Ecole Polytechnique

A Framework for Event Detection on Spatio-temporal Data and Case Study on NYC Taxi Data

by

Ferdinand Legros

A report on a Research Internship at
New York University Tandon School of Engineering

under the supervision of
Professor Juliana Freire

in the
Visualization and Data Analysis Laboratory

September 2016

Declaration of Authorship

I, Ferdinand Legros, declare that:

- The results presented in this report are the outcome of my own work.
- I am the author of this report.
- I have used no source or external result without citing them and referencing them in a proper bibliography.

I declare that this work cannot be suspected plagiarism.

Signed: ____

Date: ____

Abstract

Event detection in spatio-temporal data has been a growing research topic in various application fields such as environment, weather, traffic or smartphone data. Most of research articles in the field tailor a specific technique to address a single problem. The reasons why a certain category of algorithms are used over others are often unclear, and no global perspective is adopted. Our goal is to give a framework to the space-time event detection problem, and to show that elements from different techniques can be combined to fit any given problem. First we review the literature from a theoretical point of view. Then we implement two of the most prominent existing methods on urban taxi data to identify the influence of their different components.

Acknowledgements

I want to express my sincere gratitude to Professor Juliana Freire for welcoming me in the ViDA Lab and for providing me insightful advice during my internship. I also heartily thank her for the enthusiasm she showed about my project and for the dynamism she inspires.

I warmly thank Harish Doraiswamy for his patience, availability and readiness to share his experience. He helped me to adapt to the existing techniques used in the laboratory and to the work practices associated with an academic environment.

Then, I want to thank Daniel B. Neill for his big picture guidance as well as for his help in theoretical issues.

I am grateful to Fernando Chirigati for helping me defining my internship goals and for answering my questions about theory and the functioning of the laboratory.

I wish to thank Rmi Rampin for his precious technical support, I could not have carried out my experiments without his help.

Ultimately, I would like to thank all those who have helped me by their presence and support, and particularly Ann Messinger for her continuous help. I enjoyed working at ViDA and I thank all the members of the laboratory for ensuring a friendly atmosphere there.

Contents

Chapter 1

Introduction

Event detection consists in pointing out regular or irregular patterns in datasets. It includes the field of outlier or anomaly detection, which focuses on irregular events. Literature is vast on the subject for general datasets, and builds on a broad machine learning and statistics background. In the past decade, more and more techniques have been applied to spatiotemporal data. A typical research article in the field presents a technique adapted to a specific application example. No study compares the different available techniques together. Therefore, one may be in trouble when trying to create an event detection tool to be applied to a problem different from those studied in literature.

The goal of our study is to understand the problem definition of event detection and determine the specificities of two of the most prominent techniques available today. What sorts of events are detected by a particular technique? On which criteria can different techniques be compared? How robust are those methods, and how can parameter design influence their results?

Our study is both theoretical and practical. Implementation is required since one could only vaguely predict the results of a technique from the theoretical study of its algorithms. We chose to implement two techniques representative of broader event detection categories: scan statistics and clustering. The input data is urban New-York City taxi data. One instance of the dataset accounts for the aggregated density of taxi pickups and drop-offs in a predefined area around a space point, at a given time. This data is quite simple and standard, so the differences between techniques appear clearly and may not depend on the better fit between the data and a particular method.

In a first part, we will split the problem of event detection into subparts, and list the main different possibilities available for each of those subparts. Then, we will review the main state-of-the-art algorithms ordered by category. In the rest of the report, we will implement two techniques. The first is the Space-Time Permutation Model of SaTScan, the most known scan statistics technique. The second-one is based on DBSCAN, a classic clustering algorithm.

1.1 Scope specification

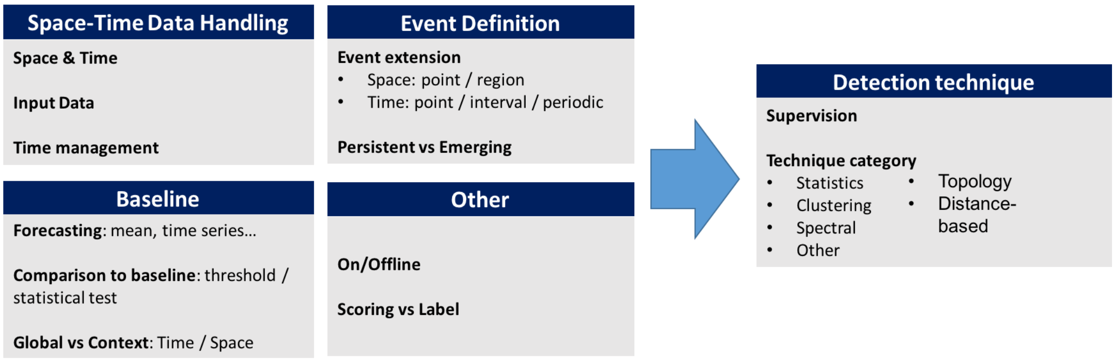
In our study, we will only consider one type of input data: space time points with one extra numerical attribute referred as numerical univariate data. This means that an instance of the dataset we consider has the format <x, y, t, v> where x,y are the spatial coordinates, t the timestamp, and v a numerical value in our study in will be taxi density. Such data is the most fundamental format that can be dealt with, that is why we chose it in order to study the problem definition of anomaly detection. Techniques handling more complex data input trajectories and graphs often build on the methods of point data anomaly detection.

Chapter 2

Subparts of a Space-Time Event Detection Problem

2.1 Problem Framework

A space-time anomaly detection problem can be split into distinct problems. Studying those independent subparts may allow one to understand the results outputted and design their own anomaly detection technique. To this date no survey on spatio-temporal anomaly detection has been done, so this work was achieved by processing various research articles. Below is the framework of our analysis, each box on the left being a major subpart of the global problem.



Space-Time Data Handling

Space & Time

Input Data

Time management

Event Definition

Event extension

- Space: point / region
- Time: point / interval / periodic

Persistent vs Emerging

Baseline

Forecasting: mean, time series…

Comparison to baseline: threshold / statistical test

Global vs Context: Time / Space

Other

On/Offline

Scoring vs Label

Detection technique

Supervision

Technique category

- Statistics
- Clustering
- Spectral
- Other
- Topology
- Distance-based

FIGURE 2.1: Event detection problem framework

We will now focus on each subproblem and highlight solutions that can be found in literature.

2.2 Space-Time Data Handling

2.2.1 Space and Time handling

Anomaly detection may take into account space dependencies, temporal dependencies or both. Techniques taking into account both space and time often are an extension of a space or a time anomaly detection technique. We focus on space-time techniques. However, it should be noted that temporal techniques can be used for parallel monitoring. That is detecting anomalies at every spatial point of the data set independently. Such approach is adopted in [1] for traffic data, in [2] for environment data and in [3] for sensor data. Those techniques have lower detection power because they examine spatial points individually, but they are less computationally expensive.

2.2.2 Time Management

For space-time techniques, time can be handled in different ways. i.) One option is to perform spatial analysis at each time step, and then compare time steps with each other. In [4], Wu et al. compares the spatial region anomalies detected in consecutive time steps. Doraiswamy et al. in [5] compares space topological profiles of all time steps between each other, which allows to also detect events with a non-continuous timespan. Generally speaking, any spatial anomaly detection technique could be used this way to create a space-time technique. ii.) As pointed out in [6], time can be treated as an additional dimension of space. This is the case of SaTScan, a prominent scan statistics technique [7]. For spatial analysis, SaTScan performs a scan of 2D regions on the monitored area. For space-time analysis, the scan is performed on 3D regions. Iii.) A less standardized time management technique is the use of time neighborhoods.

2.2.3 Data Input

The three main space-time data inputs are point or grid data, trajectory data, and graph data. For space-time data, graph connectivity may represent spatial distance, or a spatial network such as a road or a water network for instance. Many specific techniques are developed for graph and trajectory data. Description of those different data types can be found for instance in [8].

2.3 Event Definition

The event definition determines which type of anomaly will be detected.

2.3.1 Event Extension

The space extension of the event may be a single location or a region. Region detection techniques are less sensitive to grid resolution for space time grid data - and are assumed to detect subtler events than point detection techniques. Indeed, a group of slightly anomalous points can be detected as an anomalous region, while every single instance appears quite normal.

The time extension of the event may be a single time step, an interval or an irregular time steps set. For example, SaTScan defines an event as an anomalous region spanning over an interval of time. Birant et al. in [9] look for anomalies which are punctual both in space and time. Chawla et al. in [10] output anomalous single spatial points which spans over an interval of time.

2.3.2 Event Persistence

A technique may define an event as persistent or emerging. A persistent event for numerical data is a shift in values which lasts in time, and may disappear after some duration. This is the standard definition adopted in literature. An emerging event consists in a gradual increase in values. Such a form of event is addressed for instance in [11].

(A) Persistent

(B) Emerging

Persistence is also a property of the shape of the events detected. A persistent shape events keeps the same shape during its timespan. In [4], Wu et al. track moving anomalies, which is particularly adapted to weather phenomena. An event can also be fixed but shrinking or expanding.

It should be noted that a persistent technique is a good first approach, since it may still detect though maybe inaccurately emerging events.

2.4 Baseline

In order to detect anomalies, a definition of normal behavior must be adopted, even if it is very simple. This is what we will call **baseline**. The baseline is a crucial point of the detection technique, since it indirectly defines the anomalous behaviors that will be detected.

2.4.1 Forecasting Model

Most baselines are computed using forecasting models. Various forecasting techniques can be used. Basic ones include considering the mean, or a weighted mean taking into accounts effects such as day-of-week. One can perform time series analysis, this is what Guo et al. use to compute expected values for traffic data in [1]. Zhang et al. also retort to time series forecast to detect space-point anomalies in wireless sensor data in [12].

Another possibility is to fit a statistical model to represent regular data. The standard model for counts data i.e. positive numerical values is the Poison model. For instance, Kulldorff used it in the original version of SaTScan in [13].

2.4.2 Method of comparison to baseline

Once the baseline is computed, values of candidate anomalous points can be compared to it in different ways. i.) The simplest option is to examine the ratio or the difference of point value and baseline, and define a threshold over which the point is considered anomalous. This is what is used in [2, 5, 10, 14]. A common threshold is 3 standard deviations of the set of values taken by the spatial point. ii.) A statistical test can be performed in order to assess whether the difference between value and baseline is significant, as in [13, 15]. This method gives a metric of anomaly significance.

