})$ is the union area of p and $\overline{p}$.

Since our clustering result for each batch consists of sets of trees, a distance function to assess the similarity of trees from two consecutive batches is needed. Since the root of a tree, always corresponds to the lowest density threshold, a näive distance function could be defined as follows:

$$d_T^*(t, \bar{t}) = d(t.root.polygon, \bar{t}.root.polygon), \tag{2.3}$$

where, $t.root.polygon$ is the polygon at the root level (level 1) of tree t and $\bar{t}.root.polygon$ is the polygon at the root level (level 1) of tree $\bar{t}$.

We are also interested in obtaining a distance function for comparing sets of polygons at levels 2, 3,..., N of the trees. To accomplish this, firstly we define a

"forward" distance function for sets of polygons as follows:

$$d_S^*(S,\overline{S}) = \begin{cases} 0 & \text{if } S = \varnothing \text{ and } \overline{S} = \varnothing \\ 1 & \text{if } S = \varnothing \text{ or } \overline{S} = \varnothing \\ \frac{\sum_{p\in S}(min_{\overline{p}\in\overline{S}}(d_p(p,\overline{p})))}{|S|} & \text{otherwise} \end{cases}, \tag{2.4}$$

where $|S|$ is the total number of polygons in S, and the distance $d(p,\overline{p})$ is calculated using Equation 2.2. We try to sum up the closest distance from polygons in $\overline{S}$ to the polygons in S by iterating over all the polygons in S. If there exists a polygon that doesn't have any overlap with the polygons in $\overline{S}$, this distance will be 1. Finally, in order to normalize total distance, we divide the total distance by the number of polygons in S.

We also propose an area-weighted forward distance function as alternative for Equation 2.4 by putting more emphasis on agreement with respect to larger polygons, which is defined as follows:

$$d_S^*(S,\overline{S}) = \begin{cases} 0 & \overline{S} = \varnothing \\ 1 & \text{if } S = \varnothing \text{ or } \overline{S} = \varnothing \\ \frac{\sum_{p\in S}(area(p)\cdot min_{\overline{p}\in\overline{S}}(d_p(p,\overline{p})))}{\sum_{p\in S} area(p)} & \text{otherwise} \end{cases}. \tag{2.5}$$

From Equation 2.4 and Equation 2.5, we observe that sometimes $d_S^*(S,\overline{S}) \neq d_S^*(\overline{S},S)$, as S and $\overline{S}$ might consist of a different number of polygons; consequently, the last distance function we introduced is not symmetric. To deal with this problem, we use the same formula to calculate the backward distance: $d^*(\overline{S},S)$ which sums up the closest distance from polygons in S to polygons in $\overline{S}$ iterating over $\overline{S}$. By

averaging the forward and backward distances to obtain the symmetric distance for sets of polygons, which is calculated as follows:

$$d_S(S, \overline{S}) = d_S(\overline{S}, S) = \frac{d_S^*(S, \overline{S}) + d_S^*(\overline{S}, S)}{2}. \tag{2.6}$$

Now we have a distance function to assess the similarity of polygons at a specific tree level, next, we propose a distance function for pairs of contour polygon trees that compares all the levels of each tree, which is defined as follows:

$$d_T(t, \bar{t}) = \frac{d_T^*(t, \bar{t}) + \sum_{j=2}^{N} \rho^{j-1} \cdot d_S(level(j, t), level(j, \bar{t}))}{1 + \sum_{j=2}^{N} \rho^{j-1}}, \tag{2.7}$$

where N is the number of levels of t and $\bar{t}$, respectively; and $level(i, t)$ and $level(i, \bar{t})$ are the sets of polygons at level i for t and $\bar{t}$, respectively. Moreover $\rho \in (0, 1]$ is a parameter called discount factor [2].

Equation 2.7 puts more importance to polygons closer to the root when assessing the similarity between two CPTs; however, the above distance function $d_T(t, \bar{t})$ can be generalized to put more importance to a particular level of a contour polygon tree, which is defined as follows:

$$d_{T,s}(t, \bar{t}) = \frac{\sum_{j=1}^{N} (\rho^{|j-s|} \cdot d_S(level(j, t), level(j, \bar{t})))}{\sum_{j=1}^{N} \rho^{|j-s|}}, \tag{2.8}$$

where N is the number of levels of t and $\bar{t}$, respectively; s is the focus granularity level with $1 \leqq s \leqq N$; $level(i, t)$ and $level(i, \bar{t})$ are the sets of polygons at level i for t and $\bar{t}$, respectively. Moreover, $\rho \in (0, 1]$ is the discount factor. For example, if $N = 5$ and $s = 3$, level 3 agreement is weighted by 1, level 2 and 4 agreements are

[2]Disagreement counts less the deeper we go down the tree.

weighted by ρ and level 1 and 5 agreements are weighted by ρ^2 when computing the distance between t and $\bar{t}$, moreover, the following holds $d_T(t,\bar{t}) = d_{T,1}(t,\bar{t})$.

Furthermore, using the same approach of defining $d^*_S(S,\overline{S})$, the forward distance function for two contour polygon forests is defined as follows:

$$d^*_F(F,\overline{F}) = \begin{cases} 0 & \text{if } F = \varnothing \text{ and } \overline{F} = \varnothing \\ 1 & \text{if } F = \varnothing \text{ or } \overline{F} = \varnothing \\ \frac{\sum_{t\in F}(min_{\bar{t}\in\overline{F}}(d_T(t,\bar{t})))}{|F|} & \text{otherwise.} \end{cases}, \tag{2.9}$$

where $|F|$ is the total number of trees in F, and the distance $d_T(t,\bar{t})$ is calculated using Equations 2.7 or 2.8. Basically, we sum up the closest distance from the trees in $\overline{F}$ to the trees in F by iterating over all the trees in F. If there exists a tree that doesn't have any overlap with the trees in $\overline{F}$, the distance will be 1. Furthermore, in order to normalize the total distance, we divide the total distance by the number of trees in F.

We also propose an area-weighted forward forest distance function using the root area of each tree as the weight:

$$d^*_{F,s}(F,\overline{F}) = \begin{cases} 0 & \text{if } F = \varnothing \text{ and } \overline{F} = \varnothing \\ 1 & \text{if } F = \varnothing \text{ or } \overline{F} = \varnothing \\ \frac{\sum_{t\in F}(min_{t\in F}(area(t)*d_{T,s}(t,\bar{t})))}{\sum_{t\in F} area(t)} & \text{otherwise.} \end{cases}. \tag{2.10}$$

Also, from the definition above, we know that sometimes $d^*_F(F,\overline{F}) \neq d^*_F(\overline{F},F)$, as F and $\overline{F}$ might consist of a different number of trees, and we reuse the earlier

approach to make the forest distance function symmetric, which is calculated as follows:

$$d_F(F, \overline{F}) = d_F(\overline{F}, F) = \frac{d_F^*(F, \overline{F}) + d_F^*(\overline{F}, F)}{2}. \tag{2.11}$$

In summary, we defined a family of distance functions for pairs of polygons, pairs of sets of polygons, pairs of contour polygon trees and pairs of contour polygon forests. Next, we use the distance functions to propose computational methods that identify continuing relationships between consecutive batches at different granularities:

- If the distance between two contour polygons from two consecutive batches is less than a certain threshold, γ_1, we conclude that the contour polygon doesn't change significantly over two consecutive batches, and create a 'continuing' relationship between the two polygons, if the following condition holds:

$$Continuing(p, \overline{p}) \Leftrightarrow d_P(p, \overline{p}) < \gamma_1. \tag{2.12}$$

- If the distance between two contour polygon trees from two consecutive batches is less than a certain threshold, γ_2, we create a 'continuing' relationship between the two contour polygon trees, if the following condition holds:

$$Continuing(t, \overline{t}) \Leftrightarrow d_T(t, \overline{t}) < \gamma_2. \tag{2.13}$$

- Similarly, if the distance between two contour polygon forests from two consecutive batches is less than a certain threshold, γ_3, we infer a 'continuing' relationship between the two forests, if the following condition holds:

$$Continuing(F, \overline{F}) \Leftrightarrow d_F(F, \overline{F}) < \gamma_3. \tag{2.14}$$

2.5 Time and Space Complexities

Table 2.1: Time and space complexities

Phases	Time Complexity	Space Complexity
Phase 1	$O(m^2 \times n)$	$O(n + m^2)$
Phase 2	$O(m^2)$	$O(m^2 + e^2)$
Phase 3	$O(e^2)$	$O(e^2)$

Table 2.1 gives the time and space complexities of the framework we propose. We assume that the grid size we use is $m \times m$, n is the total number of points, e is the average number of edges of a spatial cluster. Since e is usually smaller than m and it is a serial approach, the overall time complexity of our framework would be $O(m^2 \times n)$; in cases where the number of data points is much larger than the number of grid cells ($n >> m^2$), the time complexity becomes $O(n)$, which means that the proposed framework is capable of processing very large data streams in approximately linear time.

2.6 Related Work

Most existing density-based clustering approaches extract clusters with one single layer of boundary, e.g., DBSCAN [26]. Approaches that obtain hierarchical clustering result also have been investigated; for example, OPTICS [5] generalizes DBSCAN to extract hierarchical clustering structure. Our framework is different from those approaches, not only in the way it creates the hierarchical clustering structure, but also in its application. OPTICS, based on DBSCAN, has the time complexity of $O(n^2)$

while our density-contour based approach achieves much lower time complexity—approximately linear complexity $O(n)$, which makes our approach more suitable for processing large-scale stream data. Hierarchical clustering result can offer additional insights into the distribution of the data, e.g., locating "hotter-spots" inside a hotspot. Approaches that use density functions are also investigated, e.g., DENCLUE [37] proposed by Hinneburg first identify density attractors as local maxima of the overall density function, then clusters are formed by associating data objects with density attractors using hill climbing, while our approach applies a contouring algorithm directly to the density function to extract spatial clusters.

Spatio-temporal clustering and hotspot discovery techniques for point objects also have been well-studied in the literature. Kulldorff et al. [47] introduced a spatial scan statistic for the detection of spatio-temporal cylinders where the point objects occur consistently for a significant period of time. Iyengar et al. [40] extended the basic scan statistics using the flexible square pyramid shape to detect clusters with restrictive shapes, and the proposed framework can model growth and shifts in location over time. Wang et al. [70] proposed a spatio-temporal clustering algorithm ST-GRID, which maps the spatial and temporal dimensions into multidimensional cells and then extracts and merges spatio-temporal dense regions to obtain a final cluster. Birant et al. [10] proposed ST-DBSCAN as an extension of DBSCAN for spatio-temporal clustering by introducing a second parameter of temporal neighborhood radius in addition to the spatial neighborhood radius. Wang et al. proposed a spatio-temporal clustering approach, ST-SEP-SNN [71], which combines spatial and

temporal distances into a joint spatio-temporal distance function, and then generalizes the SNN clustering algorithm to operate on the obtained joint distance function.

However, most existing spatio-temporal clustering algorithms mentioned above are not suitable to deal with large data streams. For example, both ST-DBSCAN and ST-SEP-SNN pass over the data several times and assume that the data fit into main memory.

Clustering from data streams has also been well-studied. Farnstrom et al. [27] proposed a single-pass partitioning algorithm known as an extension of k-means. The main idea is to use a buffer where the dataset is kept in a compressed way, and the streams are processed in blocks while all available space on the buffer is filled with points from the streams. Aggarwal et al. [2] proposed a CluStream system as an extension of BIRCH [76], which can generate approximate clusters for any user-defined time granularity. This micro-clustering approach divides the clustering process into two phases, where the first phase is online and summarizes the data stream in local models (micro-clusters) and the second phase generates a global cluster model from the micro-clusters. Barbará et al. [7] proposed a Fractal Clustering system, which is a grid-based approach. It processes data points in batches and assigns the data points to the group in which that assignment produces a less fractal impact. Chen et al. [17] proposed a framework for clustering stream data that uses an online component to map input data into a grid and an offline component to compute the grid density and cluster the grids. Wan et al. [68] also proposed an online-offline approach, which is able to detect arbitrarily shaped, evolving clusters with high quality. Hadjieleftheriou [34] introduced a grid-based framework for answering density-based

queries in moving object databases based on the notion of density in space and time.

Overall, density-based stream clustering algorithms are categorized into two broad groups, density micro-clustering algorithms and density grid-based clustering algorithms [4]. In density micro-clustering algorithms, micro-clusters extract synopsis information about stream data, then clustering is performed on the summary information. Some example algorithms in this category are DenStream [14], HDenStream [51], SOStream [39] and PreDeConStream [36]. In density grid-based clustering algorithms, the data space is divided into grids, then the clusters are formed based on the density of grids. Some example algorithms in this category are D-Stream [17], PKS-Stream [59], DENGRIS-Stream [3], and ExCC [9].

However, our proposed approach is neither strictly a micro-clustering nor strictly a grid-based clustering approach. It uses grids only for the contouring algorithm. To the best of our knowledge, our proposed approach is the first approach that uses density-contour and contour analysis to create spatio-temporal clusters from spatio-temporal data streams. Moreover, our spatio-temporal clustering approach can identify spatio-temporal clusters at different levels of granularity.

Other existing stream clustering algorithms use incremental approaches that receive and process data elements one at a time. For example, Zhang et al. [76] proposed a system called BIRCH, which compresses data and builds a hierarchical data structure to incrementally cluster the incoming points using available memory and minimizing the amount of I/O required. Song et al. [62] introduced a probability-density-based data stream clustering approach, which incrementally updates the density estimate taking only the newly arrived data and the previously

estimated density. Unlike these incremental approaches, our approach is a batch clustering algorithm. Instead of modifying an existing summary based on newly coming data, we combine and connect the summaries without changing them. Moreover, incremental clustering methods are strictly weaker than batch algorithms in their ability to detect clustering structure, while in the batch model, a good cluster structure is easier to detect [1].

2.7 Summary

In this chapter, we present a serial, density-contour based framework for spatio-temporal clustering of a point cloud stream, which first employs a non-parametric density estimation approach to obtain spatial cluster as regions enclosed by polygons generated from contour lines whose density corresponds to certain thresholds. To support these activities, our approach employs a data structure called *contour polygon tree* as a compact representation of clustering result for each batch. Using a family of novel distance functions, our approach forms spatio-temporal clusters by identifying continuing relationships between temporally consecutive spatial clusters. Therefore, spatio-temporal clusters are defined and obtained at different granularities: continuing contour polygons, continuing contour polygon trees, and continuing contour polygon forests. The proposed framework meets the one-time access requirement for streaming data processing, as the data in each batch are only read once. The framework is capable of processing very large data streams in approximately linear time. To the best of our knowledge, our approach is the first approach that

uses contouring algorithms and contour analysis to obtain spatio-temporal clusters.

Chapter 3

Serial, Density-contour Based Spatio-temporal Clustering Algorithms

3.1 Dataset for Evaluation

To evaluate all the proposed algorithms, we use the TLC Trip Record Data [66], which was collected by technology providers authorized under the Taxicab and Livery Passenger Enhancement Programs (TPEP/LPEP), containing data for over 1.1 billion taxi trips from January 2009 through June 2016. Each trip record contains precise location coordinates for where the trip started and ended, timestamps for when the trip started and ended, and a few other variables.

3.2 ST-DCONTOUR

In [78], we proposed our first serial, density-contour based spatio-temporal clustering algorithm called ST-DCONTOUR, which employs a model-based clustering methodology in phase 1 of the proposed framework.

3.2.1 Phase 1: Spatial Density Estimation

In phase 1, we use mixtures of bivariate Gaussians as a spatial density model, whose density function is defined as follows:

$$N(x|\mu, \Sigma) = \frac{1}{(2\pi)^2(|\Sigma|)^{1/2}} exp^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}, \tag{3.1}$$

where μ is a two-dimensional mean vector, Σ is a covariance matrix, and $|\Sigma|$ is the determinant of Σ.

Spatial density functions are defined as k-component Gaussians mixture models:

$$p(x|\lambda) = \Sigma_k^K w_k * N(x|\mu_k, \Sigma_k), \tag{3.2}$$

where, w_k is the weight of each component, μ_k is the mean of k-th Gaussian, Σ_k is the covariance matrix of k-th Gaussian, x is the data point under consideration, $N(x|\mu_k, \Sigma_k)$ is the density of k-th Gaussian and K is the total number of Gaussian components.

