Detecting Collective Anomalies from Multiple Spatio-Temporal Datasets across Different Domains

Yu Zheng[1,2], Huichu Zhang[2,1], Yong Yu[2]
[1]Microsoft Research, Beijing, China
[2]Shanghai Jiao Tong University, Shanghai, China
{yuzheng, v-huiczh}@microsoft.com, yyu@cs.sjtu.edu.cn

ABSTRACT
The collective anomaly denotes a collection of nearby locations that are anomalous during a few consecutive time intervals in terms of phenomena collectively witnessed by multiple datasets. The collective anomalies suggest there are underlying problems that may not be identified based on a single data source or in a single location. It also associates individual locations and time intervals, formulating a panoramic view of an event. To detect a collective anomaly is very challenging, however, as different datasets have different densities, distributions, and scales. Additionally, to find the spatio-temporal scope of a collective anomaly is time consuming as there are many ways to combine regions and time slots. Our method consists of three components: *Multiple-Source Latent-Topic (MSLT)* model, *Spatio-Temporal Likelihood Ratio Test (ST_LRT)* model, and a candidate generation algorithm. *MSLT* combines multiple datasets to infer the latent functions of a geographic region in the framework of a topic model. In turn, a region's latent functions help estimate the underlying distribution of a sparse dataset generated in the region. *ST_LRT* learns a proper underlying distribution for different datasets, and calculates an anomalous degree for each dataset based on a likelihood ratio test (*LRT*). It then aggregates the anomalous degrees of different datasets, using a skyline detection algorithm. We evaluate our method using five datasets related to New York City (NYC): 311 complaints, taxicab data, bike rental data, points of interest, and road network data, finding the anomalies that cannot be identified (or earlier than those detected) by a single dataset. Results show the advantages beyond six baseline methods.

Categories and Subject Descriptors
H.2.8 [**Database Management**]: Database Applications - *data mining, Spatial databases and GIS*;

Keywords
Urban computing, anomaly detection, big data, cross-domain.

1. INTRODUCTION
Advances in sensing technologies and large scale computing infrastructures have generated a diverse array of data on cities, such as traffic flow, human mobility and social media. These datasets are usually associated with spatio-temporal information and can be individually sparse. When deposited together, however, they may represent urban dynamics and rhythms collectively [18].

In this paper, we detect the *collective* anomalies in a city instantaneously by using multiple spatio-temporal datasets across different domains. Here, '*collective*' has two types of meanings. One denotes the spatio-temporal collectiveness. That is, a collection of nearby locations is anomalous during a few consecutive time intervals, while a single location in the collection may not be anomalous at a single time interval if being checked individually. The other is that an anomaly might not be that anomalous in terms of a single dataset but considered an anomaly when checking multiple datasets simultaneously. Such collective anomalies could denote an early stage of an epidemic disease, the beginning of a natural disaster, an underlying problem, or a potentially catastrophic accident. The follows are two examples.

SIGSPATIAL'15, November 03-06, 2015, Bellevue, WA, USA
© 2015 ACM. ISBN 978-1-4503-3967-4/15/11…$15.00
DOI: http://dx.doi.org/10.1145/2820783.2820813

Example 1: As illustrated in Figure 1, an unusual event has just happened at location r_1, affecting its surrounding locations, e.g. from r_2 to r_6. As a result, the traffic flow entering r_1 from its surrounding locations increases 10 percent. Meanwhile, social media posts and bike rental flow around these locations change slightly. The deviation in each single dataset against its common pattern is not significant enough to be considered anomalous. However, when putting them together, we might be able to identify the anomaly, as the three datasets barely change simultaneously to that extent. In addition, locations from r_1 to r_6 formulate a collective anomaly in a few consecutive time intervals, e.g. from 2 to 4 pm. If we check location r_2 individually at 2pm, it might not be considered an anomaly.

Figure 1. A collective anomaly witnessed by three sources

Example 2: The groundwater under a village is being polluted. As a result, reports of sickness in the village increase slightly. The occurrences of birds flying over the village drop a bit, and the food production yield around the village is reduced by 10 percent. The change in each individual dataset is quite normal. If we check the three or more datasets together, however, we may find that this is very unusual. Like the first example, the anomaly exists in a certain spatial range covering the village and a time span, e.g. in the last half year. Being able to detect such anomalies is of great importance to social good and people's daily lives.

The main benefits of our research are two-fold. First, we can detect anomalies that cannot be identified using a single dataset. Intrinsically, a single dataset only describes an event (or a region) from one point of view. Particularly when the dataset is very sparse, which is very common in reality, the detection of anomalies with a single set becomes very difficult. Combining multiple (sparse) datasets can mutually reinforce each other, helping detect anomalies better and earlier. Second, such a collective anomaly offers a spatio-temporal scope that can pinpoint the underlying problem in time and formulate a panoramic view of an event.

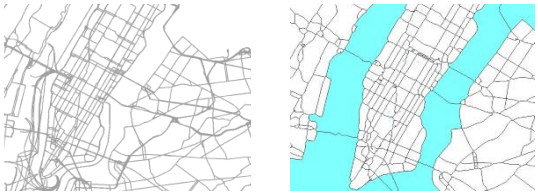
To detect collective anomalies from cross-domain datasets is very challenging for three reasons. First, as some data sets are very sparse, it is difficult to estimate their true distributions based on limited observations. As a consequence, it is hard to measure the deviation of an instance from its normal distribution. Second, datasets of different domains have different distributions and scales. To integrate them together into a collective measurement remains a challenge. Third, as there are many ways to combine regions and time slots, finding the spatio-temporal scope of a collective anomaly is very time consuming. This conflicts with the instant detection of anomalies.