2.4.3 Context

Baseline can be global or local. A local baseline may take into account spatial and/or time neighborhood. This allows the event detection technique to spot anomalies which would not have been detected while being compared to the whole dataset. Authors of [11, 13] fit a Poison model to all of the data except the suspected anomalous region, so the baseline is global. Authors of [15–18] take into account a space-time neighborhood around the suspected points. Wu et al in [4] performs a spatial scan statistic at each time step independently, so her approach is time contextual the time neighborhood being restricted to the current time step but spatially global.

Literature also points to research in space-time neighborhood discovery, see [19, 20]. Discovering neighborhoods which fit the structure of the data may allow to use better calibrated context for baseline computing.

2.5 Other criteria

Two additional criteria should be mentioned. They do not alter the nature of the problem dramatically, but could be taken into account in particular cases.

2.5.1 On/Offline

Input data can be a stream of data or a fixed dataset. Processing streaming data adds computational constraints. It is common in the analysis of sensor data. Online techniques can consist in offline techniques adapted to streaming data. For instance in [21] Principal Component Analysis classically used for offline anomaly detection is adapted so that it can be updated with the stream.

2.5.2 Anomalousness score

Anomaly detection techniques can either output labels normal or anomlaous - or anomalousness scores. Label techniques can often be considered as scoring techniques since they label suspected anomalous points when a certain metric is above a predefined threshold.

2.6 Combining independent subparts

The author of a research article in space-time anomaly detection typically presents a fully built technique and implements it on test data. Having the independent subparts that we distinguished in mind, one could adapt a technique to its own needs. For instance, a global baseline could be changed for a local one; a technique that studies adjacent time frames could be changed to study time steps of only the same days of the week. Therefore, it is crucial to understand the impact of each solution of the sub-problems we highlighted in order to combine them efficiently. This is one of the purpose of the implementation task that we led.

Chapter 3

Spatio-temporal Event Detection Categories

The approaches adopted to address the problems identified above can be classified into broad categories. Several classifications of general anomaly detection can be found in [22–24]. Among the categories pointed out in those surveys, some do not apply to space time data. We highlight the most prominent categories applied to point numerical univariate spatio-temporal data.

Supervision

We consider only unsupervised techniques. Indeed, we assume that the events present in the dataset are not previously known. This is not a significant restriction because almost all techniques in our scope are unsupervised.

3.1 Statistical-based methods

Statistical based methods fit a model to the input data to represent the normal behavior. Then they compare points values to the baseline with a statistical test. The assumption of statistical-based techniques is that normal instances occur with high probability and anomalous instance occur with low probability. One of the most popular tests used is the Likelihood Ratio Test, detailed in [25] and used in [11, 13, 15]. For statistical testing, the number of tests performed is large: one for every candidate point or region anomalies. So a simple test process would encounter a lot of false positives simply by chance this is called multiple testing. That is why replication and empirical p-values can be used to ensure the statistical significance of the tests performed.

The most prominent type of statistical-based method is scan statistics. It was popularized in [13], and was widely applied to a great variety of use cases because of the availability of the SaTScan executable software at [7]. Scan statistics assess the anomalousness of 3d space-time regions. Different region shapes are looked for in literature. SaTScan looks for circle and ellipse shapes. Neill et al in [26] and Pang et al in [11] look for rectangles in grid data. Irregular region searching is addressed by Tango et al. in [27].

Regions may be scanned following a brute-force approach, but more efficient scanning was developed. Pang et al. in [11] develop a pruning strategy that discards irrelevant regions. In [28], Neill et al. introduce the Linear Time Subset Scanning property. It states that under certain assumptions, the most anomalous region of a dataset is a sublist $[r_1, r_2, ..., r_k]$ for a certain k, where r_i are the instances sorted according a certain priority function. This property is usually not valid on whole real life datasets, but it can be used at different steps of the scanning algorithm to reduce computation time.

Most of the literature focuses on parametric models, but non parametric also exist, see for instance [29].

The advantage of statistical techniques is that they provide metrics to assess the anomalousness of the events outputted. The main limit is that modelling normal behavior is often approximate.

3.2 Clustering-based techniques

A clustering-based space-time anomaly detection technique is composed of two parts. First, a clustering algorithm is performed. Then, anomalousness is assessed among anomalous candidates.

3.2.1 Clustering Task

For an overview of space-time clustering, see [6, 8]. The most commonly used algorithms for space-time clustering are density based algorithms, in particular DBSCAN due to its efficiency and limited number of pre-set parameters. It was adopted to space-time data in [9]. In [30] another variant named ST-LDBCAN is developed. The DB-SMOT algorithm is presented in [31].

3.2.2 Assessing Anomalousness

There are three main ways to search through clusters for anomalies.
i.) The anomaly candidates can be the unclustered points. This approach usually limits the detection to individual points no spatial region event can be detected. This is the case of [32] and [30]. ii.) The candidates can be the border points inside clusters.
iii.) The candidates can be the clusters themselves. It allows the detection of region events. In [14], mean values of outputted clusters are examined and the most extreme clusters are outputted as anomalies. Telang et al in [15] perform statistical testing to determine whether the homogeneous clusters they compute are anomalous.

The pro of clustering techniques is that they are easy to implement, and they can be combined with different methods to assess anomalousness of candidate points or regions. The main drawback is that techniques which consider unclustered points detect anomalies as a by product of the clustering algorithm, so they are not optimized for anomaly detection.

3.3 Spectral Techniques

By spectral techniques, we mean the use of Principal Component Analysis. These techniques were first applied to computer network data, see [33]. They are limited to detection of anomalies that last during a time interval but are restricted to a single space point, or anomalies consisting of the whole space during a single time frame. The idea behind using PCA for anomaly detection is that principal components captures regular or anomalous behavior of data. Principal components are ranked according to their associated eigenvalue. It is assumed that the k first components capture the normal behavior, and that the following ones reflect anomalous behavior. In practice, k is set between 2 and 4. Data instances are projected on the vector space generated by anomalous eigenvectors. If projection is over a certain threshold, the instance is considered to be anomalous.

To build the input matrix for PCA, such methods consider a matrix L whose rows are spatial locations, and whose columns correspond to time steps. To detect space-time anomalies, preprocessing on this matrix is necessary. $L \times L^\top$ and $L^\top \times L$ can be considered, see [10]. However, Brauckhoff et al. explain in [34] the trouble that comes from improper preprocessing and show that the only satisfying preprocessing is applying the Karhunen-Loeve transform. Wavelet transform can also be used as a preprocessing step, for instance in [35].

PCA was applied to traffic data in [10, 35, 36]. It is used in a basic fashion to environment data in [37]. Incremental PCA developed in [21] made the technique applicable to streaming data.

The main con of PCA is that the parameter k and the threshold for projection on anomalous space are set manually.

3.4 Topology

Topology can be used to analyze spatial topological profiles. Doraiswamy et al. in [5] compare profiles of different time steps, and label matching extremes regions as events. Topology is also used by Franke et al. to analyze the topological profile of the anomalousness scores of all points in [3]. A great pro of topology based techniques is that they are computationally efficient.

3.5 Distance-based Methods

Distance-based methods are restricted to space-time point anomalies. They consider the neighbors of the candidate anomalous point according to some distance. Depending on how far the point is from its neighbors, it is labeled normal or anomalous. Since a point is compared to its neighbors, those techniques often output contextual anomalies. Such techniques are implemented in [16–18].

3.6 Additional Techniques

Some additional techniques are worth mentioning. Pure visualization techniques are not part of our scope, they can be studied in [38]. Albanese et al. present a new rough set based space-time point anomaly detection technique in [39]. Periodic event mining is another branch of the literature.

3.7 Method Testing

From an implementation point of view, the efficiency of a technique is typically assessed in four main ways. First, one could compare the events outputted to ground truth events found in the news or in reports. Second, events highlighted by a technique can be compared to the ones outputted by other methods, as in [15, 35]. A common way to determine performance metrics on a method is to apply it to controlled semi-synthetic data. A synthesized event such as a region with anomalous high counts - is injected in a real life dataset. Neill et al. inject models of simulated disease outbreak in datasets in [26]. The last method used in literature is user evaluation. Telang et al. in [15] ask users to rate anomalies outputted by their technique depending on their analysis of the counts and their visual insight.

Chapter 4

Implementation Framework

4.1 Scope

Now that we have studied how an event detection problem can be split and what the different detection categories are, let us study the results of two prominent techniques of the literature. We selected **region** event detection techniques, for the reasons given in the problem definition part.

Both of the techniques we implemented are representative of the anomaly detection category they belong to. The first one is the Space-Time Permutation Model of SaTScan, the most prominent scan statistics model. An executable can be found at [7]. SaTScan was primarily designed for health monitoring tasks in particular disease outbreak detection but was extensively applied to multiple domains such as policy, crime analysis or environmental data. The second is the method developed by Telang et al., which is clustering based. It is described and implemented on weather data in [15].

We implement and study the techniques as is. Our study gives an idea of the results that can be obtained with them. Hopefully it may guide one with an event detection problem willing to implement an existing method. It may also help to develop a new technique, because we illustrate the impact of the elements used in each technique. Thus, one can understand for instance what the effect of using a particular type of baseline is, or the differences between a clustering method versus a statistical one.

4.2 Dataset Description

The dataset is October 2011 New York City taxi GPS data. For an overview of existing anomaly detection techniques for traffic data, see [40]. Raw data consists in the time and location longitude and latitude - of pickups and drop-offs of all taxis in New York City. We transform this raw data into fixed point time series data, in different ways in the two experiments. The experiment on SaTScan will be referred as experiment A, the one on Telangs clustering method as experiment B.

We define a set of spatial points in NYC. Set A includes all road intersections in Manhattan. Set B is composed of the centers of every cell of a square grid data. The grid extends from the south extremity of Manhattan to the north of Central Park. We study those regions during the whole month of October 2011. We chose this month because we know that particular events occur during it, and we can see if and how they are detected. The time resolution of the dataset is one hour. It is precise enough to detect anomalies during a day, but still allows for fast computation.