In general, the Gaussian mixture model is parameterized by the mean vectors, covariance matrices, and component mixture weights:

$$\lambda = \{w_i, \mu_i, \Sigma_i\}, i = 1, 2, 3, \ldots, K. \tag{3.3}$$

In order to obtain the final spatial density function, the EM algorithm [22] is used to create GMMs for different k-values and model-types, maximizing the log-likelihood which is given by:

$$lnL(w, \mu, \Sigma; x_1, \ldots, x_n) = \Sigma_{i=1}^{N} ln\{w_k N(x_i|\mu_k, \Sigma_k)\} \tag{3.4}$$

3.2.2 Phase 2 and Phase 3 of ST-DCONTOUR

For phase 2 of ST-DCONTOUR, we use solely one density threshold in the contouring algorithm to identify spatial clusters.

In phase 3, to extract the relationship between temporarily consecutive spatial clusters, we use the distance function defined in 2.2.

3.2.3 Experimental Results

We created spatio-temporal clusters using yellow taxi pick-up locations collected in one-hour intervals as batches. We analyzed three consecutive hours from 11 pm on January 6th to 2 am on January 7th (2016). Figure 3.1 shows two spatio-temporal clusters that ST-DCONTOUR created: $SC1$ continues for three consecutive batches and $SC2$ appears at batch 2 and continues for two batches. According to the spatio-temporal clusters we obtained, the midtown area of New York City is busy at midnight (11 pm-2 am) as far as taxi pick-ups are concerned, which means many people hang around that area during that time, particularly nearby the Time Square area. A newly appearing cluster, such as $SC2$ located in the time square at batch 2, shows

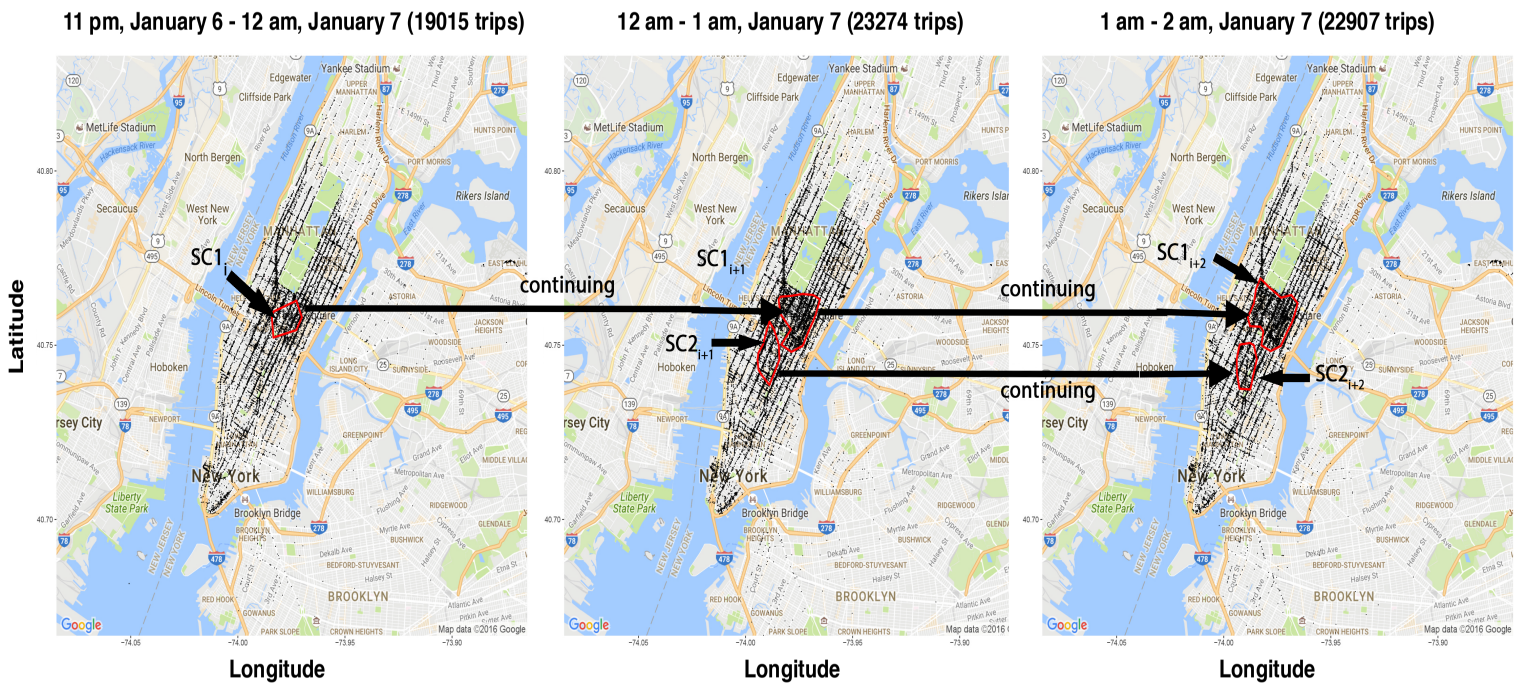


Figure 3.1: Depicted is a spatio-temporal clustering of taxi pick-up location streams for three consecutive hours. The red contours are spatial clusters we obtained for each batch. The arrows denote the continuing relationship. The black points are taxi pick-up locations.

that people start leaving that area at midnight but do not require taxi cab services at an earlier time as there isn't a spatial cluster for the 11 pm to midnight batch—it is too early for New Yorkers to go home yet.

3.3 ST-DPOLY

Instead of using the parametric Gaussian Mixture Model as the density estimation model, we decide to use non-parametric kernel density estimation (presented in Section 2.3), which is faster and improves the performance significantly. We call the new approach ST-DPOLY [81].

Using ST-DPOLY as baseline "serial" approach, we also give a thorough comparison between a serial spatio-temporal clustering approach and a parallel spatio-temporal clustering approach—ST-SNN [71]. More details will be covered in Chapter 4.

3.4 ST-COPOT

We extended ST-DPOLY to support multiple density thresholds. The obtained spatial clusters have the hierarchical structure that allows us to look into clusters at different density granularities. We name the data structure *Contour Polygon Tree*, in terms of how we obtain contour polygon trees as spatial cluster models, see Section 2.2. We propose a family of distance functions to obtain spatio-temporal clusters at different granularities, for more details, see Section 2.3. The new approach is named

ST-COPOT [79].

3.4.1 Experiment and Analysis

3.4.1.1 Demonstration of ST-COPOT

We reused the TLC Trip Record Data [66] to evaluate ST-COPOT. Specifically, we used yellow taxi pick-up locations collected on January 8th, 2014 in our experiments. The data collection area is the Manhattan metropolitan area and the total number of taxi pick-ups is over 430,000. The clustering result is presented in Figure 3.2. We picked a two-hour interval as batch size and used 0.45 as the density thresholds to extract all those continuing relationships. Using ST-COPOT can easily track the evolution of clusters over time by looking into the continuing relationship at different granularities, for example:

- There is a region centered around Time Square in the late night and before dawn, which shows that Time Square is where many people gather around during this time window.

- Early in the morning, in east of Midtown area, there are two sub-regions with high density, one is southwest of Time Square and is close to several train and bus terminals. The other sub-region is further southwest, which is close to the 34 Street Penn Station. We infer that, early in the morning, after New Yorkers get off trains, many of them go in search of taxi rides. Similarly, in the 6-8 am and 8-10 am batch, a contour polygon tree centering around the Grand Central

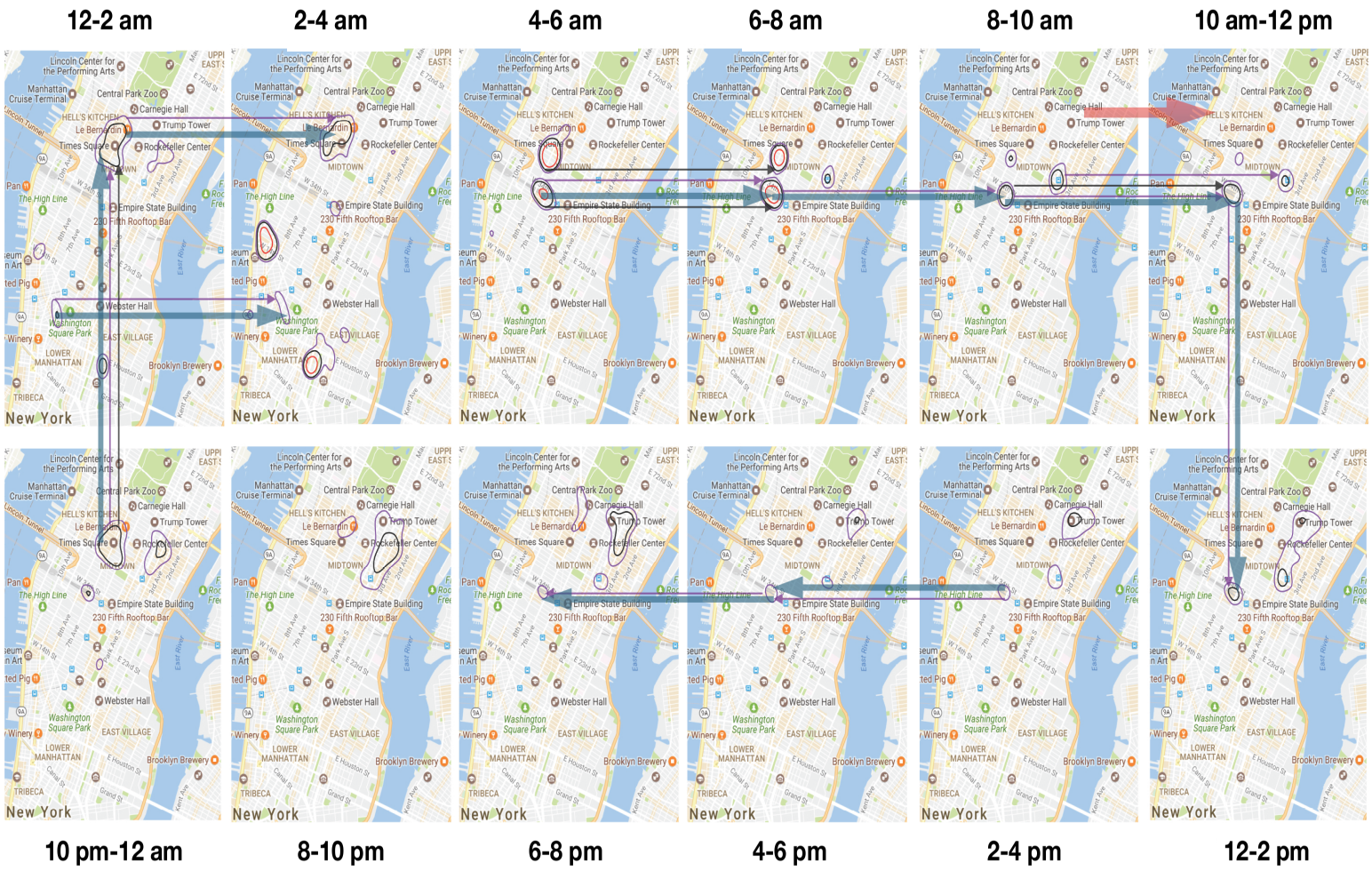


Figure 3.2: Depicted is a spatio-temporal clustering of taxi pick-up location streams for 24 consecutive hours. The contours are spatial clusters we obtained for each batch and different colors correspond to different thresholds (purple: 6.8, black: 7.0, red: 7.4). The arrows denote the continuing relationship and different colors correspond to different cases (purple: continuing polygons for threshold 6.8, black: continuing polygons for threshold 7.0, blue: continuing trees, red: continuing forests).

Terminal exhibits a similar taxi pick-up pattern.

- We observe between 2 pm and 8 pm in the afternoon, there are a lot of people who look for taxis at W 34th St. In the following 2 hours (8-10 pm), the contour polygon tree disappears, indicating that the demand for taxis decreases. Such information can help a taxi company to allocate taxis.

- Since all those spatial clusters are dense regions in terms of taxi pick-ups, it means it is harder to get a taxi in those regions during the time window when they are identified as spatial clusters. People who take taxis can better plan their activities beforehand using such information, which also shows the potential practical value of ST-COPOT for people who take taxis, especially commuters.

- Such results can also help taxi drivers to locate customers as well, as those regions are dense pick-up regions within a specific time window.

Through changing the batch size, we can identify hourly, daily, and weekly patterns as well.

3.4.2 Quality of Clustering Results of ST-COPOT

In order to evaluate the quality of the clustering results of ST-COPOT, we applied a density-based clustering validation method proposed by Moulavi [53] to the clusters obtained. Using this method, we can obtain a relative validity index for each cluster we created, the range of validity index is between -1 and 1. One flaw of the validation

method is that it does not consider the case that only a single cluster is obtained, so in this case, we ignore the validity index.

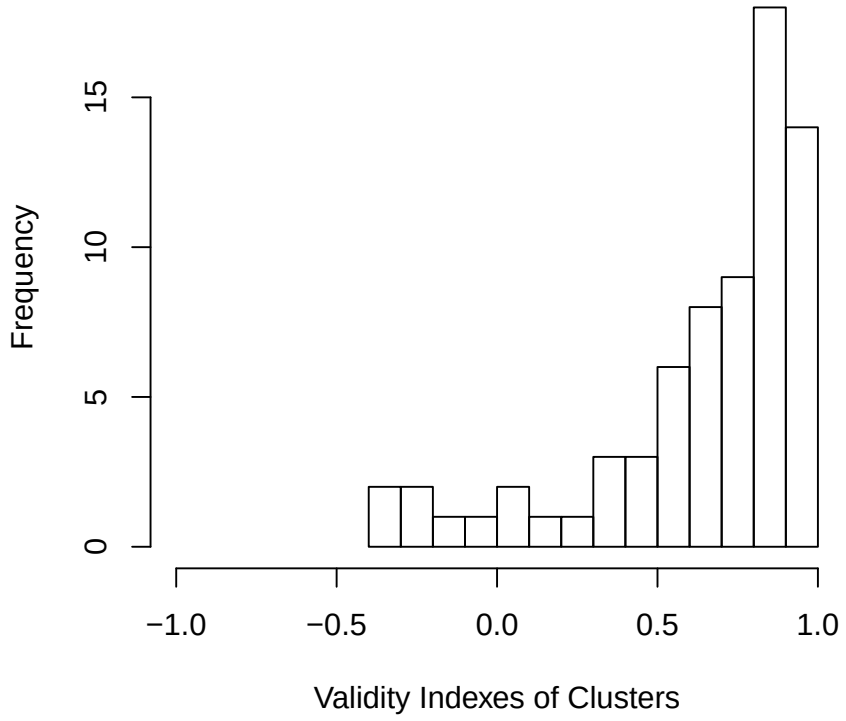
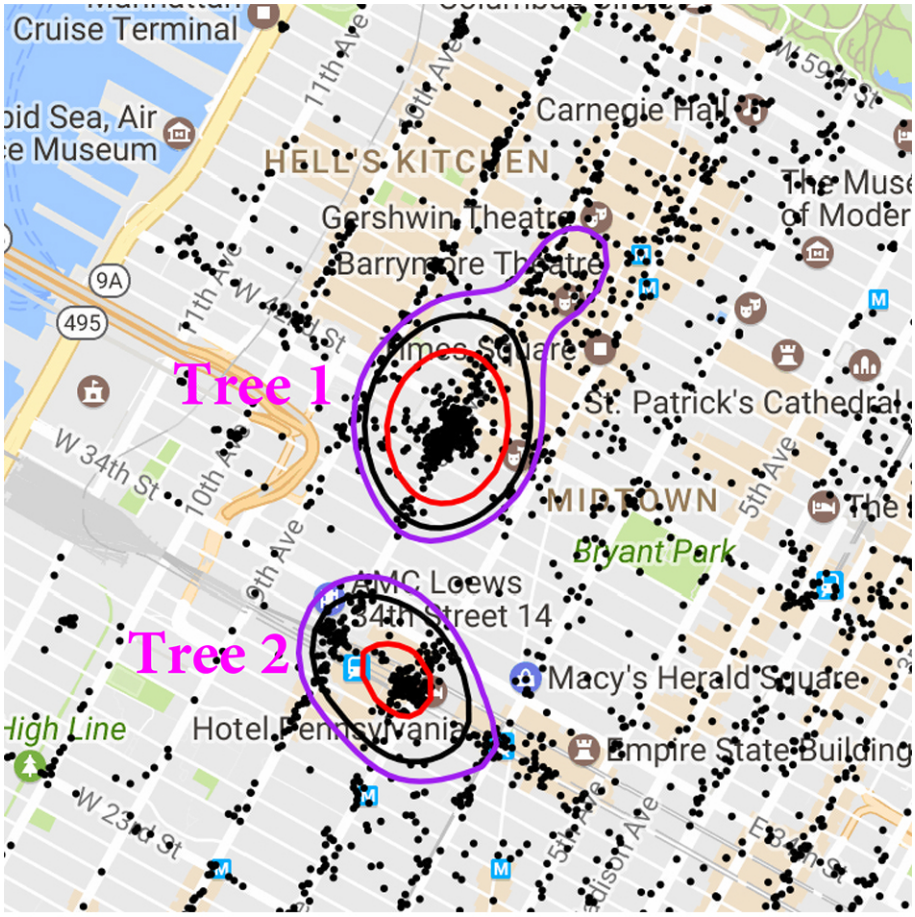


Figure 3.3: Histogram of validity indexes for clusters in Figure 3.2

In the experimental result depicted in Figure 3.2, we observe 43 contour polygon trees. The number of spatial clusters for each level are 43 clusters for root, 26 clusters for level 2, and six clusters for level 3. Figure 3.3 gives the histogram of validity indexes obtained for each cluster. Over 77 percent of the clusters have a validity index larger than 0.5, and over 69 percent of clusters have a validity index larger than 0.6. The average validity index is 0.6415, which shows that most of the

spatial clusters obtained have good quality.