To address these issues, we propose a probability-based anomaly detection method, which consists of three main components: a *Multiple-Source Latent-Topic (MSLT)* model, a *Spatio-Temporal Likelihood Ratio Test (ST_LRT)* model, and a candidate generation algorithm. The contributions of our work are as follows:

- The *MSLT* model combines multiple datasets in a topic model to better estimate the underlying distribution of a sparse dataset, leading to more accurate anomaly detection.
- The *ST_LRT* model aggregates the information of multiple datasets across multiple regions to detect anomalies, by adapting Likelihood Ratio Test to a spatio-temporal setting.
- We propose an efficient algorithm to find the anomaly candidates that satisfy spatio-temporal constraints.
- We evaluate our method using five datasets from NYC. We find the anomalies that cannot be identified if only using a single dataset. We can detect anomalies earlier than other methods only checking a single dataset. The datasets have been released at http://research.microsoft.com/apps/pubs/?id=255670.

2. OVERVIEW

Definition 1. Region: There are many definitions of location in terms of different granularities and semantic meanings. In this study, we partition a city into regions $\boldsymbol{r} = \{r_1, r_2, \ldots, r_m\}$ by major roads, such as highways and arterial roads, using a map segmentation method [13]. Consequently, each region is bound by major roads, carrying a semantic meaning of neighborhoods or communities, as illustrated in Figure 2. We then use regions as the minimal unit of location in the following study, though a region can be a uniform grid in other applications.



A) Raw road network B) Segmented regions

Figure 2. Map segmentation and regions

Definition 2. Dataset: A dataset s is a stream of instances, each of which can be simplified as a triplet $< l, t, v >$, where l is a geographic coordinate; t is a timestamp; $v \in s.C =< c_1, c_2, \ldots, c_n >$ is a categorical value, e.g. the level of traffic conditions.

Problem Definition: Given multiple datasets $\boldsymbol{S} = \{s_1, s_2, \ldots\}$ during the recent t time intervals $[t_1, t_t]$ and that over a period of historical time, we project $\boldsymbol{S}$ onto regions $\boldsymbol{r}$, formulating a spatio-temporal set $\boldsymbol{\mathcal{T}} = \{< r_1, t_1 >, < r_2, t_1 >, \ldots, < r_m, t_1 >, < r_1, t_2 >, < r_2, t_2 >, \ldots, < r_m, t_2 >, \ldots, < r_m, t_t >\}$. An entry $< r, t >$ in $\boldsymbol{\mathcal{T}}$ is associated with a vector, $\boldsymbol{v} =< s_1.|c_1|, s_1.|c_2|, \ldots, s_1.|c_n|, s_2.|c'_1|, s_2.|c'_2|, \ldots, s_n.|c''_1|, s_n.|c''_2|, \ldots >$, denoting the number of instances in each category of each dataset in region r at time interval t. We instantly detect a set of anomalies $\mathcal{A} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m\}$ each $\mathcal{T}_m$ is a collection of spatio-temporal entries from $\boldsymbol{\mathcal{T}}$, satisfying the following three criteria:

1) $\forall r_i, r_j \in \mathcal{T}_m,\ dist(r_i, r_j) \le \delta_d$,
2) $\forall t_i, t_j \in \mathcal{T}_m,\ |t_i - t_j| \le \delta_t$,
3) $ST\_LRT(\mathcal{T}_m)$== true.

Figure 3 presents an example illustrating the problem definition. Marked by red lines in the left part of Figure 3, three collective anomalies (a_1, a_2 and a_3) are detected based on two data sources $s_1$and s_2 (denoted by circles and squares respectively) from four consecutive time intervals $[t_1, t_4]$. To simplify the illustration, we assume each region is a cell in a uniform grid. The instance in s_1 pertains to two categories: $c_1$and c_2 (denoted by different colors); so does s_2 (i.e. c'_1 and c'_2). By projecting s_1 and s_2 onto these regions, we can count the vector $\boldsymbol{v}$ associated with each entry, e.g. the $\boldsymbol{v}$ of $<r_5, t_2>$ is <0, 1, 0, 2>. Anomaly a_1 contains three regions across two time intervals from t_3 to t_4 (i.e. 6 entries in total), while a_2 is comprised of three entries: $<r_5, t_2>$, $<r_4, t_3>$ and $<r_5, t_3>$. If we check s_1 and s_2 individually, $<r_5, t_2>$ only has one more instance occurring in each dataset, as compared to $<r_5, t_1>$. But, if checking s_1 and s_2 together, we find it is rare to see the two datasets increasing simultaneously. So, $<r_5, t_2>$ can be considered anomalous. In addition, if we check $<r_5, t_3>$ separately, it may not be considered anomalous either. However, when combing with $<r_4, t_3>$ and $<r_5, t_2>$, we find that the overall presence of $s_1$and s_2 in the three entries increases significantly. Thus, they may be regarded as an anomaly collectively. When checking the combination of entries in $\boldsymbol{\mathcal{T}}$, we require that the geographic distance between any two entries in the same anomaly should be smaller than a threshold δ_d (i.e. the first criterion). In addition, the time interval between any two entries in an anomaly should be smaller than another threshold δ_t (i.e. the second criterion). The two requirements ensure the spatio-temporal compactness of a detected anomaly, while aggregating individual regions and time intervals that could describe the same anomaly.