At each time step, we compute the aggregated density of pickups and drop-offs at every single spatial point. We chose to consider both pickups and drop-offs in order to capture the maximum information available on taxi activity. Taxi data is heterogeneous and noisy. So, we then compute a density function which acts as a smoother. In experiment A, for every point of the set of spatial points, pickups and drop-offs included in an ellipse centered in the point, of semi-minor axis 35m and semi-major axis 45m are aggregated. The points included in a rectangle centered in the point of width 270m and height 200m are added with a decreasing exponential weight based on the distance to the ellipse. Let us note that this leads to multiple counting. However, it provides an efficient smoother. For details on the density function see [5]. In experiment B, we use different grid resolutions. For a 50m x 50m grid, we use the same density function as experiment A. For other grid resolutions, we use a similar density function. We just increase all the dimensions ellipse, rectangle proportionally to the size of the grid.

4.3 Parameter Design

For both experiments, we analyze the results outputted by different sets of parameters. We are interested in the following problems: i.) what is the range of events detected with those different parameters? ii.) are the results consistent? Those two problems are not redundant. For instance, when changing parameters, a method may detect new events while still detecting a subset of the ones detected with the previous parameters. So it detects a wide range of events with various parameters but still remains consistent.

Chapter 5

SaTScan Space-Time Permutation Model Analysis

5.1 Executive Summary

SaTScan Space-Time Permutation (STP) model consists in detecting regions whose cells show a density significantly different from a baseline pre-computed from the **whole data aggregated**.

STP technique mostly detects **high density prominent regular patterns** of the city. It is a relevant tool to detect day-of-week and nightlife patterns in particular. This is mainly due to the fact that expected behavior is computed over the whole month.

STP model can be constrained so that is detects clusters under a certain spatial size and timespan bound. We observe an **additive bias**: detected clusters tend to extend up to the spatial bound, and up to the time bound when time bound $<$ 7h.

The algorithm allows to search for **ellipse anomalies** which better textbffit the road network than circles.

Finally, the results are **mostly stable** for different grid resolutions.

5.2 Theoretical Model

We present briefly the STP model. For details see [41].

5.2.1 Region Scanning

Original STP looks for cylinder shape regions. The basis of the cylinder is a circle including multiple spatial locations at a single time step. The height of the cylinder corresponds to the time interval considered. STP uses a greedy approach to scan for those cylinders. Successively every spatial point is taken as the center of the circle basis of the cylinder, and cylinders of all possible spatial radiuses and temporal heights are considered. This approach is very computationally expensive. Moreover, in practice we are not interested in events which are so large that we cannot interpret them. That is why we use spatial and temporal bounds to compute the cylinders.

5.2.2 Baseline Computing

We will refer as the total aggregated density over all spatial points and all time steps as C. $C = \sum_z \sum_d c_{zd}$ where c_{zd} is the density observed at location z at time d. For each space-time point, the expected density is computed as $u_{zd} = (1/C)(\sum_z c_{zd})(\sum_d c_{zd})$. The expected aggregated density is computed as $u_A = \sum_{(z,d)\in A} u_{zd}$. The underlying assumption when calculating these expected densities is that the probability of observing a given density in location z, given that it was observed on time d, is the same for all times d. Every observation that goes significantly against this assumption is considered anomalous. This element is key to understand the results that we obtained when implemented the technique.

5.2.3 Anomalousness Assessment

STP assesses for anomalousness of all space-time cylinders of the dataset. When both $\sum_{z\in A} c_{zd}$ and $\sum_{d\in A} c_{zd}$ are small compared to C, c_A is approximately Poisson distributed with mean u_A [42]. This is the case for all anomalies whose timespans and space extensions are small compared to the whole aggregated density. STP builds on that approximation and uses the Poisson generalized likelihood ratio as a metric of anomalousness for the cylinder A $(c_A/u_A)^{c_A} \times ((C - c_A)/(C - u_A))^{C-c_A}$. We will refer to this value as the **anomalousness score** of the cylinder. The higher the score, the more likely anomalous the cylinder. Let us note that this baseline is global for both spatial and time attributes - it does not take into account space-time context.

5.2.4 Preventing Multiple Testing

Since the number of anomalousness scores computed is very high, obtaining a high score by pure chance may happen. To prevent this multiple testing effect from disrupting the outcome of the algorithm, STP uses replication. One replication consists in shuffling all location information and timestamps of density points, and then computing the most anomalous cluster of the new dataset. The score of an anomalous candidate cluster obtained on the original dataset is compared to the scores of the most likely cluster of all the replications. Statistical significance is deduced from this comparison. For instance, a cluster whose score appears in the top 0.1

In our study, we only ran replications in preliminary studies which are not reported here. Indeed, the scores of the clusters reported for the original dataset were always ranked first in 999 replication scores. The studied anomalies scores were never below 500, while top replicated scores were around 20-30. The reason for this is that the events that we observe are very striking, they involve densities which peak at 5 times baseline density, or go as low as 5% of baseline. The significance test was rather thought to apply to subtler events such as an early disease outbreak.

Why this technique?

SaTScan is a region anomaly detection technique. It is one of the most - if not the most - popular region spatio-temporal anomaly detection technique used. It is representative of the statistical-based anomaly detection category.

SaTScan is declined in several variants. We specifically use the Space Time Permutation model because it is adapted to univariate density data and it requires only densities and no complementary data contrary to other available models. Moreover, contrary to many parametric statistics techniques, it does not assume directly that the normal behavior follows a theoretical model such as Poisson.

5.3 Building an Iterative SaTScan

No source code is available for SaTScan, one must do with the options implemented in the available software. In particular, SaTScan may or may not report two clusters depending on their spatial overlap. Practically speaking, it cannot report two anomalies which occurred at the same spatial location but at different times.

That is why we built a tool which runs SaTScan in an iterative fashion in order to report all the clusters detected. At each iteration, the most anomalous cluster is reported, and the density data is updated so that the influence of this cluster is removed from the dataset. We do this by modifying the densities of each point of the cluster so that they equal the baseline calculated at the next iteration.

Such process is computationally expensive. The original SaTScan computes all anomalous clusters once and reports a certain number k of clusters. Instead, we launch the whole algorithm at every single iteration to output the most anomalous cluster, so computation time is multiplied by k. This was not a hindrance to our study since we are primarily interested in the nature of the events outputted, computation is not a core issue.

SaTScan provides visualization tools that we adapted to our approach. We use Google Earth to visualize the results.

5.4 Dataset Preprocessing

We use the dataset A - see 4.1. The data input format of the software is rather standard and does not require complex preprocessing. Each instance of the input dataset should specify the location, the timestamp and the density observed. That is why we could easily use road intersections as spatial locations input.

5.5 Parameter Design

The parameters of Space-Time Permutation model are: i.) shape of anomalies looked for - circles or ellipses ii.) Scanning for high density and/or low density anomalies, iii.) Time bound of anomalies - $<$ 3h for instance, iv.) Space bound for anomalies - for ellipses the constraint is applied to the semi-minor axis.

We looked for ellipse shape anomalous cluster. We do not report the results obtained with circular clusters here. In fact, ellipse clusters can elongate and so better fit the road network. One could even set the constraint on the ellipse - strong, medium, none - to penalize or not the elongation of the ellipse. We implemented all constraints. The time bounds set is {3h, 7h, 24h}. The space bounds set is {49m, 123m, 245m, 613m}. The scan with no space bound or no time bound was too computationally expensive to perform. What is more, the results obtained with large bounds led us to think that unbounded results would not have been relevant. From now we may refer to space size bound as **sb** and to time size bound as **tb**.

5.6 Events features analysis

5.6.1 Shape

Among the three ellipse variants, the most adapted to the problem is the unconstrained ellipse. First, it best elongates to fit roads. Second, it is computationally more efficient. Applying no constraint speeds up computation by a factor of roughly 2 to 4.

5.6.2 Size

5.6.2.1 Additive bias

We note that for our dataset, the higher the space bound, the bigger the cluster. This is what we will refer as the **additive bias** of the algorithm on our data. Let us examine visually how the space size of clusters vary with the space bound. We show below the juxtaposition of the top 20 clusters obtained for all space bounds - for 613m we only considered the top 10 - with a time bound equal to 7h.

The colors [red, yellow, green, blue] correspond to the space bounds [49m, 123m, 245m, 613m]. We observe that clusters computed with low size bounds are always smaller than the clusters computed with high bounds. This may seem surprising. Indeed, if one street is blocked for instance, then the anomaly should be detected on a strictly delimited area, and should be detected in the same location by all algorithms with a sufficiently high space size bound. However, the events detected here are not as obvious as this. Those rather come from vast region with unexpected density. The whole Lower East Side and East Village districts - bottom right of the picture - are highlighted as anomalous zones during weekend nights. Those districts are the most prominent areas for nightlife in NYC. Since the high taxi density is spread over the whole area, the clusters tend to capture the most of it, that is why they extend to the maximum size allowed by the bound.

The visual analysis of clusters is confirmed by the graph representing how average cluster semi-major axis vary with semi-minor axis bound. Here we only considered the top 10 anomalous cluster for all size bounds and we aggregated results obtained with time bounds 3h and 7h.

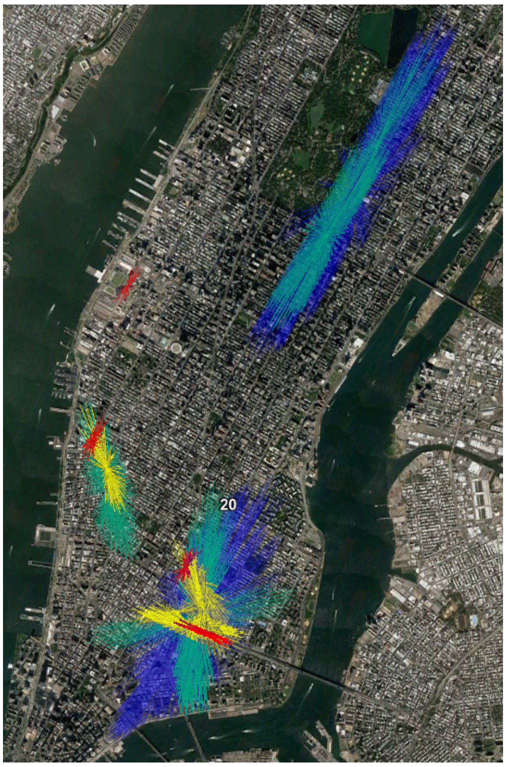


Figure 5.1: Top 20 juxtaposed events, various space bounds

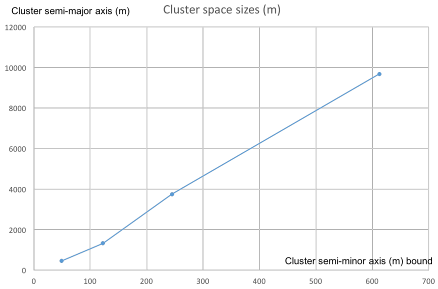


Figure 5.2: Cluster size with size bound

Let us note that the average elongation of clusters - corresponding to the semi-major axis - is close to proportional to the semi-minor axis bound set. It is coherent with the idea that the ellipses tend to extend to the maximum elongation tolerated for the semi-minor bound set.