Validity Indexes		
	Tree 1	**Tree 2**
Purple	-0.3799259	-0.2314951
Black	0.07295217	-0.02499342
Red	0.5639454	0.7487018

Figure 3.4: Depicted is a clustering with low quality. In the map, the black points are pick-up locations and the two contour polygon trees are spatial clusters we obtained. Different colors of the contours correspond to different thresholds. The bottom table depicts the validity indexes for each spatial clusters. Note: the map is extracted from the 4-5 am batch subfigure in Figure 3.2.

For those clusters with low validity indexes, it occurs when two clusters are close.

In this case, the validation method considers them as low-quality clusters, suggesting they should be merged. However, due to the characteristics of the contouring algorithm, we will obtain two separate clusters that are close to each other if the region between them has relatively low density. For example, both the purple and black clusters that are shown in Figure 3.4 have low validity indexes, but as we can see from the density of pick-up locations, it is reasonable to split them into two trees. Since there is a gap where the density of points is low, which causes the splitting by the contouring algorithm. The same thing happened to the two trees on the left of the 6-7 am figure in Figure 3.2. The validity indexes for these two trees are black (-0.3362291, -0.2200243) and purple (0.07295217, -0.02499342), which further justify the case that though some clusters obtained by ST-COPOT have low validity indexes, it does not mean they have low quality.

3.4.3 Change Analysis Using ST-COPOT

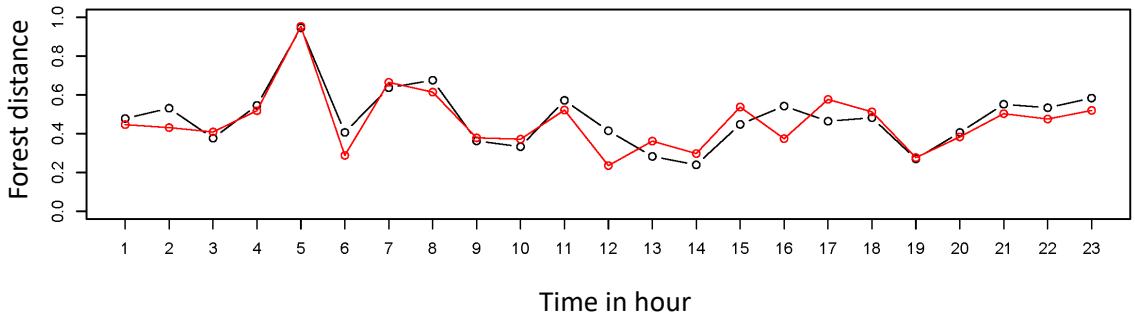


Figure 3.5: Area-weighted serial forest distances for contour polygon forests obtained for consecutive hours (black for Wednesday, red for Thursday)

In most cases, spatio-temporal data is not static, but changes over time. In terms of static data, it is reasonable to assume that the data was generated by a

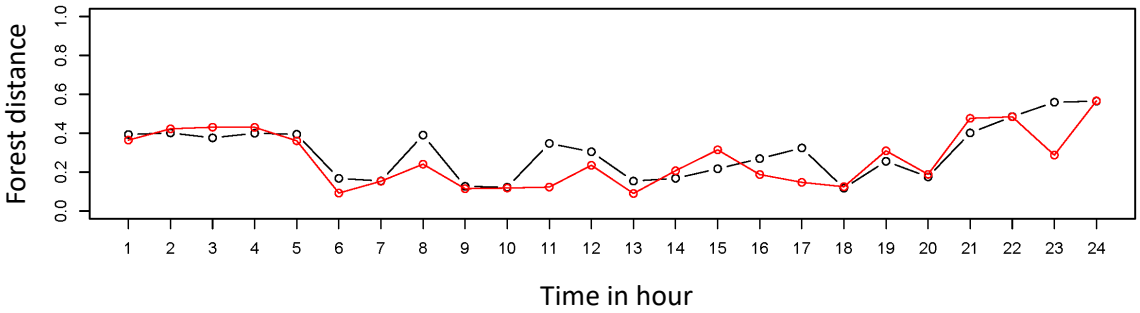


Figure 3.6: Parallel forest distances for contour polygon forests obtained for the same hour of Wednesday and Thursday (black for non-area weighted forest distance function, red for area-weighted forest distance function)

fixed process, but spatio-temporal data has a temporal dimension, and mostly the underlying process that generates the data will change over time [73]. This will create a major challenge in the field of spatio-temporal data analysis, in particular, the change analysis of spatio-temporal data. Below outlines some preliminary results we obtained concerning change analysis.

We tried to verify some interesting hypotheses using ST-COPOT. Such as, for weekdays, how dense regions change over time should show a similar pattern. Dense regions identified for the same time window of different weekdays should be similar. That is, we hypothesize that the forest distance values for consecutive hours of weekdays would exhibit a similar time-series pattern. The forest distance values for the same time window for different weekdays should be small. To verify these hypotheses, we randomly sampled 15% of one-year's data and use Wednesday and Thursday as the example. Experimental results are presented in Figure 3.5 and Figure 3.6. The forest distance value is between 0 and 1, low values mean significant agreement for two contour polygon forests and high values correspond to significant disagreement.

In Figure 3.5, it displays the serial forest distance values for consecutive hours of Wednesday and Thursday, these two time series are really close. Figure 3.6 displays the parallel forest distance values for the same hour of Wednesday and Thursday. The average non-area weighted forest distance value is 0.3 and the average area-weighted forest distance value is 0.27. These two experiments verify our hypotheses that how the dense regions evolve over time shows a very similar pattern for Wednesday and Thursday and the clustering result for the same hour of Wednesday and Thursday shows significant agreement.

3.5 Summary

Table 3.1: Comparison of three algorithms

	ST-DCONTOUR	ST-DPOLY	ST-COPOT
Phase 1	Parametric Gaussian mixture model	Parametric kernel density estimation	Parametric kernel density estimation
Phase 2	One density threshold	One density threshold	Multiple density thresholds
Phase 3	Equation 2.2	Equation 2.2	Equation 2.2 to 2.11

In summary, we present three spatio-temporal clustering algorithms (ST-DCONTOUR, ST-DPOLY, and ST-COPOT) that are based on the serial, three-phase density-contour based framework we proposed. The comparison of these three algorithms is presented in Table 3.1. We improved the algorithms over time and ST-COPOT is the complete implementation of the whole framework. ST-COPOT first employs a non-parametric kernel density estimation approach to obtain the spatial density distribution of the points in each batch. Next, we use multiple density thresholds

in contouring algorithms and post-processing to extract spatial clusters. Our approach at last forms spatio-temporal clusters by using a family of distance functions to identify continuing relationships between temporally consecutive spatial clusters. We demonstrated the effectiveness of these algorithms using the NYC taxi trips data. It shows that all of these algorithms can discover interesting spatio-temporal patterns in taxi pick-up location streams. We also learned that the clustering results have sufficient quality, and the proposed algorithms have great potential in the application of change analysis as well.

Chapter 4

"Serial" versus "Parallel": a Comparison of Spatio-temporal Clustering Approaches

As mentioned before, in order to identify spatio-temporal clusters, one major challenge that needs to be addressed is to determine how spatial and temporal information are combined. In many existing approaches, time and space are treated in a parallel fashion. Approaches that use time and space in a serial fashion are rarely investigated.

In this Chapter, a serial, density-contour based spatio-temporal clustering approach (ST-DPOLY) is compared with a parallel approach called ST-SNN [71], which relies on a spatio-temporal distance function that combines spatial and temporal distances and then modifies the well-established generic clustering algorithm—Shared

Nearest Neighbor (SNN) to operate on that distance function. In contrast to ST-DPOLY, ST-SNN does not subdivide the stream into batches. Instead, it processes all the input data together and uses its spatio-temporal distance function to compute shared neighbors for pairs of spatio-temporal objects and then uses this information to obtain the clustering results.

4.1 ST-SNN: the "Parallel" Approach

The SNN algorithm [25] is a density-based clustering algorithm proposed by Ertöz. It can identify clusters of different shapes, sizes and densities, as well as deal with noise. We generalize the traditional SNN algorithm to create spatio-temporal clusters and the obtained algorithm is called ST-SNN [71]. A spatio-temporal event a is associated with a time t when it occurs and a location vector (la, lo) indicating where it occurs. The distance function for spatio-temporal events is defined as follows:

$$d_{st}(a_i, a_j) = w \times \frac{d_s(a_i, a_j)}{MaxS} + (1 - w) \times \frac{d_t(a_i, a_j)}{MaxT}, \tag{4.1}$$

where a_i, a_j are two spatio-temporal events. Function $d_s(a_i, a_j)$ is any function that can measure the spatial distance between two points on a sphere from their longitudes and latitudes, e.g., the Haversine formula. Function $d_t(a_i, a_j)$ is any function that can compute the temporal difference between two events taking into account the cyclical behavior of time (hours, days, years, season, etc.). w is a weight factor that determines the importance of spatial and temporal distances when measuring spatio-temporal distances ($0 \leq w \leq 1$). $MaxS$ and $MaxT$ are the maximum values of the

spatial distance and temporal distance in the spatio-temporal event dataset, and are used for normalizing the spatial and temporal dimensions. The similarity between a pair of spatio-temporal events a_i, a_j, denoted as $similarity(a_i, a_j)$ is the number of the k nearest spatio-temporal neighbors that they share. Next, SNN density is computed as the sum of the similarities between the event a_i and its k nearest neighbors. Spatio-temporal events that have the SNN density of at least $MinPs$ are labeled as core points. Clusters are then formed by computing the transitive closure of events that can be reached from an unprocessed core event using its k nearest neighbor list. This process continues until all core events have been assigned to a cluster. The remaining events are classified as outliers, which are not included in any cluster.

4.2 Experiments and Analysis

4.2.1 Experimental Results of ST-DPOLY

We reused the TLC Trip Record Data [66] for the evaluation. We used 20 minutes interval as batches and analyzed taxi pick-ups from 6 am to 7 am on January 8th (2014), Figure 4.1 shows the clustering results. For 6-6:20 am batch, we obtained three clusters. For the 6:20-6:40 am batch, we obtained two clusters and we obtained four clusters for the 6:40-7 am batch. According to the result, east of Midtown of New York is a hotspot that is crowded with people looking for taxis early in the morning, as well as the region centered around the Grand Central Terminal.

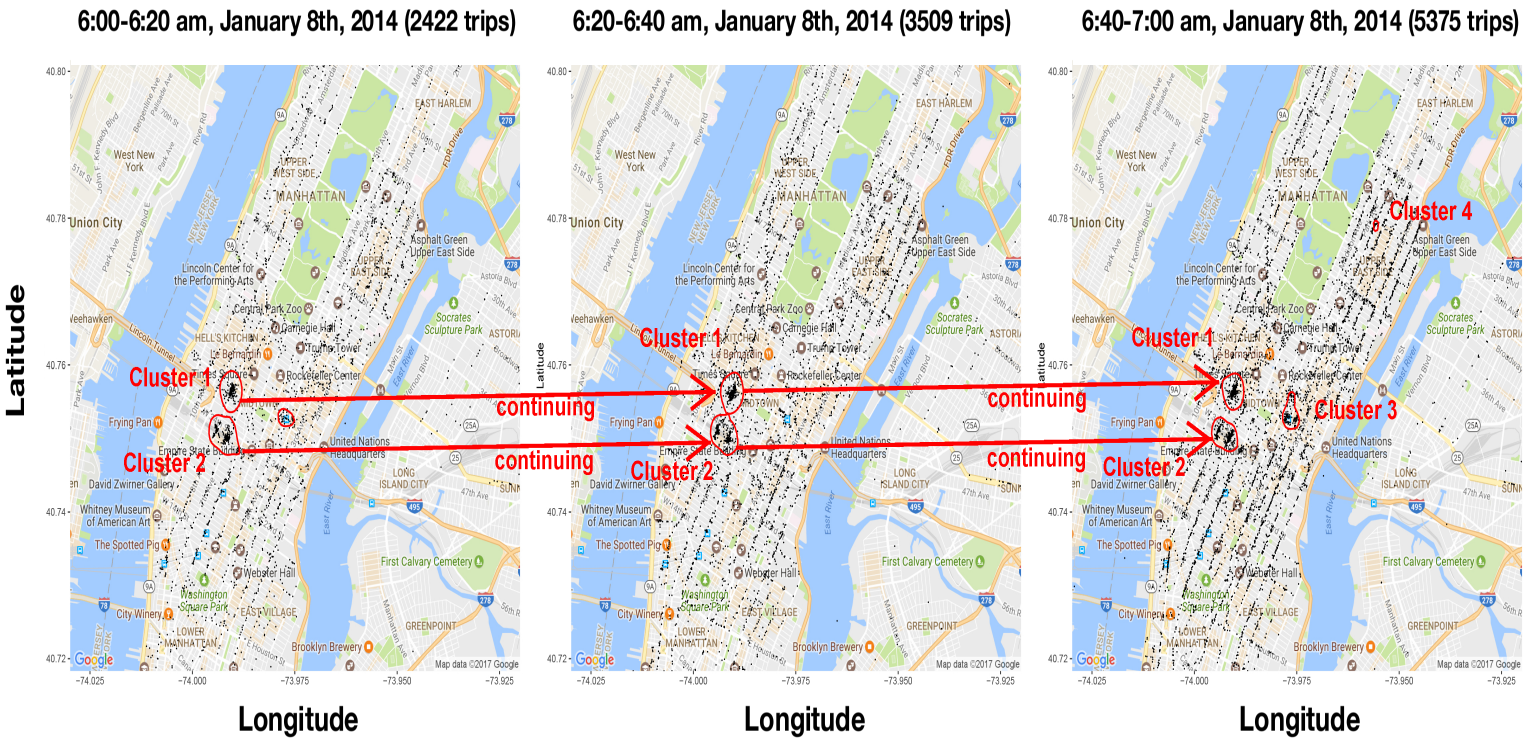


Figure 4.1: Depicted is a spatio-temporal clustering of ST-DPOLY. The black points are taxi pick-up locations and the red contours are spatial clusters we obtained for each batch. The arrows denote the continuing relationship.

Analyzing these results, we can see some interesting patterns. In terms of west of the Midtown area, two clusters continue for three consecutive batches. We find one cluster is near the south-west of Time Square, which is close to several train and bus terminals, such as 42 St - Port Authority Bus Terminal. The other cluster is also centered around several bus terminals and train stations, such as 34 Street Penn Station. We infer that, early in the morning, after New Yorkers get off trains or buses, many people choose to look for taxi rides, which explains the presence of high-density clusters of taxi pick-ups around the train and bus stations. Similarly, cluster 3 in the 6:00-6:20 am batch, and cluster 3 in the 6:40-7 am batch are all centered around the Grand Central Terminal, exhibiting a similar taxi pick-up pattern. For 2nd batch, there is no cluster found at the region around Grand Central Terminal, which shows that there are less passengers looking for pick-ups within the 6:20-6:40 am time frame.