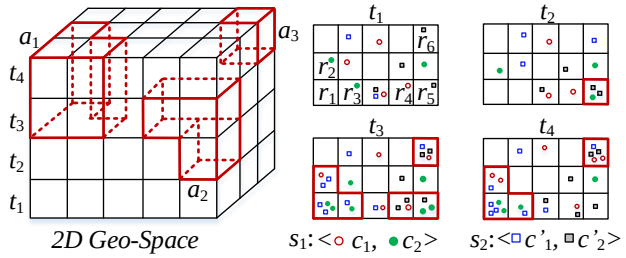


Figure 3. Illustration of the problem definition

Framework: Algorithm 1 presents the procedure of our method, where Lines 1-4 are done in an offline process, while Line 5 is online. The *MSLT* model combines multiple datasets to infer the latent functions of a geographic region (Line 1), through a mutually reinforced learning process in the framework of a topic model. A region's latent functions help, in turn, estimate the underlying distribution of a sparse dataset generated in the region (see Line 3), leading to a more accurate anomaly detection. The *ST_LRT* model first learns an underlying distribution of different datasets. Particularly, it leverages the Zero-Inflated Poisson (*ZIP*) Model and the topic-word distribution (i.e. $\boldsymbol{\varphi}$, $\boldsymbol{\theta}$ inferred by *MSLT*) to learn the underlying distribution for a sparse dataset. Second, The *ST_LRT* calculates an anomalous degree for each dataset by performing a likelihood ratio test across different regions and time intervals. Third, The *ST_LRT* aggregates the anomaly degrees of different datasets, using a skyline detection algorithm. Algorithm 3 in Section 4 details the procedure of the *ST_LRT*. The candidate

generation algorithm (Line 4) employs computational geometry to check the spatial constraint between regions. In addition, it finds an upper bound likelihood ratio for the combination of <region, time> entries, pruning impossible combinations based on the skylines that have been detected.

Algorithm 1: Collective_Anomaly_Detection

Input: Datasets $\boldsymbol{S}$, a collection of spatio-temporal entries $\mathcal{T}$, threshold δ_d and δ_t, a list of skyline outlier degrees SLA detected over a period of historical time

Output: A set of collective anomalies $\mathcal{A}$.

1. $(\boldsymbol{\varphi}, \boldsymbol{\theta})$ ←MSLT($\boldsymbol{S}, \mathcal{T}$); //refer to Section 3
2. **For each** $s \in \boldsymbol{S}$ do
3. $s.Dist$ ← Learn_Distributions(s, $\boldsymbol{\varphi}$, $\boldsymbol{\theta}$); //refer to Algorithm 2
4. $\mathcal{T}'$ ←Circel_Based_Spatial_Check($\mathcal{T}, \delta_d$); // refer to Section 5
5. $\mathcal{A}$ ←ST_LRT($\mathcal{T}', \boldsymbol{S}, SLA$); //refer to Algorithm 3
6. **Return** $\mathcal{A}$;

3. Multiple-Source Latent-Topic Model

3.1 Insight

To determine if an instance is anomalous in a dataset, we usually need to measure how far the instance deviates from its underlying distribution. This calls for an estimation of the underlying distribution of a given dataset, which is very difficult when the dataset is sparse. For example, the occurrence of a specific disease in a region may only occur once per several days. If we concatenate the occurrences into a series with zero values denoting the absences, i.e. <0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 2,…>, the mean and variance of the series are very close to zero. At this moment, if using a distance-based anomaly detection method, every non-zero entry in the series will be regarded as an anomaly, because its distance to the mean value (almost 0) is three times larger than the standard deviation (which is also close to 0).

To address this issue, in the *MSLT* model, we combine multiple datasets to better estimate the distribution of a sparse dataset in a region. *First, different datasets in a region can mutually reinforce each other*. Different datasets generated in the same region describe the region from different perspectives. For example, POIs and road network data describe the land use of a region, while taxi and bike flows indicate people's mobility patterns in the region. Thus, combining individual datasets results in a better understanding of a region's latent functions. Bridged by a region's latent functions, there is an underlying connection and influence among these datasets. For instance, the land use of a region would somehow determine the traffic flow in the region, while the traffic patterns of a region may indicate the land uses of the region. After working together to better describe a region's latent functions, different datasets in the region can mutually reinforce each other, thereby helping to better estimate their own distributions. *Second, a dataset can reference across different regions*. For instance, two regions (r_1, r_2) with a similar distribution of POIs and a similar structure of roads could have a similar traffic pattern. So, even if we cannot collect enough traffic data in r_1, we could estimate its distribution based on the traffic data from r_2.