5.6.2.2 Theoretical source of the additive bias

The additive bias is due to the fact that the baseline computation assumes that locations have the same behavior over the whole period studied - that is to say at all hours during all days of the month studied. This is obviously not the case in reality, since taxi data show strong day-of-week and hour-of-day trends. So a great proportion of space-time cells have a density significantly different from their baseline. That is why adding them to an anomalous cluster results in rising the anomalousness score.

5.6.2.3 Use cases

Now, let us examine what types of events are detected with different size bounds. We distinguish two main use cases.

Tendencies - High bound

Setting a high space bound is useful to detect large tendencies. In Fig. 16 for instance, we noticed that the algorithm with sb = 613m - results showed in blue - detects the whole area $\{\text{East Village} \cup \text{Lower East Side}\}$ as one anomalous clusters. The timespan of the cluster is 22:00 to 04:59 on Oct. 22, a Saturday night. It highlights the global activity of the most active part of the city during the busiest hours of night.

Localized patterns - Low bound

Decreasing the size bound brings about two effects.

i.) Smaller clusters concentrate on the heart of anomalous regions. For instance, consider the two red - sb = 49m - clusters at the bottom right of the picture. The one at the bottom has the same date and time span as the blue - sb = 613m - outputted. It is concentrated on Houston Street, the busiest street of Lower East Side. A large event could also be split into several clusters. We do not observe this with our top 20 cluster approach, but we would probably notice it if we increased the number of clusters reported.

ii.) New events appear. With a small size bound, the small anomalies are no more overshadowed by the big clusters whose test score benefit from the additive bias. Below example is the juxtaposition of the top 40 clusters obtained with sb = 123m, st = 7h and sb = 245m, st = 7h. Clusters obtained with sb = 245m are represented in green. Among sb = 123m clusters, we find

- Events already obtained with lower resolutions, in yellow.
- New events
  - The street blockage resulting from the Halloween Parade, on Oct. 31 between 18:00 and 22:59, with taxi density equal to 0 - in red.
  - An anomalously high activity with taxi density equal to 5.65 the expected density next to the end of the High Line, a popular walk, on Oct. 16 - a weekend day - between 8:00 and 14:00 - in orange.

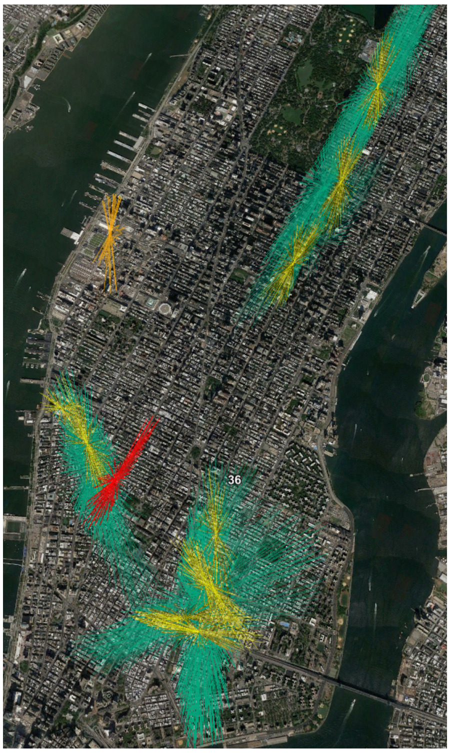


Figure 5.3: Appearing events

Thus, for data similar to ours - where baseline assumptions are not respected - one should set the spatial size to the approximate size of the events they are interested in. Running the algorithm with various space size is necessary to explore the events of a dataset.

5.6.3 Time

Over all the experiments, we observe a temporal additive bias for time bounds below 7h only. Below is plotted the average time extension of the top 10 anomalies with the time extension bound of the clusters, for each space size bound.

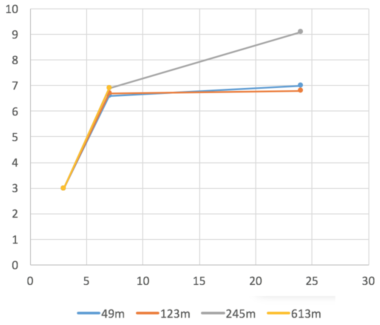


Figure 5.4: Time extension with time bound

We observe that for all space bounds, time bound almost equals average timespan for st in 3h, 7h. When time bound is set at 24h, average time span stays at 7h or only moderately increases. What happens is that experiments with st in 7h, 24h share most of their top 10 anomalies. That is due to the fact that the period during which the ratio observed / expected density is high during a day rarely exceeds 7h - think of morning rush hours, or a busy night. Some exceptions occur, for instance cluster number 5 for sb = 123m, st = 24h is 24-hour long on a Saturday. It is located in a part of mid town which is busy the week, so the algorithm detects the whole quiet week-end day as an anomaly.

Let us specify that events detected are consistent for multiple time bounds. With a smaller time bound, clusters concentrate on the most anomalous subpart of the event detected with a higher bound, as it was the case when studying spatial bounds.

5.6.4 Anomalous Densities Analysis

The ratio of the number of high density anomalies to the total number of anomalies for most experiences is > 90%, and ratio never goes below 70%.

So STP seems biased towards **high density anomalies**. This can be understood examining the score function used by the algorithm.

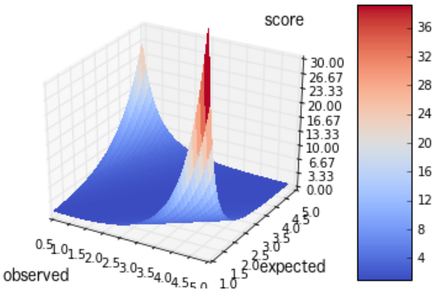


Figure 5.5: STP Score function

Here is represented the value of the test statistic given observed density and expected density. Total density was set to 1000, but other values result in the same picture. We set observed and expected densities ranges so that we obtain ratios observed / expected similar to our dataset. The minimum ratio is 1/10 - top left corner of the grid, the maximum ratio is 5 - bottom right ratio of the grid. We observe that test score is more sensitive to high density ratios. A ratio of 5 is commonly detected in our experiments, this is the case of the busy nightlife events detected.

5.6.5 NYC Events Exploration and Stability

In almost all experiments a few prominent event types occupy most of the top 40 anomalies, because they are detected multiple times. For instance, a busy night in an active district can be detected all week end nights of the month. 90% of the top 40 events belong to three main types. To illustrate this, let us examine the picture of all juxtaposed clusters obtained with all space bounds and a time bound equal to 7h. We chose 7h because it is the most versatile time bound. Colors are the same as 5.1.

We identify three main types of events.

i.) Busy week-end nightlife in Lower East Side and East Village - referred as A1. Usually spans from 22:00 to 4:59. Ratio observed density / expected density 4.5

ii.) Busy week-end nightlife in West Village - A2. Usually spans from 22:00 to 4:59. Ratio observed / expected 3.5

iii.) Quiet week-end night in East Mid Town and Upper East Side - A3. Usually spans from 22:00 to 4:59. Ratio observed / expected 0.3. This event is harder to interpret. Upper East Side and East Midtown contribute greatly to the total density of the month. The computing of the baseline assumes that this relative weight of the region is always the same. So, at the busiest hours of the city - during nightlife - the algorithm expects taxi density to rise dramatically in those regions. It is not the case, since nightlife is concentrated in downtown NYC, so an anomaly is detected.

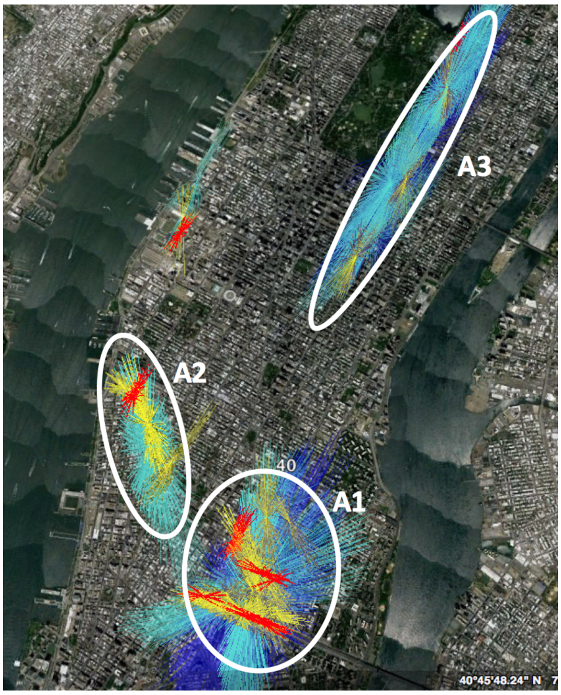


FIGURE 5.6: Event exploration

The times and dates of those events are mostly consistent for the different space bounds used. Sometimes the different instances of an event type - i.e. the different Saturday nights in East Village - have a slightly different order in the top 40.

5.7 Computation

For a total number of locations L, a total number of time steps T, a maximum space location - the maximum number of locations in a cluster - bound l and maximum time duration bound t, one iteration of SaTScan has complexity $O((L \times l)(T \times t))$ For details see [43]. We remind that we set the number of replications to 0.

Below are the computation times for single iterative of the algorithm.

We observe the dependency in $T \times t$ by noticing the vertical shift of the curve when time bound increases. The dependency in $L \times l$ is harder to observe since the spatial bound is specified in meters, there is no actual space bound counted in number of locations in the cluster.