4.2.2 Experimental Results of ST-SNN

We applied the ST-SNN algorithm to the same dataset. The input parameters for ST-SNN are assigned as $k = 100$, $MinPs = 60$ and $w = 0.5$. We used Euclidean distance to compute the spatial distance. There were 16 clusters obtained. Fig 4.2 visualizes clusters 2, 13 and 14, and they are centered around several bus terminals and train stations, which shows a similar pattern of the clustering results generated by ST-DPOLY. Clusters 2 and 13 are similar in the spatial domain, however, the time slots corresponding to these two clusters are different. We also found that ST-SNN can identify clusters with different spatial and temporal densities.

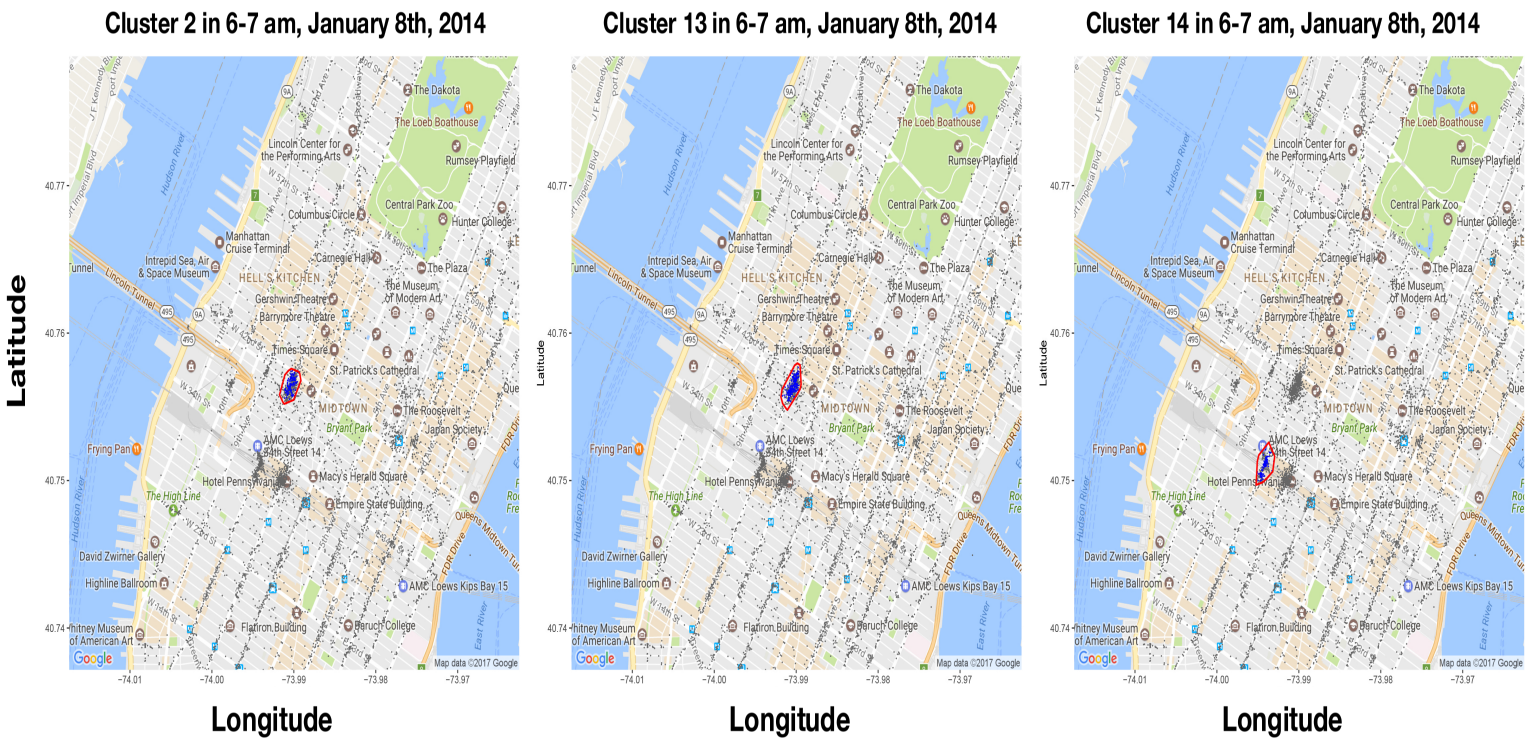


Figure 4.2: Depicted is a spatio-temporal clustering of ST-SNN. The red contours are spatial clusters we obtained for each batch. The black points are taxi pick-up locations and the blue points are pick-up locations that form a cluster. We obtained 16 clusters in total and displayed three of them.

4.3 Comparison between ST-DPOLY and ST-SNN

4.3.1 Time and Space Complexity

Table 4.1: Time and space complexities of ST-DPOLY and ST-SNN (Let's say the grid size we use in ST-DPOLY is $m \times m$, n is the total number of points, e is the average number of edges a spatial cluster has.)

	Time Complexity		Space Complexity	
ST-DPOLY	Phase 1	$O(m^2 \times n)$	Phase 1	$O(n + m^2)$
	Phase 2	$O(m^2)$	Phase 2	$O(m^2 + e^2)$
	Phase 3	$O(e^2)$	Phase 3	$O(e^2)$
ST-SNN	$O(n^2)$		$O(k \times n)$	

Table 4.1 gives the time and space complexities of ST-DPOLY and ST-SNN. For ST-DPOLY, in general, e is smaller than m, and since it is a serial approach, the final complexity would be $O(m^2 \times n)$. In cases that the number of data points is much larger than the number of grid cells ($m^2 << n$), ST-DPOLY's complexity becomes $O(n)$. The time complexity of ST-SNN is the same as SNN, which is $O(n^2)$ without the use of an indexing structure. The space complexity is $O(k \times n)$ since only k-nearest neighbor lists need to be stored. The k-nearest neighbor can be computed once and used repeatedly for different runs of the algorithms with different parameter values. In cases where $m^2 << n$, the space complexity of ST-SNN is worse than ST-DPOLY. In summary, ST-DPOLY is superior to ST-SNN in terms of both time and space complexity.

Table 4.2: Variation measurements of three clusters of ST-DPOLY in Fig. 4.1

	Time	Longitude	Latitude
range	20	0.003552	0.002071
mean (μ)	9	-73.9772	40.7528
sd(σ)	7.33	0.00116	0.000509

(a) cluster 3 from batch 1 (38 points)

	Time	Longitude	Latitude
range	20	0.004863	0.005282
mean (μ)	29.64	-73.9904	40.7563
sd(σ)	5.956	0.000773	0.00076

(b) cluster 1 from batch 2 (216 points)

	Time	Longitude	Latitude
range	0	0.005263	0.004274
mean (μ)	50	-73.993	40.75057
sd(σ)	5.547	0.00147	0.000805

(c) cluster 2 from batch 3 (259 points)

Table 4.3: Variation measurements of three clusters of ST-SNN in Fig. 4.2

	Time	Longitude	Latitude
range	19	0.0026	0.0024
mean (μ)	9.199	-73.9904	40.7565
sd(σ)	5.620	0.0005	0.0005

(a) cluster 2 (141 points)

	Time	Longitude	Latitude
range	19	0.0025	0.0032
mean (μ)	49.10	-73.9904	40.7564
sd(σ)	5.720	0.0005	0.0006

(b) cluster 13 (212 points)

	Time	Longitude	Latitude
range	19	0.002264	0.002881
mean (μ)	49.74	-73.9906	40.6862
sd(σ)	5.519	0.0086	0.009

(c) cluster 14 (122 points)

4.3.2 Temporal Flexibility

In terms of temporal flexibility, ST-SNN is more flexible as clusters have more temporal variation with respect to temporal mean and standard deviation. The temporal variation in ST-DPOLY clusters is significantly limited, since all observations belong to a time window with a fixed size and its clustering results are independent of temporal variation within a particular batch. Though batch size can be selected based on application needs, it is fixed throughout the clustering process once selected, as well as in the clustering result. However, the clustering result of ST-DPOLY is more straightforward, and in terms of change analysis in location streams, ST-DPOLY as a serial approach is more appropriate.

4.3.3 Quality of Clusters

To compare the quality of the clustering result, we measured the variation of the clusters obtained, which is shown in Table 4.2 and 4.3. The clusters generated by ST-SNN have smaller values of standard deviation and range of time, longitude, and latitude than those clusters identified by ST-DPLOY. Depending on parameter settings, ST-SNN can integrate different temporal distance functions and different weights to identify clusters that are dense in both the temporal and spatial domain, while ST-DPOLY only looks for spatial dense regions within each batch, which facilitates the visualization of its clustering results.

4.4 Summary

To summarize the comparison between "serial" and "parallel" spatio-temporal clustering approach, in terms of time and space complexity, ST-DPOLY has advantages over ST-SNN, while ST-SNN is superior in terms of temporal flexibility. In terms of clustering results, results of ST-DPOLY are easier to interpret and more straightforward, while ST-SNN usually obtains a significant number of clusters which overlap either spatially or temporarily. It makes interpreting ST-SNN's clustering results more complicated.

Chapter 5

A Novel Two-stage System for Detecting and Tracking Events in Twitter

5.1 Related Work

5.1.1 Twitter-related Research Approaches

Recently, the potential of analyzing Twitter data has been increasingly recognized by numerous research domains, e.g., social science, information science, geoscience and computer linguistics [64]. When it comes to analyzing data from Twitter, based on which information have been used, the research methodologies are mainly classified into 3 categories: semantic approaches, spatio-temporal approaches and hybrid

approaches (semantic and spatio-temporal).

Semantic approaches analyze the content of tweets only, e.g., Latent Dirichlet Allocation [12] as a probabilistic topic model has been very popular in recent years, and it has been used in a lot of papers [83, 72, 45]. Spatio-temporal approaches use geo-tags and timestamps only, e.g., [50]. The hybrid approach considers the content of tweets as well as geotags and timestamps of the tweet. For example, Boettcher et al. proposed a real-time local event detection scheme through keyword frequency analysis of DBSCAN clustered tweets [13]. Veloso et al. proposed an ST-DBSCAN based approach in which tweets are filtered with a set of keywords [31].

5.1.2 Twitter Event Detection Approaches

When it comes to event detection in Twitter, a variety of approaches have been proposed. Some approaches detect real-world events by detecting abnormal spatial, temporal, and semantic tweet frequencies in real time [74, 16]. Other approaches detect events by analyzing hashtags [19, 18, 28]. There are also approaches that involve popular machine learning techniques (e.g., term frequencyinverse document frequency (tf-idf), Naive Bayes, Support Vector Machine (SVM)). Becker et al. identified the real-world events and news content on Twitter by extracting and classifying topics using tf-idf and Naive Bayes Classifier [8]. Starbird et al. analyzed mass disruption events by using the support vector machine algorithm to identify on-the-ground Twitterers during mass disruptions [63].

There are many existing approaches that use LDA. Weng et al. used wavelet-based signal in Twitter for clustering and classifying events by applying tf-idf and LDA algorithm [72]. Lau et al. proposed an online LDA-based approach to track emerging events in Twitter [49].

The system we propose belongs to the hybrid approach category. It integrates an LDA-based approach and a density-contour based clustering approach. As mentioned above, the LDA topic modeling has been successfully used in many approaches for extracting semantic topics from Twitter. There are existing density-based clustering approaches for event detection on Twitter (e.g., [60]). However, none of them are "serial" approaches, our density-contour based approach uses time and space in a serial fashion. Its advantage over the "parallel" approach is discussed in Chapter 4.

Many existing approaches are designed for specific application, e.g., disaster management [67], disease management [61, 31], traffic management [75], etc. Some approaches need to build a pre-trained classifier, e.g., [60, 8]. But our system is quite general, it takes solely geo-tagged tweets as inputs and applies to all trending topics including disaster, concerts, games, riots, etc. Depending on the application, the user can decide to detect and track events using either "absolute" density or "relative" density as well.

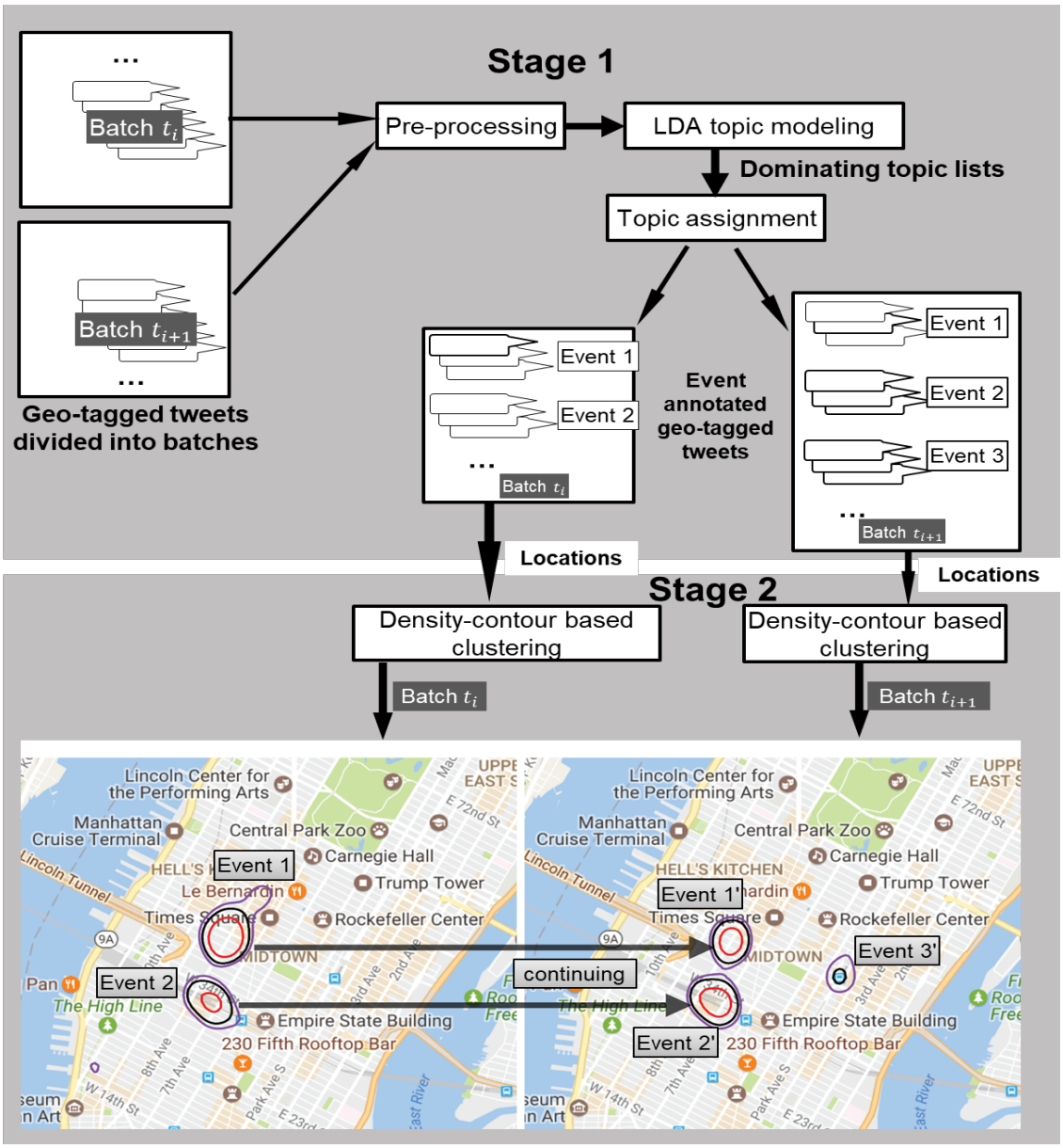


Figure 5.1: Depicted is the architecture of the Twitter event detection and tracking system. It consists of two stages, in the first stage, we take geo-tagged tweets as input and identify events as topics by using an LDA-based topic discovery step. In the second stage, after locations for each event are extracted in each batch, a density-contour based spatio-temporal approach is employed to identify spatio-temporal clusters of each event.