3.2 Graphic Presentation of MSLT

Motivated by the insight, we design a latent-topic model to fuse multiple datasets, as shown in Figure 4 A). In this model, we regard a geographical region as a document; the latent functions of a region correspond to the latent topics of a document; the categories of different datasets are regarded as words; the POIs and road network data in a region are deemed the key words of a document. A simple understanding of the *MSLT* model is that a region is represented by a distribution of latent functions, and a latent function is further represented by a distribution of words. In later presentation, the word '*topic*' equals '*function*', and '*region*' is equivalent to '*document*'.

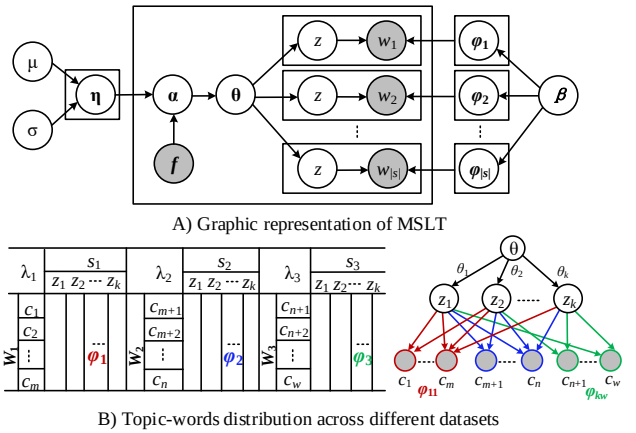


Figure 4. The graphic presentation of the *MSLT* model

More specifically, the gray nodes in Figure 4 A) are observations and the white nodes are hidden variables. $\boldsymbol{f}$ is a vector storing the features extracted from the road network and POIs located in a region. The features include the number of POIs in different categories (e.g. 5 restaurants, 1 cinema, and 1 shopping mall), the total length of roads, and the number of road segments at different levels, etc. $\boldsymbol{\eta} \in \mathbb{R}^{k \times |\boldsymbol{f}|}$ is a matrix with each row $\boldsymbol{\eta}_t$ corresponding to a latent topic; k denotes the number of topics and $|\boldsymbol{f}|$ means the number of elements in $\boldsymbol{f}$. The value of each entry in $\boldsymbol{\eta}$ follows a Gaussian distribution with a mean μ and a standard deviation σ. $\boldsymbol{\alpha} \in \mathbb{R}^k$ is a parameter of the Dirichlet prior on the per-region topic distributions. $\boldsymbol{\theta} \in \mathbb{R}^k$ is the topic distribution for region d. $\boldsymbol{\mathcal{W}} = \{\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_{|\boldsymbol{S}|}\}$ is a collection of word sets, where $\boldsymbol{W}_i$ is a word set corresponding to dataset s_i and $|\boldsymbol{S}|$ denotes the number of datasets involved in the *MSLT*. $\boldsymbol{\beta} \in \mathbb{R}^{|\boldsymbol{W}_i|}$ is the parameter of the Dirichlet prior on the per-topic word distributions of $\boldsymbol{W}_i$. A word w in $\boldsymbol{W}_i$ is one of the categories which s_i's instances pertain to, e.g. $\boldsymbol{W}_1 = \{c_1, c_2, \ldots, c_m\}$. As illustrated in Figure 4 B), different datasets share the same distribution of topics controlled by $\boldsymbol{\theta}_d$, but having its own topic-word distributions $\boldsymbol{\varphi}_i$, $1 \le i \le |\boldsymbol{S}|$, indicated by arrows with different colors. $\boldsymbol{\varphi}_{iz}$ is a vector denoting the word distribution of topic z in word set $\boldsymbol{W}_i$. The generative process of the *MSLT* model is:

1. For each topic t, draw $\boldsymbol{\eta}_t \sim \mathcal{N}(0, \sigma^2 I)$
2. For each word-set $\boldsymbol{W}_i$ and each topic t, draw $\boldsymbol{\varphi}_{it} \sim \mathcal{D}ir(\boldsymbol{\beta})$
3. For each document d (i.e. a region)
   a. For each topic t, let $\alpha_{dt} = exp(\boldsymbol{f}_d^T \boldsymbol{\eta}_t)$
   b. Draw $\boldsymbol{\theta}_d \sim \mathcal{D}irichilet(\boldsymbol{\alpha}_d)$
   c. For each word w in document d
      i. Draw $z \sim Multinomial(\boldsymbol{\theta}_d)$;
      ii. Choose $\boldsymbol{\varphi}_i$ of the corresponding word set that w belongs to;
      iii. Draw $w \sim Multinomial(z, \boldsymbol{\varphi}_{iz})$

Different from Latent Dirichlet Allocation and its variant *DMR* [7][14], the words of *MSLT* come from different datasets. Thus, there are multiple topic-word distributions $\boldsymbol{\varphi}_i$. In addition, the Dirichlet prior $\boldsymbol{\alpha}_d$ of a region also depends on its geographical properties, such as POIs and road networks, rather than an empirical setting. The *MSLT* model can be re-trained every a few months, as a region's latent functions do not change quickly over time. The topic distribution $\boldsymbol{\theta}_d$ of a region and the topic-word distribution $\boldsymbol{\varphi}_i$ of a dataset s_i are used in the *ST_LRT* model to calculate the

underlying distribution of each category in s_i, if s_i is very sparse. Section 6.1.2 gives a detailed configuration of the *MSLT* model.