5.8 Possible Improvements

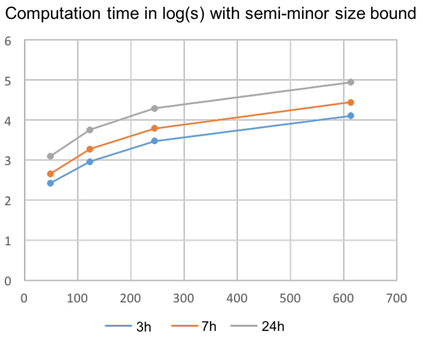


Figure 5.7: Computation times with various space and time bounds

First, we remind that the unavailability of the source code made us choose a computationally expensive method. Using the algorithm provided as an executable is still a satisfactory option if one does not want to detect anomalies at the same location for different times.

Second, the main improvement that could be made to the method is the baseline computing. Instead of aggregating densities over the whole month, one could take into account day-of-week effect. This would allow to focus on events such as parades or concerts for instance.

Chapter 6

Telang et al. Clustering-based Model Analysis

6.1 Executive Summary

Telangs technique consists in i.) grouping similar space-time cells in homogeneous clusters and ii.) detect which of those clusters are significantly different from their space-time neighborhood.

Telang method mostly detects **high density prominent regular patterns** of the city. It is a relevant tool to detect day-of-week and nightlife patterns in particular. This is mainly due to the choice of a local baseline. A candidate anomalous region is compared to its adjacent space-time neighborhood.

The main parameter - the gini threshold - can be set so that the method detects either **broad tendencies** - a busy district for instance - or **localized patterns** - a particular street activity. The time span of anomalies remains low in general (<7 hours).

The use of such clustering allows to detect **irregularly shaped anomalies**, such as a portion of road, or a region with a hole.

Finally, results are **mostly stable** for different grid resolutions.

6.2 Theoretical Model

We present briefly the method developed by Telang et al. in [15], see the full article for details. The method can be split into two parts: cluster formation and anomaly assessment of clusters.

6.2.1 Cluster Formation

The cluster algorithm used is close to DBSCAN on grid cell data. From an arbitrary unclustered cell, the 3x3 cells around are considered to potentially grow the cluster. A neighboring cell is added if the dispersion of the new cluster - measured by the gini index of the current cluster - remains below a certain pre-defined threshold. The gini index of a cluster is calculated as

$$gini(X_1, ..., X_N) = \frac{N+1}{N-1} - \frac{2}{N(N-1)u}(\sum_i P_i X_i)$$

where N is the number of data points in the cluster, u is the mean of the distribution and Pi is the rank of the data point after sorting the points in the cluster. To be accurate, cells are not added one by one to a growing cluster. One constraint is that the event is persistent, so if a spatial location is included in the cluster, all the time steps of the clusters for this location must be part of the cluster. From the current cluster, the algorithm considers the following cells sets to be added

- the same spatial locations at the first time step out before the cluster
- the same spatial locations at the first time step out after the cluster
- one adjacent space cell at every time steps included in the cluster

When no more cells can be added to the cluster while maintaining the gini index under the threshold, the algorithm starts another cluster from an arbitrary cell.

The value of the gini index threshold determines how homogeneous clusters are. Low gini thresholds will lead to clusters with homogeneous values. High gini thresholds will lead to clusters with dispersed values.

6.2.2 Anomaly Assessment of Clusters

Computing Neighborhood of cluster For every cluster, the method takes into account their space-time neighborhoods. For a given cell of spatial row index i_0, spatial column index j_0, timestamp t_0, the neighborhood is composed of cells with i in $[i_0 - r, i_0 + r]$, j in $[j_0 - r, j_0 + r]$, t in $[t_0 - r, t_0 + r]$, with r being the **range** of the neighborhood. To build the neighborhood of a cluster, the neighborhoods of every cell of this cluster are added to the neighbors set, and then cluster cells are removed from this set.

Statistical testing The method assesses the anomalousness of computed clusters using a statistical Likelihood Ratio Test. The alternative hypothesis is that the cluster has an underlying Poisson distribution $P(\lambda_r)$, and that the neighborhood of the cluster - cluster excluded - has a distribution $P(\lambda_n)$ with $\lambda_r \neq \lambda_n$. λ_r and λ_n are obtained by maximum likelihood estimation - in practice it is the mean of the values of the respective distributions. The null hypothesis is that cluster U neighborhood follows a common Poisson distribution, with the maximum likelihood estimate parameter λ_g. The likelihood ratio considered is

$$-2ln(\frac{likelihood\ for\ null\ model}{likelihood\ for\ alternative\ model})$$

In the original method developed by Telang et al., this ratio is then compared to the chi square value corresponding to the desired statistical significance. In our study, as for

SaTScan, the ratios of the top 40 clusters obtained in preliminary studies are far more significant than the 5% significance usually considered.

Why this technique?
First, we implemented this technique because it fulfills our criteria of detecting region anomalies. Second, it uses a variation of DBSCAN, which is one of the most popular clustering algorithm applied to spatial data. Other clustering-based space-time anomaly detection methods are likely to use similar clustering algorithms. Then, it is not designed for a specific use but rather addresses a fundamental problem: detecting persistent events in grid data. So it could be easily adapted to other applications. Finally, it is ready to use with no need of implementing additional anomaly detection modules.

6.3 Dataset Preprocessing

We use grid data defined as Set B - see 4.1. We implement various grid resolutions: 70m, 100m, 140m, 190m, 250m, 360m. The grid data only contains cells fully included in Manhattan. The presence of cells fully or partially outside Manhattan brought about troubles in the results. The method would detect anomalous low density regions composed of those cells. Indeed, their densities close to zero - not equal to zero due to the smoother - differ significantly from their spatial adjacent neighbors inside Manhattan which show regular Manhattan activity.

Let us note that this is an important feature of the technique. Cells which are always at a very low count will influence the detection of anomalies around them. Their regular neighbors may be pointed out as high counts anomalies. For instance, this phenomenon affects cells adjacent to Central Park. Therefore, regions adjacent to Central Park most likely have an artificially high test statistic. Since those regions do not appear in the top anomalies that we study, we did not remove Central Park cells. Still, such preprocessing step should be kept in mind when implementing Telang et al. method.

6.4 Parameter and experiment design

In all our experiments on Telang et al. method, we use a fixed neighborhood range equal to 2 cells. This is the range used by Telang et al. We replicate experiments with various gini thresholds: 0.01, 0.03, 0.1, 0.3. Telang et al. set the threshold to 0.01 in their study. Our data - taxi density - is more heterogeneous that theirs - average temperature. That is why we implemented higher gini thresholds. **Notation: from now, we will refer to grid resolution as r, and gini threshold as g**.

To analyze the features of the technique, we will examine on the one hand indicators related to core characteristics of anomalies: spatial size, timespan, test statistic score, anomalous cells counts. On the other hand, we will lead a visual analysis of the outputted anomalies. We will focus on the top 40 anomalies - that is the 40 anomalies with the highest test statistics score. This allows to focus on prominent anomalies while giving an idea of the range of events outputted by a given parameter set.

6.5 Events Features Analysis

6.5.0.1 Space size

Additive bias

Two cluster sizes matter in this method. The first one is the average size of all homogeneous clusters created during the clustering step. This value quantifies how homogeneous the data is. The second one is the average size of the clusters detected as anomalies. It gives a notion of the homogeneity of anomalies. We report those two average sizes for a fixed resolution of 140m and various gini thresholds.

First, we observe that both sizes increase with gini threshold. This is consistent with the fact that higher gini thresholds allow for more heterogeneous clusters, so clusters grow bigger. More importantly, let us note that anomalies average sizes are much higher than overall average cluster size. It seems that the method favors big anomalous region. So, as for SaTScan, we observe an additive bias.

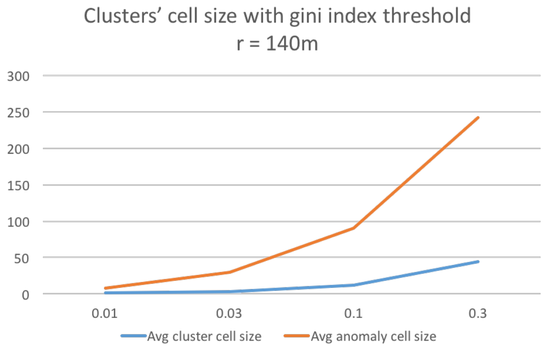
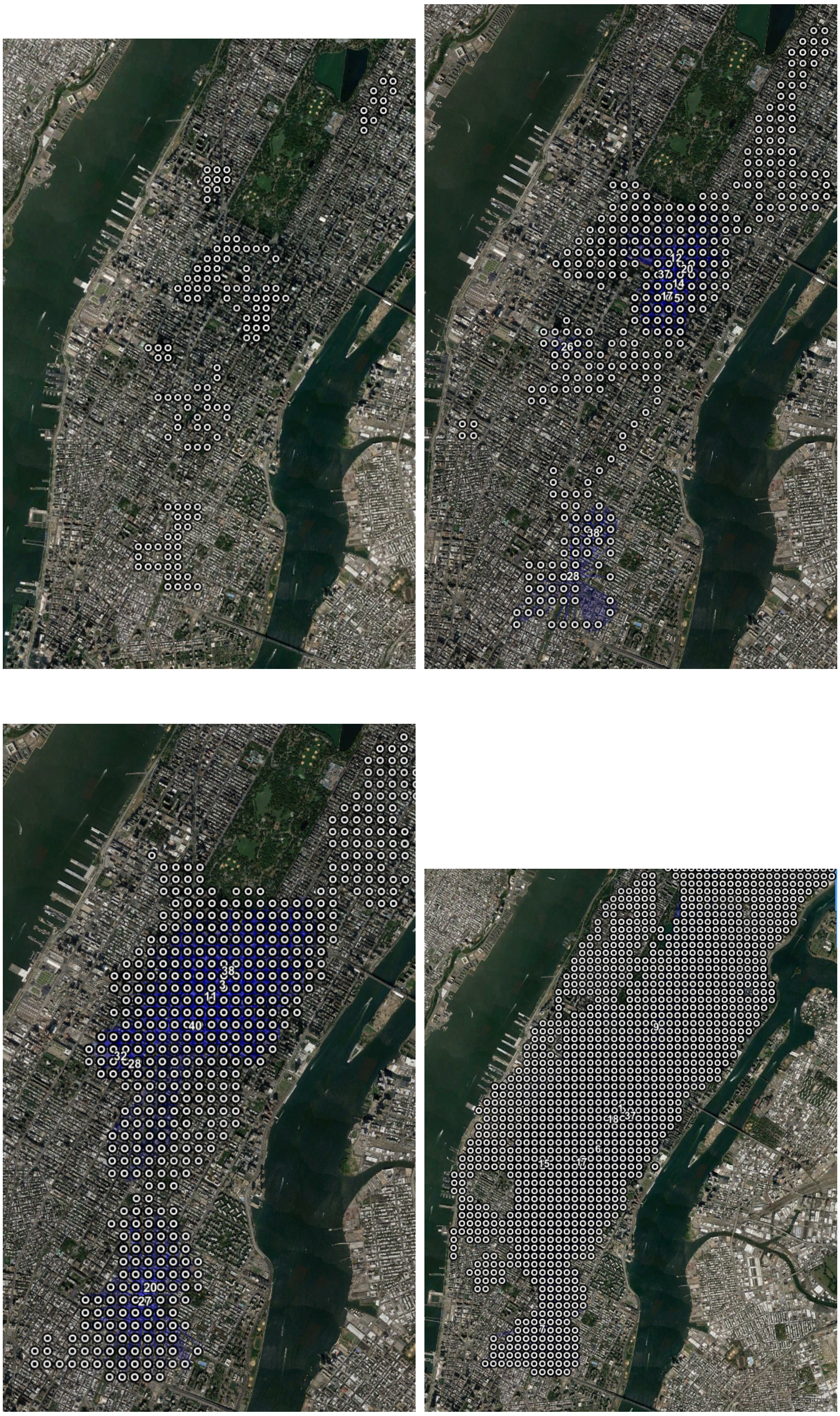