5.2 Overview of the Two-stage System

5.2.1 The Architecture of the System

Figure 5.1 presents an architecture of the proposed two-stage system for detecting and tracking events on Twitter. Stage 1 mainly consists of the following steps:

1. The geo-tagged tweets are divided into temporal batches associated with a fixed-size time window.
2. Each tweet is preprocessed, e.g., removing non-alphabetical characters, URLs, stop words, etc.
3. An LDA-based approach is applied to tweets within each batch to extract dominating topics as our benchmark event labels.
4. The most probable topic is assigned to each tweet by iterating over all the topics, summing up the posterior probabilities of all words in a tweet for each topic. In case the sum of the weights for the most probable topic is less than a certain threshold, no topic label will be assigned to the tweet.

In stage 2, for each event, the density-contour based clustering approach is applied to all event-annotated tweet locations to extract spatio-temporal event clusters [1]. It mainly consists of the following steps:

1. Both relative and absolute kernel density estimation of the locations of a particular topic are obtained.

[1] In case we have more than one event, we run stage 2 for each event separately.

2. Contouring algorithms and post-processing are applied to extract spatial clusters as polygons describing the scope of the event, e.g., three event clusters in Figure 5.1.

3. Spatio-temporal clusters are extracted by establishing the continuity for each event for consecutive batches. Topic continuity is established by calculating KL-divergence between topics, spatio-temporal continuity is established by a family of distance functions.

4. A drill down operation is used to locate the events at different geographical granularities, then steps 1-3 of stage 2 are repeated, more details will be given in Section 5.3.2.5.

By integrating an LDA-based approach and density-contour based clustering approach, we can not only identify trending events from tweet messages semantically as topics, but also identify spatial event clusters and establish event continuity temporally for consecutive batches. Our framework represents events as follows:

- Semantically, an event is viewed as a set of weighted words that describe a specific topic.

- Spatio-temporally, an event is visualized as a set of polygons with a hierarchical density structure indicating the spatial scope of the event, and the continuity of event is visualized as a directed graph connecting event scope polygons for consecutive batches.

5.2.2 Inputs and Outputs of the System

The inputs to our system are geo-tagged tweets, including the content of the tweet, the geolocation and the timestamp of each tweet. Stage 1 will produce a list of topics as intermediate results, each topic consists of a list of weighted words. Stage 2 will produce a set of hotspots on map. Depending on how we calculate the densities, each hotspot is either a region with high density of tweets or a region with high percentage of tweets associated with a particular event. Specifically, each hotspot will be a multilayer of polygons extracted by a contouring algorithm using different density thresholds. After the temporal continuity is established, temporally continuing spatial clusters in consecutive batches will be connected.

In Figure 5.1, on the bottom of Stage 2, it shows an example output of the two-stage system. For batch t_i, we identify two events, both of them consist of three layers of polygons. For batch t_{i+1}, we identify three event clusters. After establishing continuity for both the topics and the spatial event clusters, we find that event 1 and event 2 continue for two batches while event 3 is a newly appearing event at batch t_{i+1}.

5.3 The Two-stage System

5.3.1 Stage 1: LDA-based Topic Extraction

5.3.1.1 Preprocessing

We first divide the geo-tagged tweets into batches based on their timestamps. Next, text preprocessing is applied to each tweet where we remove all non-alphabetical characters, URLs, mentions, and stop words. To facilitate the analysis, we convert all letters to lower case as well.

5.3.1.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) proposed by Blei et al. [12] is a generative model that allows sets of observations to be explained by unobserved groups. If observations are words collected into documents, it hypothesizes that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics. The LDA generative model mainly consists of the following 3 steps.

1. The term distribution β is determined for each topic by

$$\beta \sim Dirichlet(\delta). \tag{5.1}$$

2. The proportions θ of the topic distribution for the document ω are determined by

$$\theta \sim Dirichlet(\alpha). \tag{5.2}$$

3. For each of the N words ω_i,

   a) choose a topic $z_i \sim Multinomial(\theta)$,

   b) choose a word w_i from a multinomial probability distribution conditioned on the topic $z_i : p(w_i \mid z_i, \beta)$,

where β is the term distribution of the topics and contains the probability of a word occurring in a given topic.

Estimating parameters for LDA by directly maximizing the likelihood of the whole dataset is intractable. Multiple approximate estimation methods have been proposed to solve the problem, e.g., variational expectation-maximization (VEM) [38] and Gibbs sampling [32]. For Gibbs sampling, the posterior distribution $p(z|w)$ is obtained by sampling from (for details, see [32, 58])

$$p(z_i = K|w, z_{-i}) \propto \frac{n^{(j)}_{-i,K} + \delta}{n^{(.)}_{-i,K} + V\delta} \frac{n^{(d_i)}_{-i,K} + \alpha}{n^{(d_i)}_{-i,} + k\alpha}, \tag{5.3}$$

where z_{-i} is the vector of current topic memberships of all words without the ith word w_i. The index j indicates that the word w_i and jth term in the vocabulary are equal. $n^{(j)}_{-i,K}$ denotes the frequency of the jth term being assigned to topic K without w_i. d_i is the document in the corpus where w_i belongs to. The dot . indicates performing the summation over this index. The predictive distributions of the term distribution of the topic β and the topic distributions of documents θ is given by

$$\widehat{\beta}^{(j)}_K = \frac{n^{(j)}_K + \delta}{n^{(.)}_K + V\delta} (i = 1, 2, ..., V), \tag{5.4}$$

$$\widehat{\theta}_K^{(d)} = \frac{n_K^{(d)} + \alpha}{n_{(.)}^{(d)} + k\alpha}(d = 1, 2, ..., D). \tag{5.5}$$

There are several implementations that use Gibbs sampling for Bayesian estimation. In our system, we use GibbsLDA++ proposed by Phan et al. [58] as our baseline approach for topic extraction.

5.3.1.3 Topic Assignment

After we train an LDA topic model on all tweets in each batch, we obtain both the topic distribution z_i for all the tweets and the term distributions β for topics. By iterating over all the topics summing up the posterior probabilities w_{ij} of all the words in a tweet, we pick the topic with the highest sum. If the maximal sum is less than a certain threshold θ, no topic label will be assigned. The topic label l is calculated as follows:

$$l = \begin{cases} \text{none} & if\ max(\sum_{j=1}^{J} p(w_{ij} \mid z_i, \beta)) < \theta, i = 1, 2, 3, ..., k \\ \text{i} & with\ maximum\ value\ for\ \sum_{j=1}^{J} p(w_{ij} \mid z_i, \beta), i = 1, 2, 3, ..., k \end{cases}, \tag{5.6}$$

where j is the word index for the tweet, i is the topic index and k is the number of topics. Next, we extract all the locations for the topic that we are interested in, and use them as the input for the density-contour based approach in stage 2 to track the events spatio-temporally.

5.3.2 Stage 2: Spatio-temporal Event Tracking

In this section, we first present two different mechanisms for estimating the densities of the event-annotated tweet locations. After we obtain the location density distribution for the event, we will use contouring algorithms and post-processing to extract the spatial clusters for each event. Finally, the topic continuity is established by calculating the KL-divergence and the spatio-temporal continuity is established by using a family of distance functions.

5.3.2.1 "Absolute" Density Estimation

We use non-parametric kernel density estimation to estimate the "absolute" densities of all locations of event tweets. For more details, please refer to Section 2.4.

5.3.2.2 "Relative" Density Estimation

In epidemiological studies, one important topic is to investigate the dispersion of some diseases within a geographical region. A common objective of such study is to determine the way in which the risk of contraction of the disease varies over the spatial data collection area. To avoid confounding by the underlying population dispersion, it is necessary to not only obtain the disease case location data, but also control case data describing the at-risk population. *Relative Risk Functions* have been proposed in the literature to provide relative density estimation capabilities [11]. In our system, we employ this approach to estimate the relative density of event tweets with non-event tweets.

A straightforward relative risk function can be expressed as the ratio of the actual case density f and control case density g, respectively

$$r(z) = \frac{f(z)}{g(z)}. \tag{5.7}$$

In our application, we consider the event tweet locations as the case, and all the geo-tagged tweets collected in the data collection area as the control case. In order to make the treatment of these two densities symmetric, Kelsall et al. suggest the use of log-risk function $\rho = log(r)$ [44].

When it comes to the bandwidth selection, choosing standard fixed bandwidth would fail in many applications, as the lack of the spatial adaptability would question its appropriateness for relative density estimation. For example, human population distributions tend to be highly heterogeneous with natural geographical features such as rivers, cities, and mountains affecting both the case and control densities within the data collection region. In terms of fixed bandwidth case, a large amount of smoothing is applied to densely populated regions in order to control the noise where the data is sparse. It makes it hard to capture important finer details in densely populated regions [20]. To solve this problem, using an adaptive bandwidth to handle inhomogeneities in the distribution of data has been proposed. In our system, we use the symmetric adaptive smoothing schema proposed by Davies et al. for the bandwidth selection [21]. It uses a single bandwidth function for both cases and controls, which is defined as follows:

$$h(x) = h_0\alpha(x)(i = 1, 2, ..., n), \tag{5.8}$$

where the term $\alpha_i \equiv \alpha(x_i)$ is called ith local bandwidth factor:

$$\alpha(x) = \frac{1}{\gamma f(x)^{1/2}} \tag{5.9}$$

and

$$\gamma = exp\left\{\frac{1}{n}\sum_{i=1}^{n} log[f(x_i)^{-\frac{1}{2}}]\right\}. \tag{5.10}$$

From the equations above, we can see that the kernel smoothing will decrease for relatively high density regions, and increase for regions with sparse observations.

5.3.2.3 Edge Correction for Density Estimation

As the data is collected over a restricted region, there is a common problem that part of the kernel contributions of observations that lie near the observation area boundary are underestimated. Since there are no observations occurring on the other side of the boundary, it creates bias near the boundary. To solve this issue, Marshall et al. [52] proposed a *sample-point adaptive* estimator, defined as:

$$\widehat{f}_{h_0}(y|X) = \frac{1}{n}\sum_{i=1}^{n} h(x_i; f)^{-2} K\left(\frac{y - x_i}{h(x_i; f)}\right) q_{h(y)}(y|D)^{-1} \quad (y \in D), \tag{5.11}$$

where D is the study region, $q_{h(y)}(y|D)^{-1}$ is the correction factor, which is calculated as:

$$q_{h(y;f)}(y|D)^{-1} = h(y; f)^{-2} \int_D K\left(\frac{u - y}{h(y; f)}\right) du \quad (y \in D). \tag{5.12}$$

This can be considered as the proportion of the kernel weight that falls within D for a kernel centered at y with bandwidth h.

5.3.2.4 The Extraction of Spatial and Spatio-temporal Clusters for an Event

After we get a density distribution of locations of event-annotated tweets at the data collection area, we identify spatial dense regions as spatial clusters using contouring algorithms and post-processing. For more details, see Section 2.3 .

To extract spatio-temporal clusters of the event, we need to establish the temporal continuity including the topic continuity and the spatial continuity. The topic continuity among consecutive batches is established as follows. Since we train an LDA for tweets in each batch, as a result, we get a list of topics for each batch. By calculating the KL-divergence of topic lists for consecutive batches, we are able to calculate the distances between those topics. If the distances between two topics are less than a specific threshold, we consider them as continuing topics/events.

In order to identify a continuing relationship between spatial clusters of the same event obtained for two consecutive batches, we will reuse distance functions introduced in Section 2.4.

5.3.2.5 Drill Down Operation to Locate Events at different Geographic Granularities

Using the system we introduced so far, we are able to obtain the spatial event clusters that locate the area affected by a particular event. But sometimes we want to locate the center of an event or we want to provide a much more fine-grained analysis of the spatial variation in the region occupied by the spatial cluster. We propose the

following drill down operation.

We use the region of the spatial cluster as the observation area, rerun the 2nd stage of the system with a much smaller grid cell size, much higher density thresholds, and larger bandwidth for density estimation step. Since the event center will be a much smaller region compared to the region affected by an event. The key idea of the drill down operation is to initially identify spatial clusters by just running the spatial clustering framework for a few parameters at a somewhat low level of granularity, and then to conduct a more in-depth analysis for the obtained spatial clusters. We also claim that this approach is more efficient, as fine-grained analysis is only conducted for the regions obtained as spatial clusters and not for other parts of the observation area. That is, we initially identify the spatial clusters using a larger grid cell size—the spatial clustering will also be obtained more quickly for a larger grid cell size—and then, supposing we get multiple contour polygon trees, we use the root polygon of each tree as the new boundary of the observation area and rerun the second stage of the system with a new set of parameters to obtain more fine-grained contour lines.

In summary, after we obtained spatial clusters, the drill down operation provides a much more fine-grained analysis of the spatial variation in the region that the spatial cluster occupies.

5.4 Experimental Results

We demonstrate and verify our approach using two real-world events and geo-tagged tweets data collected from Twitter. Data is downloaded from [57].

5.4.1 Case Study 1: Buffalo Snowstorm Event

The case study centers on a snowstorm event. On November 17-19 2014, a major lake effect snowstorm hit the Great Lakes region, dumping a record-breaking eight feet (2.4 m) of snow in the Buffalo, New York region. The storm stranded hundreds, killed 13, and caused a state of emergency.

Table 5.1: The top 20 words of snowstorm-related topic for three consecutive days

Date	Word list of snow-related topic
Nov 17	back, going, work, home, school, snow, morning, class, wait, sleep, tomorrow, rain, wanna, days, bed, come, cold, coming, weather, winter
Nov 18	snow, today, school, buffalo, tomorrow, house, still, days, car, crazy, high, coming, south, stuck, closed, open, help, feet, west, hit
Nov 19	snow, buffalo, still, house, days, way, long, help, car, weather, stay, finally, hit, open, hours, away, storm, hour, closed, end

We applied our approach to daily batches and used the State of New York as the data collection area. The word list of the snowstorm-related topic for three consecutive days is presented in Table 5.1. The KL-divergence between these topics are 0.6424 (Nov 17→Nov 18) and (Nov 18→Nov 19) 0.4524. Both values indicate a strong continuity among the topics listed in 5.1. Some sample tweets are presented in Table 5.2. As outlined, people started tweeting about the snow on Nov 17, but

Table 5.2: Sample snowstorm-related tweets for three consecutive days

Date	Sample snow-related tweets
Nov 17	1. And the snow has begun to fall in Buffalo. 2. Wet snow this morning. 3. And the heavy snow begins.
Nov 18	1. Drove by 4 cars stuck on the California Abbott hill. Deepest snow on Route 20 I've ever seen. #LakeEffect #Snow #Warning. #WNYwx 2. Only in West Seneca still stuck at work with co-workers and the snow continues @ Rosina Food Prods.
Nov 19	1. My car buried in the snow behind me @Comfort Inn near Walden Galleria Mall 2.Snowstorm in Buffalo Area Kills at Least 4.

the snow hadn't caused a serious problem yet. On Nov 18 and Nov 19, the topic shifted a little and people started tweeting about the effect of the snowstorm, e.g., traffic, house, school, car, etc.

Figure 5.2 shows the clustering result for the snowstorm event. As depicted, Buffalo city is a continuing hotspot of the snowstorm events for two consecutive days. But according to the heat map, we know that there is also a high density of snow-related tweets at New York City, which has a much higher tweet density.

Next, we focused on the Buffalo metropolitan area only, where the clustering results are presented in Figure 5.3. Using relative density, we are not able to identify any continuing clusters. On November 18th, three dense regions are identified, east of Cheektowaga, West Seneca, and a small region at West Seneca. Using absolute density, a continuing cluster at the west of West Seneca is identified. Another cluster is located in downtown Buffalo, the area of cluster increases on the following day.