3.3 Learning Process

While σ^2, $\boldsymbol{\beta}$ and k are fixed parameters, we need to learn $\boldsymbol{\eta}$ and $\boldsymbol{\varphi}$ based on observed $\boldsymbol{f}$ and $\boldsymbol{\mathcal{W}}$. We train the model with a stochastic EM algorithm, in which we alternate between the following two steps. One is sampling topic assignments from the current prior distribution conditioned on the observed words and features. The other is numerically optimizing the parameters $\boldsymbol{\eta}$ given the topic assignments.

The estimation step: This step allocates the topics to words by using Gibbs sampling with the following equation:

$$p(z_i = t|\boldsymbol{w}, \boldsymbol{z_{-i}}, \boldsymbol{\alpha}, \beta) = \frac{\alpha_{dt}+C_t^{-i}}{\Sigma_t(\alpha_{dt}+C_t^{-i})} \frac{\beta+C_{t,w_i,j}^{-i}}{\Sigma_{w\in\mathbb{W}}(\beta+C_{t,w,j}^{-i})}, \quad (1)$$

Equation 1 denotes the probability that topic t is assigned to word w_i ($w_i \in \boldsymbol{w}$), which is the i-th word in document d; α_{dt} is the t-th dimension of Dirichlet prior of region d; $\mathbb{W}$ is the word-set that word w_i belongs to; $C_{t,w,j}$ is the times that topic t has been assigned to word w in the j-th word-set; C_t is the times that topic t has been assigned to words, $C_t = \sum_j \sum_w C_{t,w,j}$; C_t^{-i} calculates the times that topic t has been assigned to words excluding w_i; $\boldsymbol{z_{-i}}$ stands for the excluded topics that have been assigned to w_i.

The numerical optimization step: Integrating over the multinomial $\boldsymbol{\theta}$, we can construct the complete log likelihood for the portion of the model involving the topics $\boldsymbol{z}$:

$$P(z, \boldsymbol{\eta}) = \prod_d \left(\frac{\Gamma(\Sigma_t \exp(\boldsymbol{f}_d^T\boldsymbol{\eta}_t))}{\Gamma(\Sigma_t \exp(\boldsymbol{f}_d^T\boldsymbol{\eta}_t)+n_d)} \prod_t \frac{\Gamma(\exp(\boldsymbol{f}_d^T\boldsymbol{\eta}_t)+n_{t|d})}{\Gamma(\exp(\boldsymbol{f}_d^T\boldsymbol{\eta}_t))}\right) \times \prod_{t,p} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\eta_{tp}^2}{2\sigma^2}\right); \quad (2)$$

where n_d is the number of words in document d, $n_{t|d}$ is the time that topic t occurs in document d. The derivative of the log of Equation 2 with respect to the parameter η_{tp} for a given topic t and the p-th feature in $\boldsymbol{f}$.

$$\frac{\partial \ln P(z,\boldsymbol{\eta})}{\partial \eta_{tp}} = \sum_d f_{dk} \exp\left(\boldsymbol{f}_d^T\boldsymbol{\eta}_t\right) \times \Big(\psi\left(\sum_t \exp\left(\boldsymbol{f}_d^T\boldsymbol{\eta}_t\right)\right) - \psi\left(\sum_t \exp\left(\boldsymbol{f}_d^T\boldsymbol{\eta}_t\right) + n_d\right) + \psi\left(\exp\left(\boldsymbol{f}_d^T\boldsymbol{\eta}_t\right) + n_{t|d}\right) - \psi\left(\exp\left(\boldsymbol{f}_d^T\boldsymbol{\eta}_t\right)\right)\Big) - \frac{\eta_{tp}}{\sigma^2} \quad (3)$$

The numerical optimization step is solved by BFGS algorithm, which is an iterative method for solving unconstrained nonlinear optimization problems. We set $\boldsymbol{\alpha}$ to an initial value and perform the aforementioned two steps iteratively, until the convergence or a certain round of iterations has been conducted.

4. ST_LRT

In this section, we first outline the general idea of the likelihood ratio test, applying this model to the detection of spatio-temporal anomalies with a single dataset. Second, we address the sparsity problem in a dataset and aggregate the results of multiple datasets.

4.1 Likelihood Ratio Test

4.1.1 Preliminaries of LRT

In statistics, a likelihood ratio test is used to compare the fit of two models, one of which (the null model) is a special case of (or 'nested within') the other (the alternative model). This often occurs when testing whether a simplifying assumption for a model is valid, as when two or more model parameters are assumed to be related. Each of the two competing models, the null model and the alternative model, is separately fitted to the data with the log-likelihood recorded. The test statistic (often denoted by Λ) is negative twice the difference in these log-likelihoods:

$$\Lambda = -2\log\frac{likelihood\ for\ null\ model}{likelihood\ for\ alternative\ model}; \quad (4)$$

Whether the alternative model fits the data significantly better than the null model can be determined by deriving the probability or p-value of the obtained difference Λ. In many cases, the probability distribution of the test statistic Λ can be approximated by a chi-square distribution $\chi^2(\Lambda, df)$ with $df = df_2 - df_1$, where df_1 and df_2 represent the number of free parameters of the null model and the alternative model, respectively.