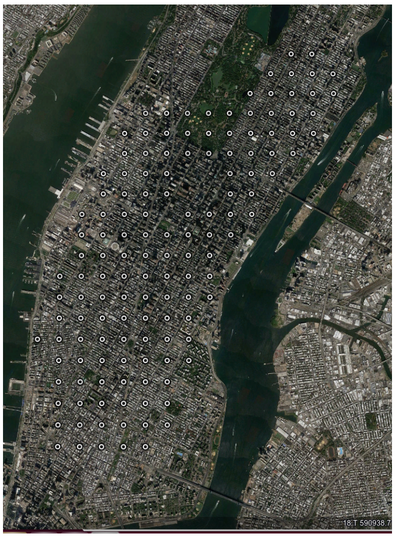
Figure 6.1: Cluster size with gini threshold, r = 140m

City coverage analysis

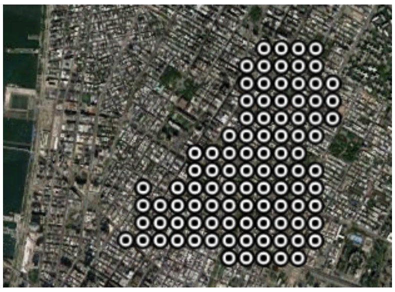
Let us now study the spatial size of clusters - that is the space extension, and not the size of the clusters as their number of cells. Below we show the global city coverage of NYC with various $g \in [0.01, 0.03, 0.1, 0.3]$ for a 140m grid resolution. By city coverage, we mean the subset of the spatial points monitored which appear at least once in an anomalous cluster of the top 40 anomalies. It gives a good idea of how big the outputted anomalous clusters are. Those points are showed below, each point being the center of one grid cell. The numbers on the pictures are minor visualization defects.



We observe that with the 0.01 gini threshold, anomalous clusters are detected only around specific busiest parts of NYC: along Broadway, next to Grand Central and Penn Station, in the East Village and Lower East Side. With such threshold, the technique detects localized patterns. City coverage gradually increases with gini threshold. With the higher threshold - g = 0.3 - it includes almost all Manhattan North of Houston Street.



(A) r = 360m, g = 0.3



(B) r = 140m, g = 0.3

Use cases

Grid resolution may amplify the effect of gini thresholds. Lower grid resolutions tend to increase spatial anomalies sizes. Indeed, data is aggregated in larger grid cells and thus is more homogeneous, which leads to larger clusters. We observe the opposite effect with fine grid resolutions. Thus, the effects of grid resolution and gini threshold add up or oppose each other. We distinguish two main use cases of the technique.

i.) Broad tendencies

The displayed anomalous cluster for {r = 360m, g = 0.3} below is typical of the outputted anomalies for such parameters. The cluster is ranked 6 in test statistic - we will from now refer to test statistic rank with the **#6 notation**, that is to say it is the sixth most anomalous cluster. It includes most of the monitored area and lasts between 18:00 and 23:59 on a week day. This is the most active hours of the day for taxis. It should be noted that for big clusters, the neighborhood is mostly composed of the same spatial locations at adjacent time steps, so the analysis is mostly temporal. With a high gini threshold and a low resolution, most of the observed clusters span over most of Manhattan.

With a high gini threshold and a fine grid resolution, the tendencies observed are less spread. We show the cluster #7 of {r=140m, g=0.3}. The #7 cluster lasts from 1:00 to 2:59, and includes the districts with the most active nightlife in NYC: East Village, Lower East Side, Nolita. Even if those neighborhoods do not have the exact same nightlife patterns, the high gini threshold allows clustering to group them together.

ii.) Localized patterns With a low gini threshold and high grid resolution, more punctual events can be detected. The example given is obtained with {r=140, g=0.01}. It is cluster #24, it only spans from 7:00 to 7:59 on a weekday. It highlights the morning activity of Penn Station. The number of pickups is high as many commuters arrive by train.

With low gini threshold and low resolution, the technique focuses on localized patterns, but fails to detect them properly. In fact, outputting a couple of large spatial cells makes the identification of the event tedious. It is illustrated by the example {r=500, g=0.01}.

(A) r = 140m, g = 0.01

(B) r = 500m, g = 0.01

In this example cluster #8 lasts from 8:00 to 9:59 on a week day. The interpretation of this event is unclear due to the low resolution. The cell at the extreme left of the cluster includes a part of Broadway. The one at the center is close to Central Station. The high values detected at those points could be explained by high activity on Broadway and/or next to Central Station. It is hard to say that a single event is at the source of the creation of this homogeneous cluster - remember that a low gini threshold leads to homogeneous cells in the cluster. The best we can do is trying to understand the influence of the spatial features included in the cluster. In short, a low gini threshold should output localized patterns, but a low resolution makes it difficult to understand what exactly caused the anomaly, because cells are too large.

A versatile tool

This study shows that Telang method can be used in different ways depending on the gini threshold used. Even if the impact of grid resolution is limited, the choice of r should not result in an extremely opposite effect against the one of the gini threshold. Practically and broadly speaking, grid resolutions <140m are the most relevant given our urban data set. A gini threshold of 0.01 outputs very localized pattern, g > 0.1 outputs tendencies. g = 0.03 is the most versatile choice.

6.5.1 Time Size

Let us examine how temporal lengths of events vary with g and r. In the plot below, the curves show how average cluster timespan in the top40 anomalies vary with gini threshold.

First, we note that tendencies are similar for the different grid resolutions - except for 140m, which is slightly different. Average timespan is broadly similar with g = 0.01 and g = 0.03, between one and two hours - with the slight exception of r=360m. Is is also very close between g=0.1 and g=0.3. The big gap lies between 0.03 and 0.1. We could explain this by the patterns of the data. Taxi pickups and drop-offs is quite heterogeneous, so for g < 0.03, most clusters stick to a one-hour length. When g reaches 0.1, the cluster sizes become so big that the clusters neighborhoods consist mostly of their purely temporal neighbor, so the algorithm would give high test scores to regions which stand out temporally. So when g increases over 0.1, the same temporal stand-out periods are detected - nightlife or early activity mostly, and so the average cluster timespan remains stable.

6.5.2 Anomalous Densities Analysis

It appears that the top-ranked anomalies are almost always high density anomalies. The mean proportion of high density anomalies in the top40 anomalies of all our experiments - g takes values in [0.01, 0.03, 0.1, 0.3] and r in [100, 140, 190, 250, 360] - is > 97%.

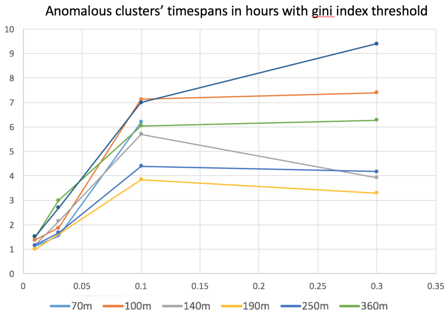


Figure 6.4: Average event timespan with gini threshold

So it appears that as STP this method favors the detection of **high density anomalies**. This is due to the nature of the test statistic. The test performed assesses whether a given region has a mean cell count significantly different from the mean value of its space-time neighborhood. The absolute difference between means is more important for usual high density anomalies - typically 4 times the mean density - than for usual low density anomalies.

6.5.3 Shape

The method outputs many irregular clusters. The use of a homogeneity-constrained DBSCAN algorithm creates clusters with very little shape constraint, cells are only grouped when they have similar densities. The main pro is that clusters can fit irregular urban patterns. We note two main irregular patterns outputted.

The first one consists in elongated clusters. It is particularly noticeable with low gini thresholds, when clusters are rather small. The example in 6.5 shows how clusters can fit parts of the road network. It is cluster #28 obtained with g= 0.01, r = 100m, lasting from 19:00 to 19:59. It fits a part of Broadway, which in that case shows anomalous high activity.