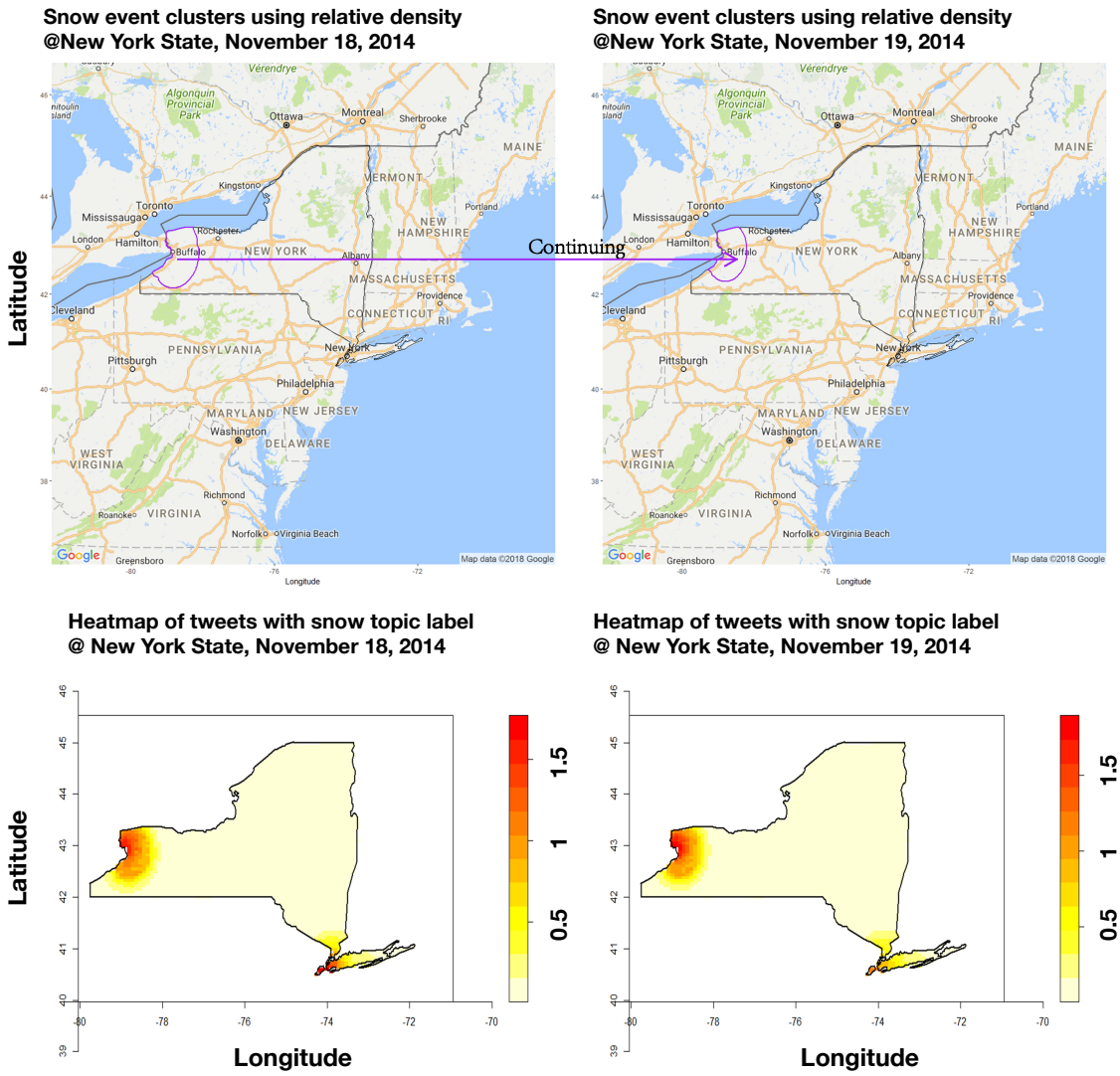


Figure 5.2: Depicted are the snowstorm event tracking results using New York State as our data collection area. The upper two subfigures are clustering results for two consecutive days using relative density, the purple polygons are snowstorm event clusters and the arrow denotes the continuing relationship. The lower two subfigures are the heatmaps of tweets with snowstorm event label, which reflects the absolute density.

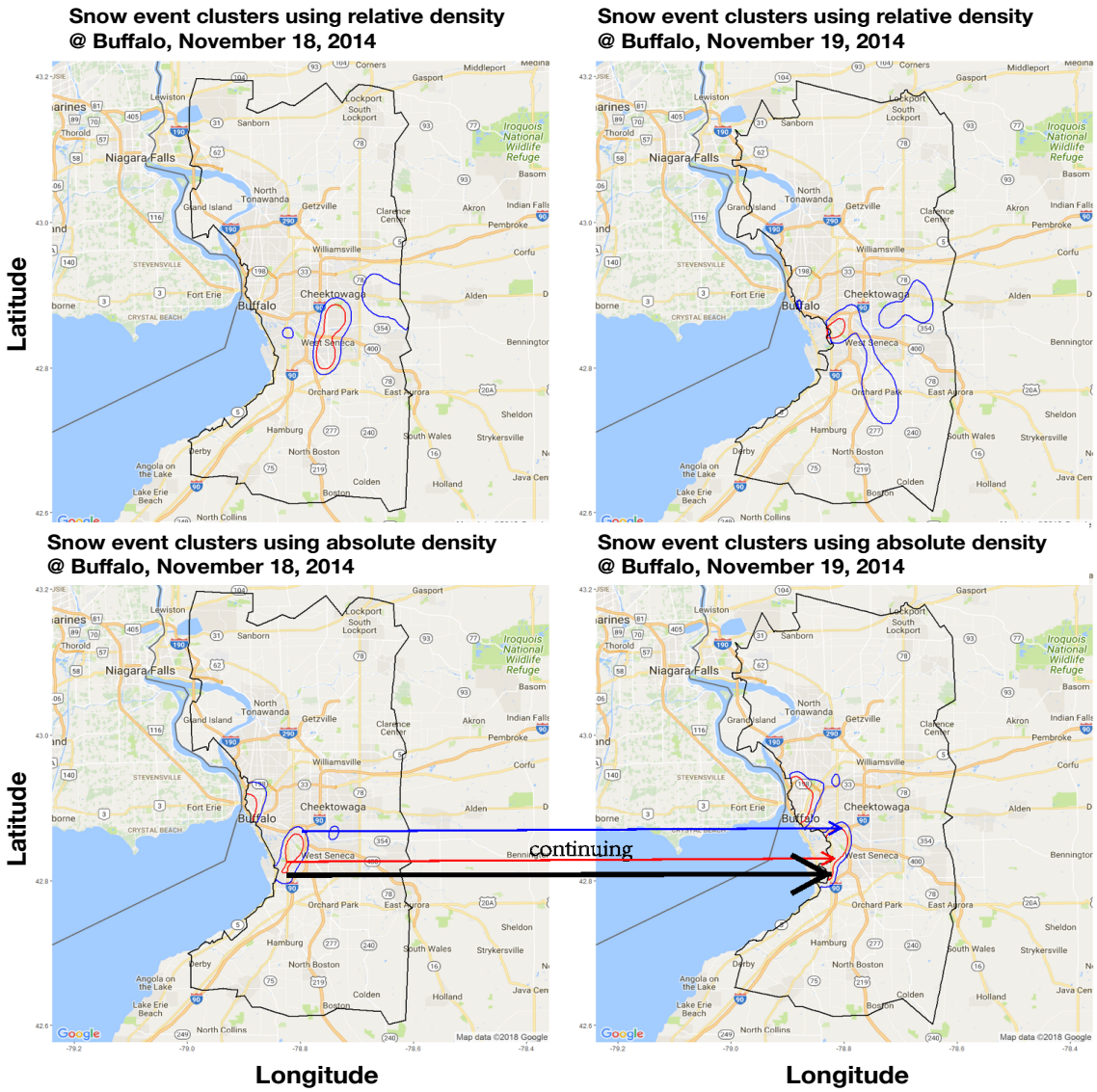


Figure 5.3: Depicted are the snowstorm event tracking results for two consecutive days using Buffalo City as our data collection area. The upper two subfigures are clustering results using relative density and the lower two subfigures present results using absolute density. The contours are snowstorm event clusters and different colors correspond to different thresholds. The arrow denotes the continuing relationship and different colors correspond to different cases (black arrows: continuing contour polygon trees, other arrows: continuing polygons).

5.4.2 Case Study 2: Ferguson Riots Event

The second case study centers on a riot event. On November 24-26, riots break out in Ferguson (a suburb of St. Louis, Missouri) after it was announced that there was insufficient evidence to indict Officer Darren Wilson for shooting Michael Brown. The protests included mass looting and the burning of 12 buildings in Ferguson, as well as 29 arrests.

Table 5.3: The top 20 words of riot related topic for three consecutive days

Date	Word list of riot-related topic
Nov 24	ferguson, tonight, police, city, louis, stl, brown, grand, jury, decision, safe, justice, family, wilson, peace, fire, stay, violence, verdict, fergusondecision
Nov 25	ferguson, police, city, louis, say, man, stl, well, see, missouri, fire, protesters, car, keep, grand, national, guard, michael, burning, kansas
Nov 26	ferguson,people, police, stl, right, tonight, missouri, need, black, man, brown, live, south, grand, see, new, car, great, louis, way

Table 5.4: Sample riot related tweets for three consecutive days

Date	Sample snow-related tweets
Nov 24	1. The Michael Brown case verdict is supposed to be announced at 9 ET. I believe Ferguson will violently riot no matter what. Start praying now 2. The purge is beginning down here! #StLouis.
Nov 25	1. Tear gas everywhere, police dogs, helicopter circling overheard #Ferguson 2. Wild afternoon protesters shut down I70 and police use pepper spray and arrests to disperse crowd. #mikebrownverdict.
Nov 26	1. National guards take over #Ferguson tonight. 2. 44 arrested tonight. #Ferguson

Table 5.3 shows the word list of the riot-related event and Table 5.4 shows some

sample tweets. People started tweeting about the decision on Nov 24 and the riot had started. On Nov 25 and Nov 26, the topic shifted a little and people started tweeting about how the government handled the riot, e.g., National Guard, police, etc.

We first applied our approach to daily batches and used State of Missouri as the data collection area. The result with absolute density is presented in Figure 5.4. We are able to identify continuing clusters at St. Louis. For relative density, we get similar results.

Next, we focused on the City of St. Louis only. The clustering result is presented in Figure 5.5. Using relative density, we were able to locate continuing clusters at Ferguson from November 24th to November 25th. The identified cluster became larger on November 26th. On November 25th and November 26th, the density became higher, which reflected that the riot event discussions became more intense. Using absolute density, we obtained a continuing cluster at Ferguson for three consecutive days. We also identified a cluster in downtown St. Louis on November 25th, which means a high density of people at downtown St. Louis who also cared about this event and discussed it on Twitter.

5.4.3 Practical Experience of LDA-based Topic Extraction

When we designed and evaluated our system, we applied LDA to a lot of tweets. This section summarizes our experiences in using LDA-based approaches to identify topics in tweets.

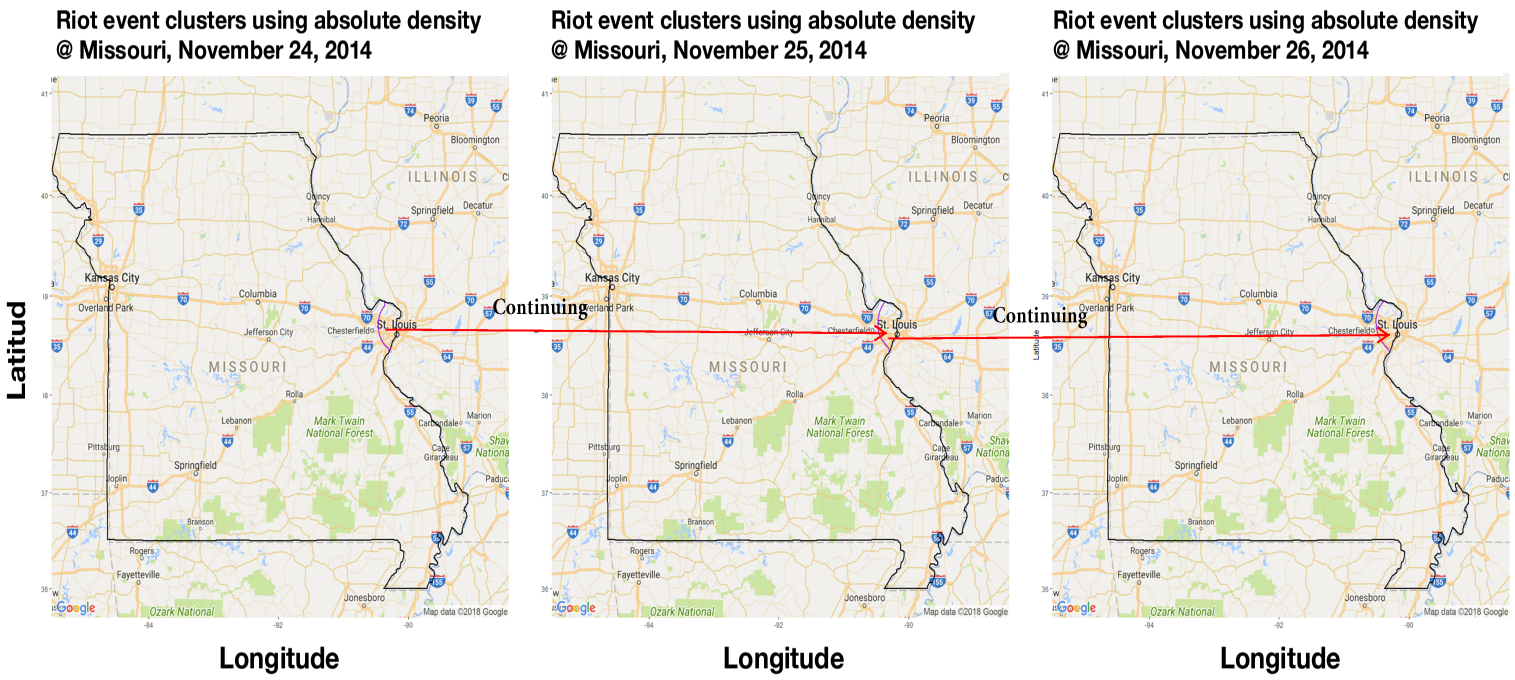


Figure 5.4: Depicted are the Ferguson riot event tracking results using State of Missouri as our data collection area. The figures are clustering results for three consecutive days using absolute density, the purple polygons are riot event clusters and the arrow denotes the continuing relationship.

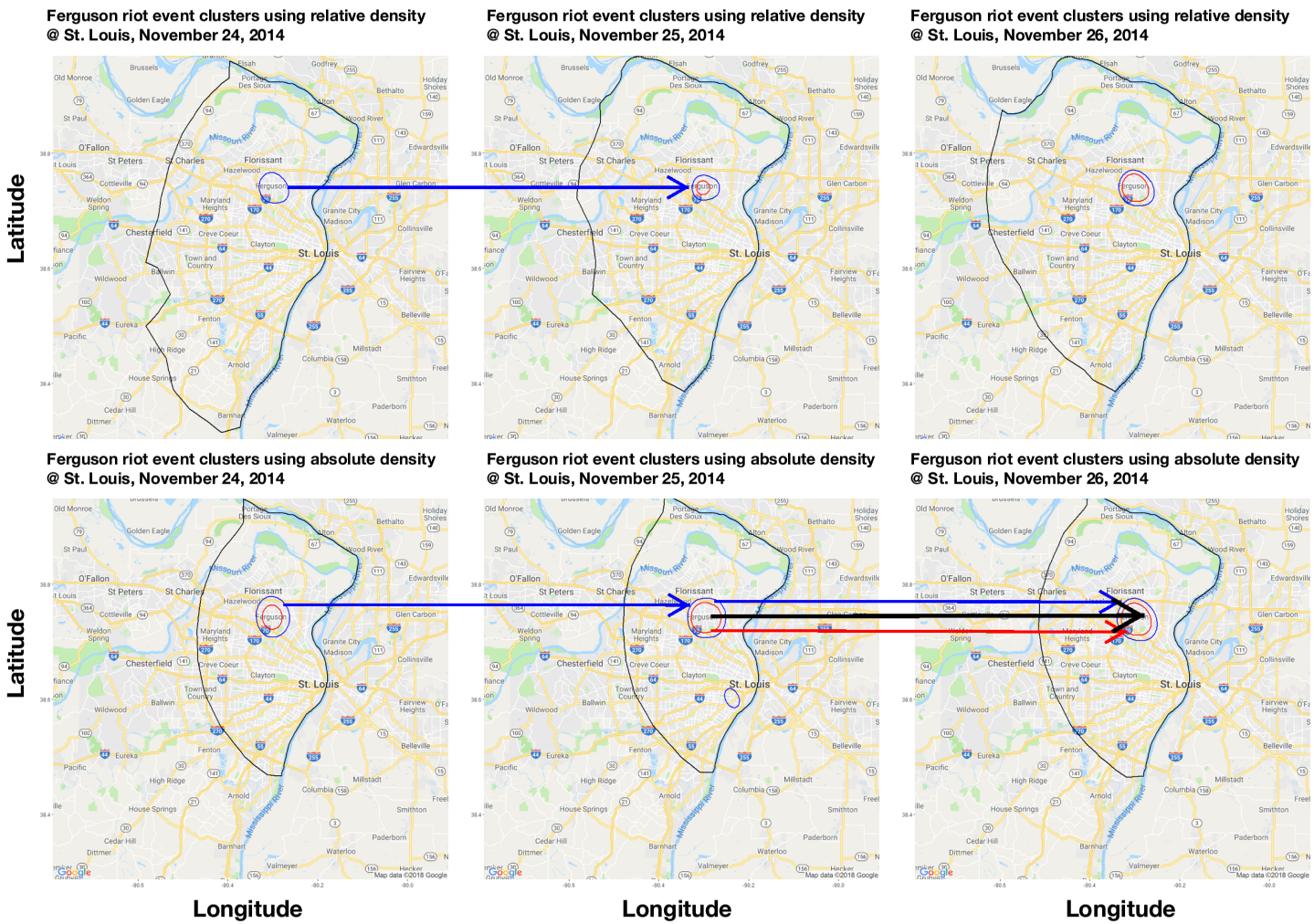


Figure 5.5: Depicted are the Ferguson riot event tracking results for three consecutive days using City of St. Louis as our data collection area. The upper three subfigures are clustering results using relative density and the lower three subfigures present results using absolute density. The contours are riot event clusters and different colors correspond to different thresholds. The arrow denotes the continuing relationship and different colors correspond to different cases (black arrows: continuing contour polygon trees; other arrows: continuing polygons).