4.1.2 Applying LRT to a Single Set in one Region

When applying LRT to a single dataset s in a single region r, i.e. $\{< r, t_1 >, < r, t_2 >, \dots, < r, t_n >\}$, we assume $< r, t_i >$ follows a certain distribution $\mathcal{P}$ with parameter Θ, e.g. the Poisson distribution with an arrival rate of λ. Suppose the number of occurrences of s observed in $< r, t_i >$ is x_i, the likelihood ratio is defined as:

$$\Lambda(s) = -2\log\left(\frac{\mathcal{P}(x_i|\Theta)}{sup\{\mathcal{P}(x_i|\Theta')\}}\right), \quad (5)$$

where Θ' is the new parameter changing over Θ that fits the observed data best; sup denotes the supremum function that finds the Θ' maximizing $\mathcal{P}(x_i|\Theta')$ and returns the latter [12]. The anomalous degree od of this test is calculated by Equation 6:

$$od = \chi^2\_\text{cdf}(\Lambda, df); \quad (6)$$

where χ^2_cdf denotes the cumulative density function of the Chi-Square distribution; fd is the freedom defined in Section 4.1.1. The time slots, with od larger than a given threshold (i.e. Λ value drops in the tail of χ^2 distribution), are likely to be anomalous.

Figure 5 presents two examples. As shown in Figure 5 A), we first consider a single region and a single time slot, i.e. $< r, t >$, with an underlying Gaussian distribution whose variance is proportional to the mean ($mean$=200 and var=1300). Suppose the number of occurrences of s at time slot t (i.e. x_t) is 70, then the anomalous degree of s is calculated as follows:

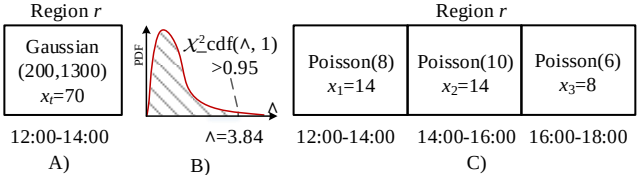


Figure 5. Illustration of applying LRT to a single dataset

1) Calculate the Likelihood of the null model:

$$L_{null} = Gaussian(70|mean = 200, var = 1300); \quad (7)$$

2) Calculate Θ': In this case, we can achieve the maximum likelihood for the alternative model by setting its mean to 70. Since we assume that the distribution's variance is proportional to its mean, we should multiply the variance by $p = \frac{70}{200} = 0.35$. Thus, the new parameter Θ' for the alternative model is ($mean$= 200×0.35=70; var =1300×0.35=455);

3) Calculate the Likelihood of the alternative model

$$L_{alter} = Gaussian(70|mean = 70, var = 455); \quad (8)$$

4) Calculate $\Lambda(s)$ and od: As we assume the invariant linear relationship between the variance and mean, df is 1. According to Equation 5 and 6, the outlier degree is calculated as follows:

$$\Lambda(s) = -2\log\left(\frac{L_{null}}{L_{alter}}\right) = 14.05;$$
$$od = \chi^2\text{\_cdf}(14.05, fd = 1) = 0.999; \quad (9)$$

As depicted in Figure 5 B), if we set the threshold of od to 0.95, $< r, t >$ is apparently an anomaly. The Λ corresponds to 0.95 in the χ^2 distribution with 1-freedom is 3.84. So, $\Lambda(s)$=14.05 > 3.84 is considered the tail of the χ^2 distribution.

In the second example, as illustrated in Figure 5 C), we consider a region r across three consecutive time slots: $\{< r, t_1 >, < r, t_2 >, < r, t_3 >\}$. We suppose the underlying distribution is a Poisson Distribution but different time slots have a different λ: λ_1 =8, λ_2=10, and λ_3=6. The number of occurrences of the dataset at the three times slots are 14, 14, and 8.

1) Calculate the likelihood of the null model:

$$L_{null} = Poi(14|\lambda_1 = 8) \times Poi(14|\lambda_2 = 10) \times Pois(8|\lambda_3 = 6);$$

2) Calculate $\Theta' = \{\lambda'_1, \lambda'_2, \lambda'_3\}$: To maximize the likelihood of the alternative model, we multiply λs by (assuming fd=1):

$$p = \frac{14+14+8}{8+10+6} = 1.5;$$

$$\lambda'_1 = 8\times1.5=12,\ \lambda'_2=10\times1.5=15,\ \lambda'_3=6\times1.5=9; \quad (10)$$

3) Calculate the likelihood of the alternative model:

$$L_{alter} = Poi(14|\lambda'_1) \times Poi(14|\lambda'_2) \times Pois(8|\lambda'_3); \quad (11)$$

4) Calculate $\Lambda(s)$ and od:

$$\Lambda(s) = -2\log\left(\frac{L_{null}}{L_{alter}}\right) = 5.19;$$

$$od = \chi^2\_\text{cdf}(5.19, fd = 1) = 0.978; \quad (12)$$

According to Figure 5 B), if setting the threshold of od to 0.95, the three time slots are regarded as an anomaly.

4.1.3 Apply to a Single Set in Multiple Regions

When applying LRT to multiple regions, there are two situations:

In the first situation, a dataset varies in different regions consistently. For example, when an event happens, the volume of social media posted by people may increase simultaneously in the regions around the place where the event is happening. In this case, different regions can share the same new parameter space Θ', therefore having a collective od calculated based on the process of the 2nd example mentioned above. If a dataset has multiple categories, we calculate a new parameter space for each category according to Equation 10, and then calculate a joint likelihood of multiple categories in different regions by Equation 11.