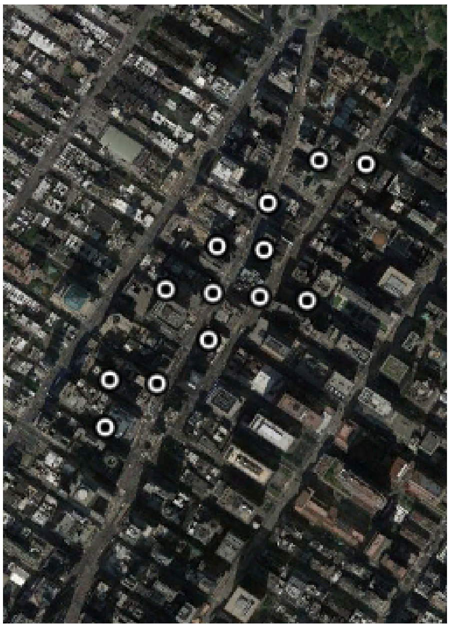
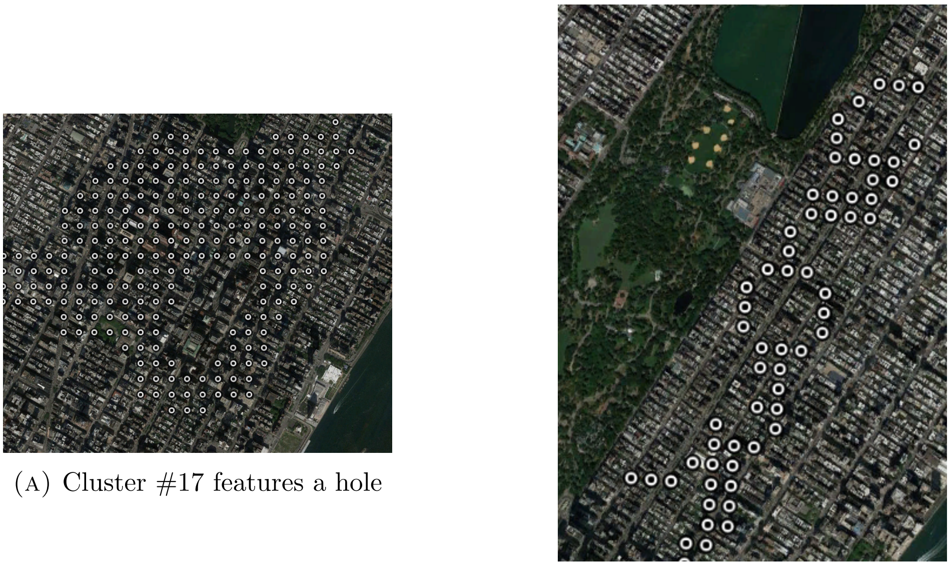


Figure 6.5: Cluster #28 fits Broadway

The second pattern consists in region with holes. The example given in 6.6a is cluster #17 obtained with g = 0.3, r = 100m, lasting from 7:00 to 8:59. It shows how a small region can be isolated inside the anomalous cluster. It can be due to too high or too low density compared to the rest of the cluster. We observe here that the region close to Grand Central is excluded from the cluster, so that the cluster captures the global early activity of the eastern part of Midtown. Since it corresponds to train arrivals rush hours, we believe that it was excluded because of too high density. However, we did not find the complementary Grand Central cluster in the top40 anomalies.

The drawback of outputting irregular clusters is that sometimes event interpretability is lost. In 6.6b cluster #30 was obtained with g = 0.03, r =



(A) Cluster #17 features a hole

(B) Cluster #30: losing interpretation

Figure 6.6: Irregular shape events

100m. It displays an elongated shape with prolongations in various directions and a belt-shaped part. We could not find any city pattern matching those shapes. However, those clusters remain in small number in our study.

6.5.4 Test Score

We already observed particular features of the test statistic used by the method. It favors high density anomalies and big regions. Now let us understand how it globally varies with parameters. Let us examine in 6.7 how the test score vary with the parameters.

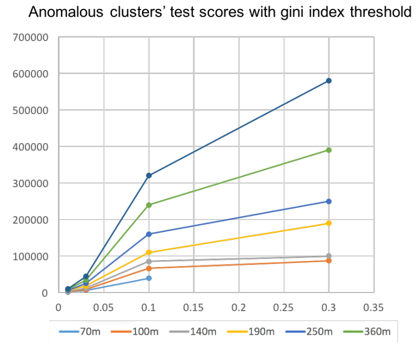


Figure 6.7: Test score with various r and g

We note that test score increases with gini threshold and decreases with grid resolution. It is due to the fact that high gini thresholds and low grid resolutions both drive densities up, and we saw that the used test statistic is sensitive to high densities. The higher g, the bigger the clusters, the higher the total density. With low grid resolutions, the individual cell densities increase because aggregation of taxi presence is performed over a larger spatial area. Moreover, multiple counting is more important.

Test score is useful to compare events significance for a fixed g and r. However, it is not consistent to use it to compare events obtained with different parameters. Indeed, a global tendency in Midtown would have a much higher score than a road blockage for a parade, but the two events are relevant.

6.5.5 NYC Events Exploration and Stability

As for SaTScan, we note that most prominent patterns appear multiple times in the top 40 anomalies outputted. Often the two most prominent events occupy more than 60% of the top 40.

Broadly speaking, those most prominent events are robustly detected for various grid resolutions. The impact of grid resolution on Telang et al. method is similar to the effect of space bound on SaTScan. Grid resolution affects the event detection task in three ways.

1) The same event is detected with multiple grid resolutions, but not with the exact same and time space extension. In those cases, as with SaTScan the anomaly detected with a lower grid resolution concentrates on the heart of the event - e.g. the busiest hours for nightlife.

2.) A single event may be detected by the algorithm with high grid resolution as two different events located in two close places. This was observed for SaTScan too. In the case of nightlife, East Village and Lower East side may be grouped on weekends with low resolution, and detected separately with high resolution.

3.) Events appear when grid resolution decrease, as for SaTScan.

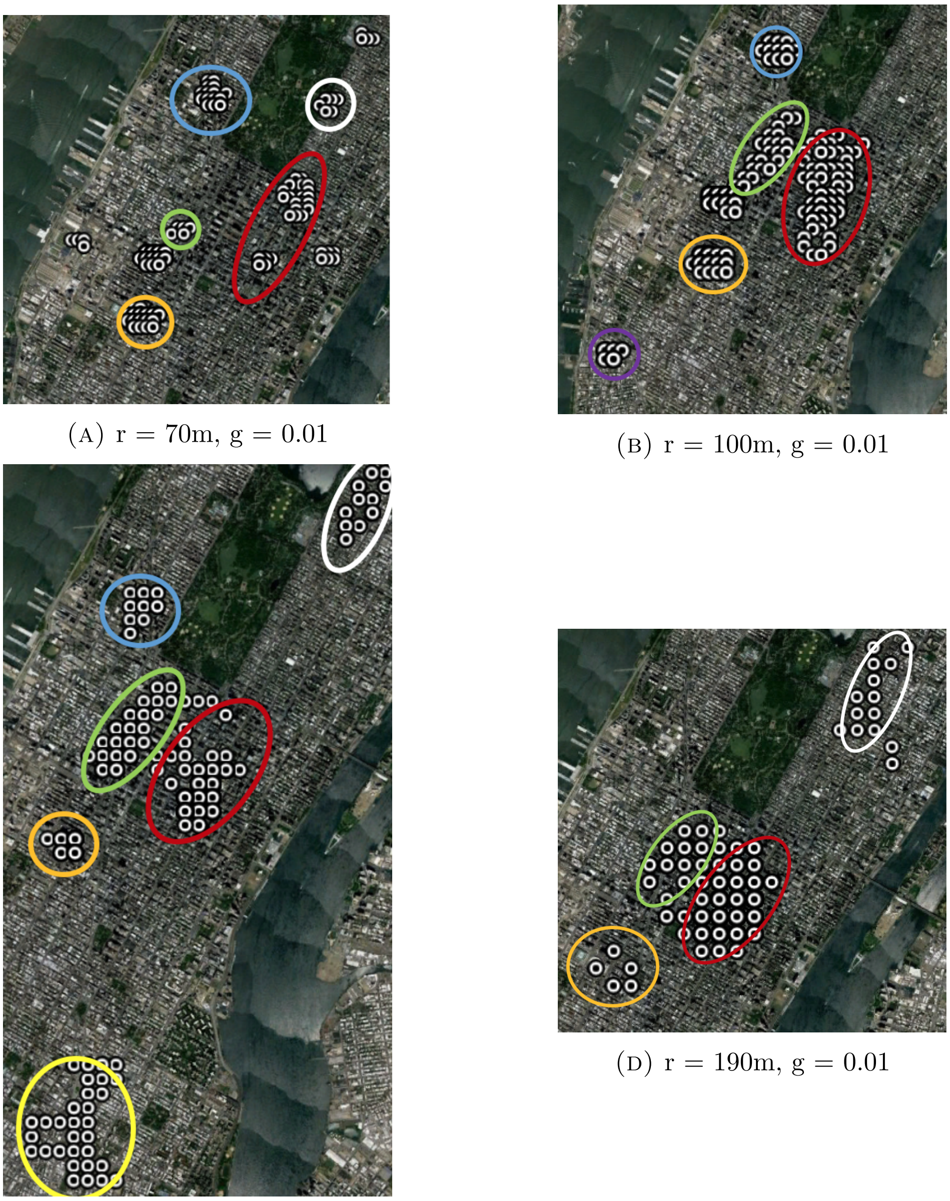
We give an idea of the exploration performed and the stability of the results with a visual comparison of the city coverage obtained with a fixed g = 0.01 and varying grid resolution. We chose a low gini threshold because it is easier to distinguish between clusters when all are juxtaposed. In 6.8, each color circle corresponds to a type of event. For instance, several high density anomalies observed at the same approximate location on the same hours on week days will be highlighted as one single event type.

We indicate the events detected in the table below. All are high density anomalies.

Color	Location	Day	Start hour	Duration	Comment
red	NE Midtown	Weekday	8am	1h/2h	Early activity
orange	Penn Station	Weekday	8am	1h/2h	Train arrivals
blue	Lincoln Center	Variable	10pm	1h	End of shows
green	NW Midtown	Weekend	7pm	1h/2h	
white	Upper East Side	Weekday	7am	1h	
yellow	EastV/LowerEV	Weekend	11pm	2h	Nightlife
purple	West Village	Weekend	11pm	2h	Nightlife

Most of those recurrent events can be easily interpreted. A couple of them are still unclear to us. Upper East Side being a wealthy area, people may tend to take more taxis to commute to work, which could explain the events circled in white.

Let us note that decreasing grid resolution leads to unveil more isolated events. Events outputted in the r = 70m experiment include a particular week end afternoon next to central park for instance, and other non-recurrent events.