- People use Twitter to report daily routine activities, which are not of much interest to our system and are challenging to be filtered out automatically. For example, some people like to post a tweet before they go to bed. Some examples of top-weighted words for such topic are night, sleep, late, tired, bed, etc.

- LDA-based tools force us to give a number k and then the LDA-based tools identify exactly k topics for an input set of tweets. In case our selected k value is too small, we risk missing out an important topic, e.g., sometimes the important event might be ranked lower than the daily routine topic.

- In the case we choose a high k value, sometimes we might end up with more than one topic discussing the same event. In extreme case, during a popular event, such as the FIFA World Cup when the majority of the tweet stream is discussing the same topic, there will be one single dominating event. Almost all the topics identified will be referring to the same event. To deal with this problem, we have to post-process the topic list to merge tweets that are associated with the same event. Again, it seems difficult to come up with an automated procedure for this "unavoidable" topic merging task.

- LDA is very sensitive to initialization as well: different runs will return different results, which exhibit discrepancies with respect to top-weighted words and top-ranking topics.

- In order to get a clean word and topic lists with LDA, we have to use a large list of stop words. Initially we used LDA with a default stop word list and

the results the topic list we obtained were often "blurry" and very hard to understand. Only after we merged stop word lists from multiple sources, were we able to identify more "crisply" defined topics.

5.5 Extension to Emotion Mapping

We extended the proposed two-stage system and successfully applied it to a Twitter's emotion mapping approach [6]. Here are some major modifications:

- At the first stage, instead of doing LDA-based topic modelling, we applied a rule-based sentiment analysis model called VADER to obtain the emotional scores for each tweet [30]. It is a sentiment analysis tool specifically built for the sentiments expressed in social media. It uses a lexicon of commonly used sentimental words to rate each tweet. For each tweet, the analyzer parses the tweet and then checks its lexicon for the sentimental words. Finally, the weighted average of sentimental words is returned as the final emotional score. The obtained scores range in [-1,1], represent negative, neutral, and positive emotions.

- At the second stage, instead of doing regular relative and absolute density estimation, we came up with an emotion weighted density estimation approach. The density in datasets O is defined as follows: the influence of o on v is measured as a product of $E(o)$ and a Gaussian kernel function. The influence of object $o \in O$ on a point $v \in F$ is defined below:

$$f_{influence}(v, o) = E(o) * e^{\frac{-d(v,o)^2}{2\sigma^2}}. \tag{5.13}$$

If $\forall o \in O$, $E(o) = 1$ holds, the above influence function becomes a Gaussian kernel density function. The parameter σ determines how quickly the influence of o on v decreases as the distance between o and v increases. The accumulated influence of all data objects $o \in O$ on a point $v \in F$ is used to define a density function $\psi^O(v)$, as follows:

$$\psi^O(v) = \Sigma_{o \in O} f_{influence}(v, o). \tag{5.14}$$

- In addition to identifying a continuing relationship, the new approach generalizes it to a change analysis schema by looking at other relationships as well, e.g., growing, shrinking, disappearing, etc.

We have used daily batches in our experiments. Figure 5.6 and Figure 5.7 show some experimental results of the emotion mapping framework. As we can see, overall there are more positive emotions. One interesting observation is that June 1 seems to have more clusters than June 2. This can be attributed to the fact that June 1 was a Sunday when people were more active on social media. We also see that the spatial clusters are more concentrated in the cities, e.g., Buffalo and New York City.

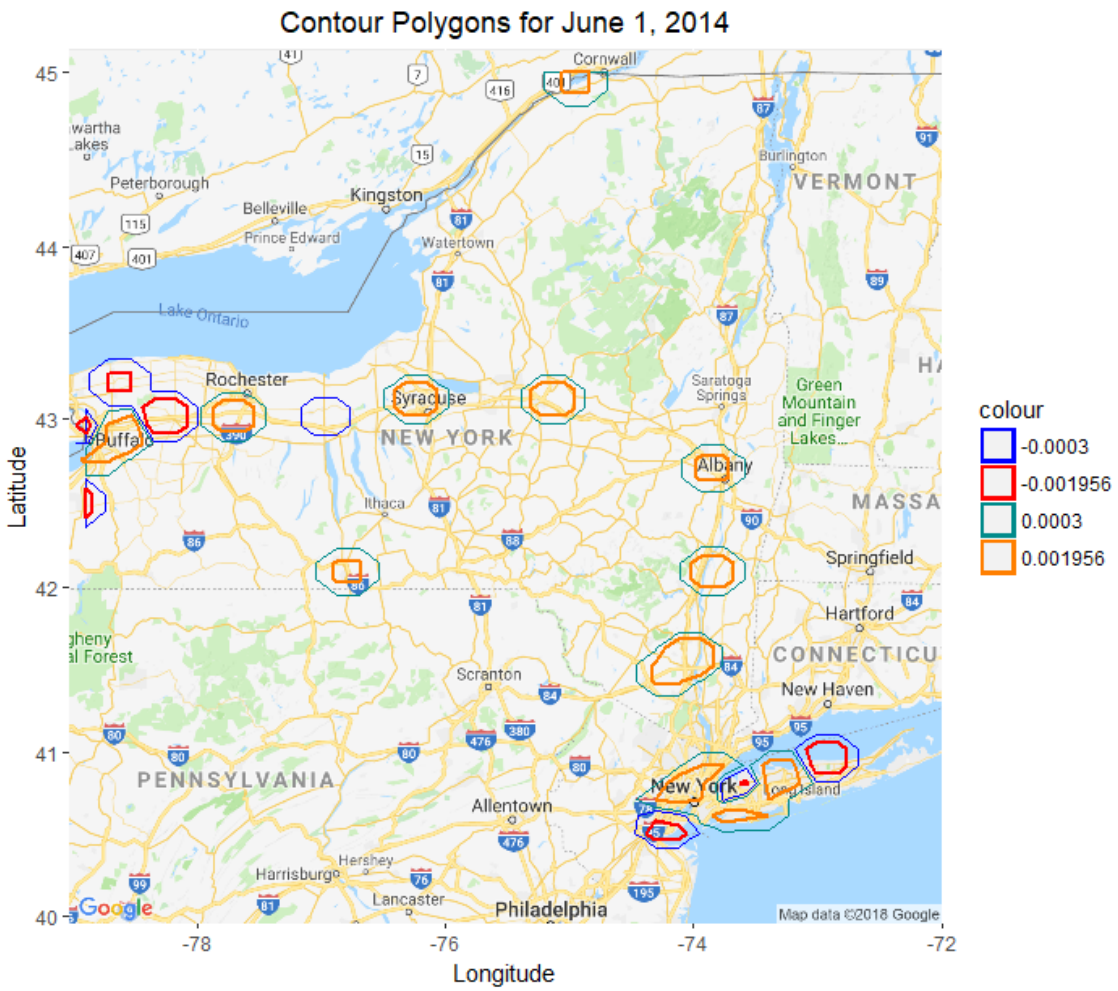


Figure 5.6: Depicted is a spatial clustering of emotions in tweets for June 1, 2014. The contours are spatial emotion clusters and different colors correspond to different thresholds. Red and blue contours are clusters with high negative emotion while green and orange contours are clusters with high positive emotion.

5.6 Summary

The chapter introduces a novel serial, spatio-temporal system for detecting and tracking trending events in tweet streams. The system characterizes such events using a

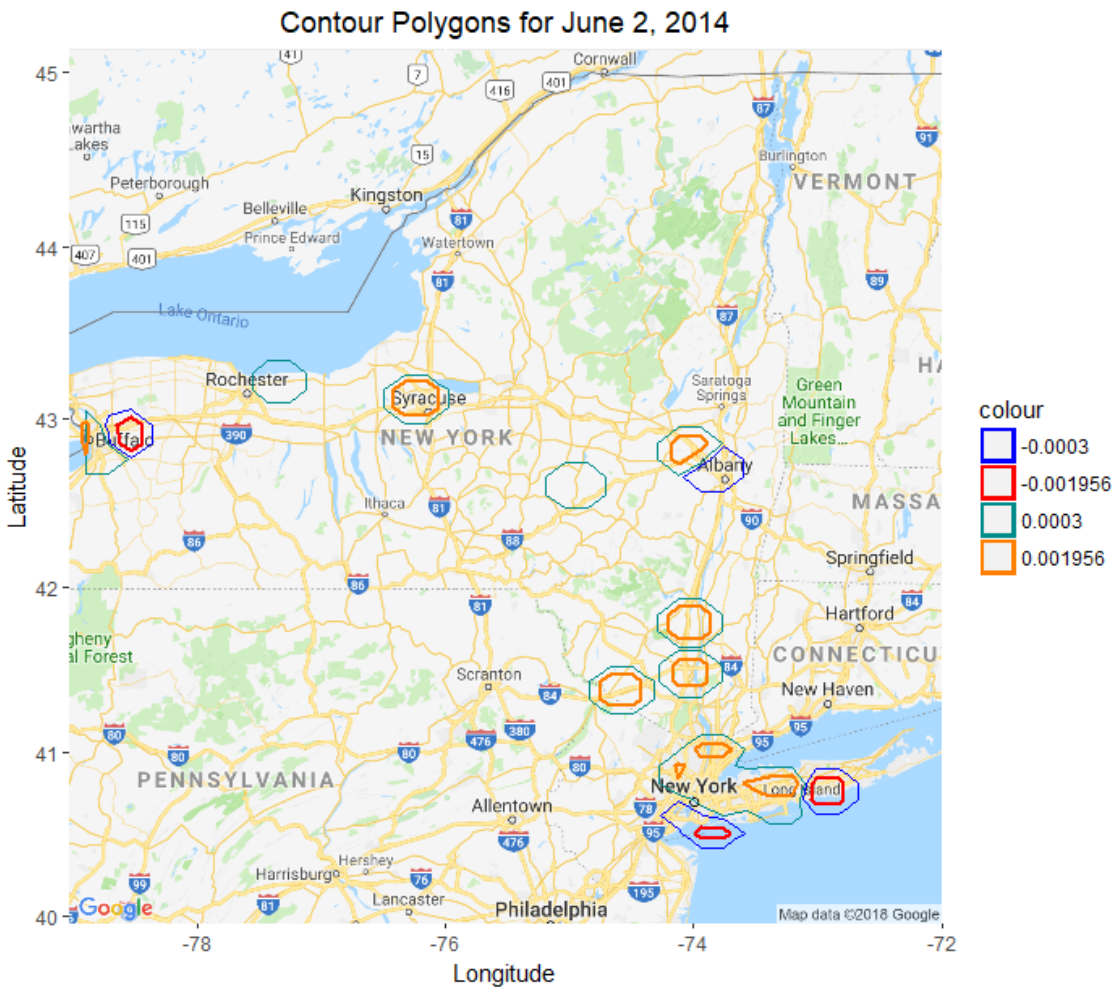


Figure 5.7: Depicted is a spatial clustering of emotions in tweets for June 2, 2014. The contours are spatial emotion clusters and different colors correspond to different thresholds. Red and blue contours are clusters with high negative emotion while green and orange contours are clusters with high positive emotion.

set of weighted keywords, spatial regions describing where the particular event occurs and by establishing continuity of those event regions over time. The approach relies on contouring algorithms to obtain such regions and area-weighted distance functions to assess temporal continuity. This is the first system that considers relative

densities in addition to absolute densities for event tracking, and our experimental results demonstrate the need for providing such capabilities. Our spatio-temporal clustering approach supports a drill drown operation that identify high densities of tweets about a particular event at finer-grained level of granularity. Through two case studies, we demonstrate that the proposed system can effectively detect and track trending events. At last, our proposed approach has been successfully applied to an emotion mapping approach, which proves its effectiveness.

Chapter 6

Conclusion

The main objective of the presented study is the development of density-contour based framework for spatio-temporal clustering algorithms and event detection. In this research, we proposed a serial, density-contour based framework for spatio-temporal clustering of point cloud streams (e.g., location streams), including several algorithms we proposed, namely ST-DCONTOUR, ST-DPOLY, and ST-COPOT. They all rely on the three-phase clustering approach, which takes the point cloud stream as input and divides it into batches based on fixed-size time windows. Next, a density estimation approach and contouring algorithms were employed to obtain spatial clusters as polygon models. Finally, spatio-temporal clusters were formed by identifying continuing relationships between spatial clusters in consecutive batches. The framework was successfully applied to NYC taxi trips data, the experimental results showed that all the algorithms could effectively discover interesting spatio-temporal patterns in taxi pick-up location streams.

To further utilize the density-contour based framework, we proposed a novel system to detect and track events from Twitter streams by integrating an LDA-based topic discovery approach with our density-contour based approach. The system characterizes events using a set of weighted keywords, spatial regions describing where the particular event occurs and by establishing continuity of those event regions over time. This is the first system that considers relative densities in addition to absolute densities for event tracking. We also proposed a drill drown operation that identify high densities of tweets about a particular event at very fine levels of granularity. The experimental results demonstrated that the proposed system can effectively detect and track trending events.

In summary, the main contributions are as follows:

- We propose a serial, density-contour based framework for spatio-temporal clustering.

    1. Our approach treats time and space in a serial fashion, which creates spatial clusters first and then spatio-temporal clusters are constructed as continuing spatial clusters in consecutive batches. Our proposed approach operates directly on density functions and uses density contours and contour analysis to create spatio-temporal clusters from spatio-temporal data streams.

    2. We propose a novel data structure called *contour polygon tree* as our spatial cluster model. We propose a family of novel distance functions that operate on contour polygon trees to establish continuity between spatial

clusters in consecutive batches. Spatio-temporal clusters are created at different levels of granularity, e.g., continuing polygons, continuing trees, and continuing forests.

3. The proposed framework is time efficient and can achieve approximately linear time complexity for large-scale point cloud streams.

4. We evaluate the algorithms that are based on the proposed framework in a challenging real-world case study involving NYC taxi trips data [66]. The experimental results show that all of them can effectively discover interesting spatio-temporal patterns in taxi pick-up location streams.

- We propose a novel two-stage system by integrating an LDA-based approach and an efficient density-contour based approach for event detection and tracking in Twitter.

  1. Our approach employs "relative" and "absolute" density, in particular, the density contours for event detection in Twitter.

  2. We propose a drill down operation—that operates on the identified event clusters that have been obtained by our system—which summarize the spatial variation of tweet locations that are related to an event at a finer granularity. This is accomplished by rerunning the spatial clustering algorithms with a different set of parameters, such as different density thresholds, kernel bandwidth, and grid cell sizes.

3. We demonstrate our approach using real-world data collected from Twitter. The experimental results show that the proposed system can effectively detect and track events from tweets.