In the second situation, a dataset changes differently in different regions. E.g. the inflow of taxicabs in some regions increases during an anomaly, while others drop. Thus, we need to calculate a different parameter space Θ' for different entries. As a result, each $< r, t >$ has its own od for a dataset. We then aggregate these individual ods by Equation 13, where m is the number of ods. When a dataset contains multiple properties, e.g. inflow and outflow of traffic, we calculate the od for each property in $< r, t >$ individually and then aggregate them by Equation 13.

$$od(s) = \sqrt{\frac{\sum_i od^2(\{< r_i, t_i >\})}{m}}; \quad (13)$$

Equation 13 aggregates a dataset's different behavior across different regions and time intervals, making different combinations of spatio-temporal entries comparable.

4.2 Deal with Multiple Datasets

When applying LRT to multiple datasets, we face two challenges: 1) different underlying distributions and scales, and 2) the aggregation of anomalous degrees of different datasets.

4.2.1 Dealing with Different Distributions and Scales

1) *Different distributions*. In the field of Stochastic Process, the arrival of a time series is usually assumed to follow a Poisson Process. So, Poisson distributions are widely used to model underlying distributions of spatio-temporal data, e.g. the arrival of a 311 complaint or a report of disease in a region. For some datasets, like the mobility of taxicabs, however, a Poisson distribution is not suitable, because they are often over-dispersed, i.e. the variance of the data is much larger than its mean. To model such datasets, we use a Gaussian distribution with its variance proportional to its mean. A more sophisticated way of choosing a proper model for a given dataset can be done by applying the χ^2 test to the data over a period of time. After knowing the underlying model, we match the occurrences of a dataset against its distribution, turning different datasets' values into a likelihood ratio.

2) *Different scales and densities*. Some datasets, such as the mobility of taxicabs, are relatively dense, while other datasets, like 311 complaints and disease reports, are very sparse. For example, the number of insurance claims within a population for a certain type of risk would be zero-inflated by those people who have not taken out insurance against the risk and thus are unable to claim. If formulating the data in a time series (with 0 denoting no reports and $x_t \neq 0$ standing for the number of observed reports), over 90 percent of entries in the series are zero. The excess zero count data brings trouble to identifying the underlying distribution of a dataset, further affecting the anomaly detection. In the *ST_LRT* model, we tackle the sparsity problem in a dataset by using two strategies: the zero-inflated Poisson model and the topic-word distribution inferred by the *MSLT* model.

The zero-inflated Poisson (ZIP) model concerns a random event containing excess zero-count data in unit time. The *ZIP* employs two components that correspond to two zero generating processes [4]. The 1st process is governed by a binary distribution that generates structural zeros. The 2nd is governed by a Poisson distribution that generates counts, some of which may be zero.

1) $X = 0$, with a probability p;

2) $X \sim \text{Poisson}(\lambda)$, with a probability $1 - p$;

Thus, the data with excess zeros can be modeled as follows:

$$X = 0, \text{with probability } p + (1-p)e^{-\lambda}; \quad (14)$$

$$X = h, \text{with probability } (1-p)\frac{e^{-\lambda}\lambda^h}{h!}; \quad (15)$$

where the outcome variable X has any non-negative integer value, λ is the expected Poisson count; p is the probability of extra zeros.

Using latent topic-word distribution: An instance of a dataset is associated with multiple categories, e.g. different types of complaints or diseases. Though following the same distribution, parameters for different categories can be different. Learning parameters of distributions for different categories in a sparse dataset becomes even more challenging, as we need to further allocate sparse observations into different categories, i.e. a sparser problem in each category. At this moment, the *ZIP* model cannot handle it very well either. To address this issue, for each region, we first learn an overall parameter for a dataset, e.g. the total arrival rate λ of all categories of instances, by using the *ZIP* model (suppose it is a Poisson distribution). We then leverage the latent topic distribution $\boldsymbol{\theta}_d$ and the topic-word distribution $\boldsymbol{\varphi}$ in a region to calculate the proportion of each word $prop(w_i)$ as follows (note that a category is denoted as a word in the *MSLT* model):

$$prop(w_i) = \sum_t \theta_{dt}\varphi_{tw_i}; \quad (16)$$

where θ_{dt} is the distribution of topic t in region d, and φ_{tw} denotes the distribution of topic t on word w. As the *MSLT* model learns $\boldsymbol{\theta}_d$ and $\boldsymbol{\varphi}$ based on multiple datasets, which mutually reinforce each other, it is more accurate to estimate $prop(w_i)$ by Equ-

ation 16 than based on the count of each category. Later, given the overall λ of a region, λ_i of different categories is calculated as:

$$\lambda_i = \lambda \times prop(w_i); \quad (17)$$

Algorithm 2 gives an implementation that learns the distributions of each category of a dataset, concerning the scale of the dataset.