(A) r = 70m, g = 0.01

(B) r = 100m, g = 0.01

(C) r = 140m, g = 0.01

(D) r = 190m, g = 0.01

Figure 6.8: Overall event classification

6.6 Computation

The computation time of Telang algorithm is roughly $O(nmp^2ln(p))$ where n is the number of space-time cells, m the number of neighbors per grid cell and p the average cluster size - see [15]. Below are the computation times with all g and r used. The experiments were performed with a processor AMD Opteron 6276 2.3 GHz and 1TB of RAM.

The influence of cluster size being $p^2ln(p)$, it is low compared to the influence of the number of cells. Increasing gini threshold increases average cluster size, while refining grid resolution increases cell number. That is why computation time is mostly driven by grid resolution.

6.7 Possible Improvements

The main application of Telang method on our taxi dataset is to detect high density regular patterns of the city. This is mainly due to the baseline which grants high score to regions different from their adjacent neighborhoods, and which takes into account absolute density. The homogeneity constraint restrict timespan to several hours. Depending on the gini threshold, it outputs broad tendencies or localized patterns. Even though grid resolution has limited effects on results, extremes should be avoided since they interfere with the influence of the gini parameter.

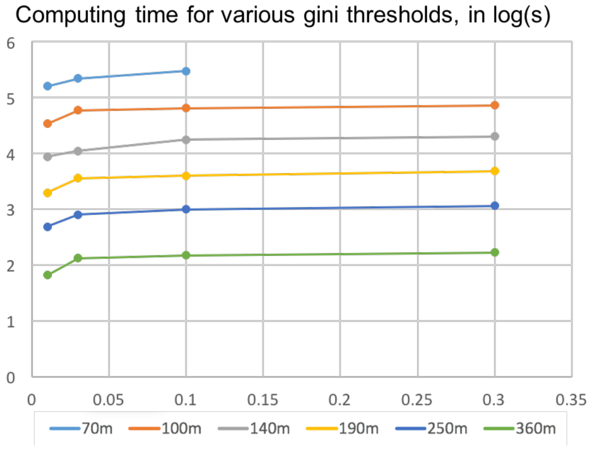


Figure 6.9: Computation times for various r and g

To change the type of events detected by the method, we could consider the following changes.

- Incorporating an expectation to the baseline. Instead of treating raw absolute densities, a monthly or day-of-week adapted mean could be subtracted, or be used to compute a ratio. Further time series analysis could be performed to generate an expectation. This may lead to output patterns which would not be regular patterns of the city since those patterns would have been removed from the data before.

- Changing neighborhood definition. Depending on the objective of the analysis, the neighborhood range could be different for time and space. The neighborhood could even be composed of non adjacent cells. For instance, time steps corresponding to the same hours of the day for different days, or for other week days, could compose the neighborhood instead of adjacent time steps. This would allow to incorporate regular patterns to the baseline too.

Chapter 7

Comparison and Conclusion

7.1 Big Picture Comparison of STP and Telang et al.

Let us identify the common points and main differences of the two methods implemented. First, both mostly identify day-of-week and hour-of-day patterns, because their baselines do not take into account those two effects. Both show a spatial additive bias: bigger anomalous clusters tend to have higher anomalousness score. They are also more sensitive to high counts anomalies. Both techniques can be used to detect broad tendencies or localized patterns depending on how their parameters are set. Their results are mostly consistent when parameters vary.

The clustering method ensures that the clusters computed have rather homogeneous value, whereas SaTScan clusters may be heterogeneous. STP clusters have regular ellipse shapes, while Telang et al. clusters may show irregular shapes. Computation shows opposite trends for the two clusters. Using Telang et al. method to detect localized patterns requires to set a fine grid resolution, which is associated to high computation time. Using STP for the same purpose is associated with low space bounds and so low computation time. More diverse events were found with Telang et al. method, so it may be better suited for event exploration.

7.2 Conclusion and Possible Future Work

We showed in our theoretical study that event detection on space-time point data can be split into independent problems. We presented the most prominent solutions to those problems. Moreover, we implemented two major techniques. We tried to understand both their use cases as is and the influence of every aspect of them. We showed that the most important aspect of those two techniques are the definition of the baseline. Both baselines led to detect day-of-week and hour-of-day effect, but except the most striking events, the events outputted by the two techniques differ.

When having a space time point event detection problem, one may choose to implement one of the two techniques illustrated in this report. One could also choose to combine various elements presented in the study in order to tailor ones approach to their specific problem. In applied research in the field there still exist great opportunities for incremental progress, since many combinations of solutions to the sub-problems highlighted were not implemented.

Further work could consist in include in the comparison techniques from the unaddressed main categories of space-time anomaly detection: PCA and Topology. Then, it would be interesting to measure the detection power gap between a parallel monitoring technique and a space time technique. One could also focus on traffic data and tailor a particular baseline to better address irregular events. Finally, application fields could be mapped to the most adapted category or to the most adapted baseline computing given the typical features of their data.

Bibliography

[1] Jianhua Guo et al. Real time traffic flow outlier detection using short-term traffic conditional variance prediction. *Transportation Research Part C Emerging Technologies*, 2014.

[2] Sergi Trilles Oliver et al. Real-time anomaly detection from environmental data streams. *Lecture Notes in Geoinformation and Cartography*, 2015.

[3] Franke et al. Detection and exploration of outlier regions in sensor data streams. *2008 IEEE International Conference on Data Mining Workshops*, 2008.

[4] Wu et al. Spatio-temporal outlier detection in precipitation data. *Knowledge Discovery from Sensor Data*, 2010.

[5] Doraiswamy et al. Using topological analysis to support event-guided exploration in urban data. *IEEE Trans Vis Comput Graph*, 2014.

[6] Tork et al. Spatio-temporal clustering methods classification. 2012.

[7] www.satscan.org.

[8] Kisilevitch et al. Spatio-temporal clustering: a survey. *Data Mining and Knowledge Discovery Handbook*, 2010.

[9] Birant et al. St-dbscan: An algorithm for clustering spatialtemporal data. *Data Knowledge Engineering*, 2007.

[10] Chawla et al. Inferring the root cause in road traffic anomalies. *2012 IEEE 12th International Conference on Data Mining*, 2012.

[11] Pang et al. On detection of emerging anomalous traffic patterns using gps data. *Data Knowledge Engineering*, 2013.

[12] Zhang et al. Statistics-based outlier detection for wireless sensor networks. *International Journal of Geographical Information Science*, 2012.

[13] Kulldorff. A spatial scan statistics. *Communications in Statistics*, 1997.

[14] K.P. Agrawal et al. Spatio-temporal outlier detection technique. 2015.

[15] Telang et al. Detecting localized homogeneous anomalies over spatio-temporal data. *Data Mining and Knowledge Discovery*, 2014.

[16] Cheng et al. Spatial and temporal reasoning for ambient intelligence systems. *COSIT 2009 Workshop Proceedings*, 2009.

[17] Yuxiang et al. Detecting spatio-temporal outliers in climate dataset: A method study. *Geoscience and Remote Sensing Symposium*, 2005.

[18] Rogers et al. Detecting spatio-temporal outliers with kernels and statistical testing. *Geoinformatics, 2009 17th Int...*, 2009.

[19] McGuire et al. Spatiotemporal neighborhood discovery for sensor data. *Proceeding Sensor-KDD'08 Proceedings of the Second international conference on Knowledge Discovery from Sensor Data*, 2008.

[20] Janeja et al. Spatial neighborhood based anomaly detection in sensor datasets. *Data Min Knowl Disc*, 2010.

[21] Bhushan et al. Incremental principal component analysis based outlier detection methods for spatiotemporal data streams. *International Workshop on Spatiotemporal Computing*, 2015.

[22] Chandola et al. Anomaly detection: A survey. *ACM Comput. Surv*, 2009.

[23] Gupta et al. Outlier detection for temporal data: A survey. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, NO. 1, JANUARY 2014 1*, 2014.

[24] Rao et al. Spatiotemporal data mining: Issues, tasks and applications. *International Journal of Computer Science Engineering Survey (IJCSES) Vol.3, No.1, February 2012*, 2012.

[25] Wu et al. A lrt framework for fast spatial anomaly detection. *KDD '09 Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009.

[26] Neill. Expectation-based scan statistics for monitoring spatial time series data. *International Journal of Forecasting 25*, 2009.

[27] Tango and Takahashi. A flexibly shaped spatial scan statistic for detecting clusters. *International Journal of Health Geographics*, 2005.

[28] Neill. Fast subset scan for spatial pattern detection. *Journal of the Royal Statistical Society*, 2012.

[29] Lingwall Neill. A nonparametric scan statistic for multivariate disease surveillance. *Advances in Disease Surveillance*, 2007.

[30] Cheng et al. Spatiotemporal outlier detection: Did buoys tell where the hurricanes were? *Papers in Applied Geography*, 2016.

[31] Rocha et al. Db-smot: A direction-based spatio-temporal clustering method. *Intelligent Systems (IS)*, 2010.

[32] Birant and Kut. Spatio-temporal outlier detection in large databases. *Journal of Computing and Information Technology*, 2006.

[33] Lakhina et al. Diagnosing network-wide traffic anomalies. *SIGCOMM*, 2004.

[34] Brauckhoff et al. Applying pca for traffic anomaly detection: Problems and solutions. *IEEE Communications Society*, 2009.

[35] Kuang et al. Detecting traffic anomalies in urban areas using taxi gps data. *Mathematical Problems in Engineering*, 2015.

[36] Yand et al. Detecting road traffic events by coupling multiple timeseries with a nonparametric bayesian method. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 2014.

[37] Lanorte et al. On the use of the principal component analysis (pca) for evaluating vegetation anomalies from landsat-tm ndvi temporal series in the basilicata region (italy). *Lecture Notes in Computer Science*, 2015.

[38] Anselin. Local indicators of spatial associationlisa. *Geographical Analysis*, 1995.

[39] Albanese et al. Rough sets, kernel set and spatio-temporal outlier detection. *IEEE TRANSACTIONS OF KNOWLEDGE AND DATA ENGINEERING*, 2011.

[40] Souto and Liebig. On event detection from spatial time series for urban traffic applications. 2015.

[41] Kulldorff et al. A spacetime permutation scan statistic for disease outbreak detection. *PLOS Medicine*, 2005.

[42] Evans et al. Statistical distributions, 3rd ed. 2000.

[43] Satscan user guide.