6.1 Looking forward

Our work on density-contour based framework inspired various follow-up research, e.g., efficient clustering of data streams, as well as novel systems that require efficient clustering approaches. Spatio-temporal clustering is becoming more important as huge amounts of spatio-temporal datasets are becoming available these days. We envision more efficient and reliable spatio-temporal clustering algorithms to come up in the next several years.

We provide initial work for density-contour based systems for event detection scenarios. We envision novel work that may further push the boundary and improve the efficiency and reliability of these systems. The potential use for our work in this dissertation could be extended to other systems that require tracking temporal changes as well.

Bibliography

[1] M. Ackerman and S. Dasgupta. Incremental clustering: The case for extra clusters. In *Advances in Neural Information Processing Systems*, volume abs/1406.6398, pages 307–315, 2014.

[2] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In *Proc. International Conference on Very Large Data Bases*, pages 81–92, Berlin, Germany, September 9-12 2003.

[3] A. Amini and T. Y. Wah. Dengris-stream: A density-grid based clustering algorithm for evolving data streams over sliding window. In *Proc. International Conference on Data Mining and Computer Engineering*, pages 206–210, 2012.

[4] A. Amini, T. Y. Wah, and H. Saboohi. On density-based data streams clustering algorithms: a survey. *Journal of Computer Science and Technology*, 29(1):116–141, 2014.

[5] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. In *Proc. ACM International Conference on Management of Data*, pages 49–60, Philadelphia, PA, USA, June 1-3 1999.

[6] R. Banerjee, K. Elgarroussi, S. Wang, Y. Zhang, and C. F. Eick. Tweet emotion mapping: Understanding us emotions in time and space. In *Proc. IEEE International Conference on Artificial Intelligence and Knowledge Engineering*, pages 93–100, Laguna Hills, CA, USA, September 26-28 2018.

[7] D. Barbará and P. Chen. Using the fractal dimension to cluster datasets. In *Proc. ACM International Conference on Knowledge Discovery and Data Mining*, pages 260–264, Boston, MA, USA, August 20-23 2000.

[8] H. Becker, M. Naaman, and L. Gravano. Beyond trending topics: Real-world event identification on twitter. In *Proc. International AAAI Conference on Weblogs and Social Media*, Barcelona, Catalonia, Spain, July 17-21 2011.

[9] V. Bhatnagar, S. Kaur, and S. Chakravarthy. Clustering data streams using grid-based synopsis. *Knowledge and Information Systems*, 41(1):127–152, 2014.

[10] D. Birant and A. Kut. St-dbscan: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, 60(1):208–221, 2007.

[11] J. Bithell. Estimation of relative risk functions. *Statistics in Medicine*, 10(11):1745–1751, 1991.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

[13] A. Boettcher and D. Lee. Eventradar: A real-time local event detection scheme using twitter stream. In *Proc. IEEE International Conference on Green Computing and Communications*, pages 358–367, Besancon, France, November 20-23 2012.

[14] F. Cao, M. Estert, W. Qian, and A. Zhou. Density-based clustering over an evolving data stream with noise. In *Proc. SIAM International Conference on Data Mining*, pages 328–339, Las Vegas, NV, USA, June 26-29 2006.

[15] S.-H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1, 2007.

[16] J. Chae, D. Thom, H. Bosch, Y. Jang, R. Maciejewski, D. S. Ebert, and T. Ertl. Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition. In *Proc. IEEE Conference on Visual Analytics Science and Technology*, pages 143–152, Seattle, WA, USA, October 14-19 2012.

[17] Y. Chen and L. Tu. Density-based clustering for real-time stream data. In *Proc. ACM International Conference on Knowledge Discovery and Data Mining*, pages 133–142, San Jose, CA, USA, August 12-15 2007.

[18] M. Cordeiro. Twitter event detection: combining wavelet analysis and topic inference summarization. In *Doctoral Symposium on Informatics Engineering*, pages 11–16, 2012.

[19] A. Cui, M. Zhang, Y. Liu, S. Ma, and K. Zhang. Discover breaking events with popular hashtags in twitter. In *Proc. ACM International Conference on Information and Knowledge Management*, pages 1794–1798, Maui, HI, USA, October 29 - November 2 2012.

[20] T. M. Davies and M. L. Hazelton. Adaptive kernel estimation of spatial relative risk. *Statistics in Medicine*, 29(23):2423–2437, 2010.

[21] T. M. Davies, K. Jones, and M. L. Hazelton. Symmetric adaptive smoothing regimens for estimation of the spatial relative risk function. *Computational Statistics & Data Analysis*, 101:12–28, 2016.

[22] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[23] X. Dong, D. Mavroeidis, F. Calabrese, and P. Frossard. Multiscale event detection in social media. *Data Mining and Knowledge Discovery*, 29(5):1374–1405, 2015.

[24] V. A. Epanechnikov. Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications*, 14(1):153–158, 1969.

[25] L. Ertöz, M. Steinbach, and V. Kumar. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proc. SIAM International Conference on Data Mining*, pages 47–58, San Francisco, CA, USA, May 1-3 2003.

[26] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, OR, USA, August 2-4 1996.

[27] F. Farnstrom, J. Lewis, and C. Elkan. Scalability for clustering algorithms revisited. *ACM SIGKDD Explorations Newsletter*, 2(1):51–57, 2000.

[28] W. Feng, C. Zhang, W. Zhang, J. Han, J. Wang, C. Aggarwal, and J. Huang. Streamcube: hierarchical spatio-temporal hashtag clustering for event exploration over the twitter stream. In *Proc. IEEE International Conference on Data Engineering*, pages 1561–1572, Seoul, South Korea, April 13-17 2015.

[29] G. Fuchs, N. Andrienko, G. Andrienko, S. Bothe, and H. Stange. Tracing the german centennial flood in the stream of tweets: first lessons learned. In *Proc. ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information*, pages 31–38, Orlando, FL, USA, Nov 5-8 2013.

[30] C. H. E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proc. International Conference on Weblogs and Social Media*, Ann Arbor, MI, USA, June 1-4 2014.

[31] J. Gomide, A. Veloso, W. Meira Jr, V. Almeida, F. Benevenuto, F. Ferraz, and M. Teixeira. Dengue surveillance based on a computational model of spatio-temporal locality of twitter. In *Proc. ACM International Web Science Conference*, page 3, Koblenz, Germany, June 15-17 2011.

[32] T. L. Griffiths and M. Steyvers. Finding scientific topics. *National Academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[33] T. H. Grubesic and E. A. Mack. Spatio-temporal interaction of urban crime. *Journal of Quantitative Criminology*, 24(3):285–306, 2008.

[34] M. Hadjieleftheriou, G. Kollios, D. Gunopulos, and V. Tsotras. On-line discovery of dense areas in spatio-temporal databases. *Advances in Spatial and Temporal Databases*, 2750:306–324, 2003.

[35] J. A. Hartigan. Clustering algorithms (probability & mathematical statistics), 1975.

[36] M. Hassani, P. Spaus, M. M. Gaber, and T. Seidl. Density-based projected clustering of data streams. In *Proc. International Conference on Scalable Uncertainty Management*, pages 311–324, Marburg, Germany, September 17-19 2012.

[37] A. Hinneburg, D. A. Keim, et al. An efficient approach to clustering in large multimedia databases with noise. In *Proc. International Conference on Knowledge Discovery and Data Mining*, volume 98, pages 58–65, New York City, New York, USA, August 27-31 1998.

[38] M. Hoffman, F. R. Bach, and D. M. Blei. Online learning for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 856–864, 2010.

[39] C. Isaksson, M. H. Dunham, and M. Hahsler. Sostream: Self organizing density-based clustering over data stream. In *Proc. International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 264–278, Berlin, Germany, July 13-20 2012.

[40] V. S. Iyengar. On detecting space-time clusters. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Dining*, pages 587–592, Seattle, WA, USA, August 22-25 2004.

[41] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Twitter power: Tweets as electronic word of mouth. *Journal of the Association for Information Science and Technology*, 60(11):2169–2188, 2009.

[42] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *Proc. ACM WebKDD and 1st SNA-KDD 2007 workshop on Web mining and Social Network Analysis*, pages 56–65, New York, NY, USA, August 12 2007.

[43] S. J. Kazemitabar, F. Banaei-Kashani, and D. McLeod. Geostreaming in cloud. In *Proc. ACM SIGSPATIAL International Workshop on GeoStreaming*, pages 3–9, Chicago, IL, USA, November 1-4 2011.

[44] J. E. Kelsall, P. J. Diggle, et al. Kernel estimation of relative risk. *Bernoulli*, 1(1-2):3–16, 1995.

[45] F. Kling and A. Pozdnoukhov. When a city tells a story: urban topic analysis. In *Proc. ACM International Conference on Advances in Geographic Information Systems*, pages 482–485, Redondo Beach, CA, USA, November 6-9 2012.

[46] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 03 1951.

[47] M. Kulldorff. *Spatial Scan Statistics: Models, Calculations, and Applications*, pages 303–322. Boston, MA, 1999.

[48] M. Kulldorff, R. Heffernan, J. Hartman, R. Assuno, and F. Mostashari. A space-time permutation scan statistic for disease outbreak detection. *PLOS Medicine*, 2(3), 02 2005.

[49] J. H. Lau, N. Collier, and T. Baldwin. On-line trend analysis with topic models: Twitter trends detection topic model online. In *Proc. International Conference on Computational Linguistics*, pages 1519–1534, Mumbai, India, December 8-15 2012.

[50] R. Lee and K. Sumiya. Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection. In *Proc. ACM SIGSPATIAL International Workshop on Location Based Social Networks*, pages 1–10, San Jose, CA, USA, November 3-5 2010.

[51] J. Lin and H. Lin. A density-based clustering over evolving heterogeneous data stream. In *Proc. IEEE International Colloquium on Computing, Communication, Control, and Management*, volume 4, pages 275–277, Sanya, China, August 8-9 2009.

[52] J. C. Marshall and M. L. Hazelton. Boundary kernels for adaptive density estimators on regions with irregular boundaries. *Journal of Multivariate Analysis*, 101(4):949–963, 2010.

[53] D. Moulavi, P. A. Jaskowiak, R. J. Campello, A. Zimek, and J. Sander. Density-based clustering validation. In *Proc. SIAM International Conference on Data Mining*, pages 839–847, Philadelphia, PA, USA, April 24-26 2014.

[54] D. Murthy and S. A. Longwell. Twitter and disasters: The uses of twitter during the 2010 pakistan floods. *Information, Communication & Society*, 16(6):837–855, 2013.

[55] A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *Proc. International Conference on Language Resources and Evaluation*, volume 10, Valletta, Malta, May 17-23 2010.

[56] T. Pei, C. Zhou, A.-X. Zhu, B. Li, and C. Qin. Windowed nearest neighbour method for mining spatio-temporal clusters in the presence of noise. *International Journal of Geographical Information Science*, 24(6):925–948, 2010.

[57] J. Pfeffer and F. Morstatter. Geotagged twitter posts from the united states: A tweet collection to investigate representativeness. *GESIS Data Archive, Dataset*, `http://dx.doi.org/10.7802/1166`, 2016.

[58] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proc. ACM International Conference on World Wide Web*, pages 91–100, Beijing, China, April 21-25 2008.

[59] J. Ren, B. Cai, and C. Hu. Clustering over data streams based on grid density and index tree. *Journal of Convergence Information Technology*, 6(1), 2011.

[60] T. Sakai and K. Tamura. Real-time analysis application for identifying bursty local areas related to emergency topics. *SpringerPlus*, 4(1):162, 2015.

[61] M. Sofean and M. Smith. A real-time architecture for detection of diseases using social networks: design, implementation and evaluation. In *Proc. ACM Conference on Hypertext and Social Media*, pages 309–310, Milwaukee, WI, USA, June 25-28 2012.

[62] M. Song and H. Wang. Highly efficient incremental estimation of gaussian mixture models for online data stream clustering. In *Defense and Security*, pages 174–183. International Society for Optics and Photonics, 2005.

[63] K. Starbird, G. Muzny, and L. Palen. Learning from the crowd: collaborative filtering techniques for identifying on-the-ground twitterers during mass disruptions. In *Proc. International Conference on Information Systems for Crisis Response and Management*, Vancouver, Canada, April 22-25 2012.

[64] E. Steiger, J. P. Albuquerque, and A. Zipf. An advanced systematic literature review on spatiotemporal analyses of twitter data. *Transactions in Geographic Information System*, 19(6):809–834, 2015.

[65] M. Steinbach, P.-N. Tan, V. Kumar, C. Potter, S. Klooster, and A. Torregrosa. Clustering earth science data: Goals, issues and results. In *Proc. KDD Workshop on Mining Scientific Datasets*, San Francisco, CA, USA, August 26-29 2001.

[66] N. Taxi, L. Commission, et al. Tlc trip record data. *Accessed October 12*, `http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml`, 2017.

[67] R. Thomson, N. Ito, H. Suda, F. Lin, Y. Liu, R. Hayasaka, R. Isochi, and Z. Wang. Trusting tweets: The fukushima disaster and information source credibility on twitter. In *Proc. International Conference for Information Systems for Crisis Response and Management*, pages 1–10, Vancouver, Canada, April 22-25 2012.

[68] L. Wan, W. K. Ng, X. H. Dang, P. S. Yu, and K. Zhang. Density-based clustering of data streams at multiple resolutions. *ACM Transactions on Knowledge discovery from Data*, 3(3):14, 2009.

[69] M. Wand and B. Ripley. Kernsmooth: Functions for kernel smoothing for wand & jones (1995). *R Package Version*, 2:23–15, 2015.

[70] M. Wang, A. Wang, and A. Li. Mining spatial-temporal clusters from geodatabases. In *Proc. International Conference on Advanced Data Mining and Applications*, pages 263–270, Xi'an, China, August 14-16 2006.

[71] S. Wang, T. Cai, and C. F. Eick. New spatiotemporal clustering algorithms and their applications to ozone pollution. In *Proc. IEEE International Conference on Data Mining Workshops*, pages 1061–1068, Dallas, TX, USA, December 7-10 2013.

[72] J. Weng and B. Lee. Event detection in twitter. In *Proc. International Conference on Weblogs and Social Media*, Barcelona, Catalonia, Spain, July 17-21 2011.

[73] G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23(1):69–101, 1996.

[74] S. Yardi and D. Boyd. Tweeting from the town square: Measuring geographic local networks. In *Proc. International Conference on Weblogs and Social Media*, pages 194–201, Washington, DC, USA, May 23-26 2010.

[75] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Who, where, when and what: discover spatio-temporal topics for twitter users. In *Proc. ACM International Conference on Knowledge Discovery and Data Mining*, pages 605–613, Chicago, IL, USA, August 11-14 2013.

[76] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: an efficient data clustering method for very large databases. In *ACM Sigmod Record*, volume 25, pages 103–114, 1996.

[77] Y. Zhang and C. F. Eick. Novel clustering and analysis techniques for mining spatio-temporal data. In *Proc. ACM SIGSPATIAL PhD Workshop*, page 2, Dallas, TX, USA, November 4-7 2014.

[78] Y. Zhang and C. F. Eick. St-dcontour: a serial, density-contour based spatio-temporal clustering approach to cluster location streams. In *Proc. ACM SIGSPATIAL International Workshop on GeoStreaming*, page 5, San Francisco, CA, USA, October 31 - November 3 2016.

[79] Y. Zhang and C. F. Eick. St-copot: Spatio-temporal clustering with contour polygon trees. In *Proc. ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 84:1–84:4, Redondo Beach, CA, USA, November 7-10 2017.

[80] Y. Zhang and C. F. Eick. A novel two-stage system for detecting and tracking events in twitter. In *Proc. IEEE International Conference on Artificial Intelligence and Knowledge Engineering*, pages 77–84, Laguna Hills, CA, USA, September 26-28 2018.

[81] Y. Zhang, S. Wang, A. M. Aryal, and C. F. Eick. serial versus parallel: A comparison of spatio-temporal clustering approaches. In *Proc. International Symposium on Methodologies for Intelligent Systems*, pages 396–403, Warsaw, Poland, June 26-29 2017.

[82] D. Zhao and M. B. Rosson. How and why people twitter: the role that micro-blogging plays in informal communication at work. In *Proc. ACM International Conference on Supporting Group Work*, pages 243–252, Sanibel Island, FL, USA, May 10-13 2009.

[83] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *Proc. European Conference on Advances in Information Retrieval*, pages 338–349, Dublin, Ireland, April 18-21 2011.