Algorithm 2: Learn_Distributions

Input: A data source s with $s.C =< c_1, c_2, \ldots, c_n >$, $\boldsymbol{\varphi}$, $\boldsymbol{\theta}$ from *MSLT*.
Output: The underlying distributions of each category $s.Dist$.
1. **If** s is sparse (i.e. many zero valued entries, μ, σ close to 0)
2. $\lambda \leftarrow$ *Zero-Inflated Poisson*(s);
3. $s.Dist[i] \leftarrow \lambda \times \sum_t \theta_{dt} \varphi_{tc_i}$;
4. **Else if** variance(s) $\gg mean(s)$
5. $s.Dist \leftarrow$ Gaussian(μ, σ);
6. **Else** $s.Dist \leftarrow$ Poisson(λ).
7. **Return** $s.Dist$;

4.2.2 Aggregate ods of Multiple Datasets

We cannot directly apply distance-based methods to multiple datasets which may have different distributions, densities and scales. To address this issue, we first learn an underlying distribution for a dataset. Then, we calculate the likelihood of the null and alternative models, generating an anomalous degree *od* for each dataset according to the method introduced in Section 4.1.2. Given *od*s of multiple data sources $\boldsymbol{S} = \{s_1, s_2, \ldots\}$, we can represent an anomaly candidate as a point in a $|\boldsymbol{S}|$-dimension space. Then, we perform a two-step outlier detection as follow:

Step 1: Using a skyline detection algorithm [1], we can find the skyline points that are not dominated by other points. Here, "*point A dominates point B'* means every dimension of point *A* has a larger *od* then point *B*. So, *'not dominated'* means we cannot find any other anomaly candidates that simultaneously have a bigger *od* in each dataset than these skyline points. Figure 6 illustrates an example of detecting anomalies using three datasets (i.e. in a three dimension space), where the red solid dots denote skyline points. Since different datasets have different meanings and scales, the skyline algorithm integrates different datasets properly without losing information from each dataset. The skyline-based method captures not only the anomalies in which only one dataset has changed tremendously but also the anomalies where a few datasets changed for a certain degree but not that tremendously.

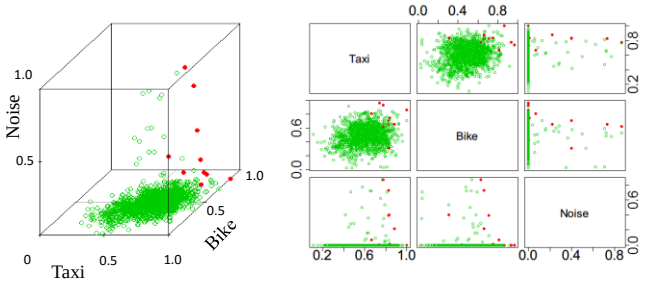


Figure 6. Aggregate multiple *od*s by skyline algorithms

Setp 2: Since we will always find some skyline points from a detection, we need to further check if these skyline points are truly anomalous. A simple solution is to set a threshold for $||\boldsymbol{od}||_2$. Another approach is to deposit the skyline points detected (at the same time of day) from the data over a long period (e.g. in the recent 3 months) in a collection *SLA*. Then, we can check if the distance between a newly detected skyline point and *SLA*'s mean is three times larger than *SLA*'s variance. If so, the skyline point is considered truly anomalous. The Mahalanobis distance, which normalizes the effect of variances along different dimensions, can be used to measure the extremeness of skyline points.

The first step checks the spatial neighbors of an anomaly at current time intervals, while the second step checks its temporal neighbors in the history. The procedure of *ST_LRT* is presented in Algorithm 3, with a setting introduced in Section 6.1.2.

5. Anomaly Candidate Generation

There are many combinations of spatio-temporal entries $< r, t >$ that could satisfy the spatial and temporal constraints (δ_d and δ_t). To find the entry combination is a time-consuming process. In addition, the computational cost of *ST_LRT* is very high. This does not allow us to check many combinations in a short period. To address this issue, we devise an efficient candidate-generation algorithm consisting of two components: 1) A circle-based spatial search and 2) a pruning strategy.

5.1 Checking Spatial Constraint

The circle-based spatial searching algorithm seeks a collections of regions, in each of which any two regions has a distance smaller than the spatial constraint δ_d. The goal of this algorithm can be converted to finding the candidate regions that fall in a circle with a diameter of δ_d , e.g. (r_1, r_3, r_4) depicted in Figure 7 A). This algorithm consists of three major steps:

Step 1: We start from a region, e.g. r_1, searching for its neighbors with a distance to r_1 smaller than δ_d, e.g. $(r_2, r_3, r_4, r_5, r_6)$ shown in Figure 7 A). The distance between two regions is represented by the Euclidian distance between the two regions' centers. This step is further expedited by using an R-Tree spatial index, which organizes the centers of regions with a tree of bounding boxes.

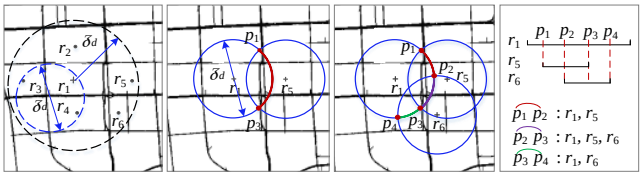


Figure 7. circle-base spatial-search algorithm

Step 2: As demonstrated in Figure 7 B), if the distance between two regions are within δ_d, the center of the circle covering the two regions can be any point on the curve $\overline{p_1 p_3}$. Likewise, as illustrated in Figure 7 C), the center of the circle covering three regions (r_1, r_5, r_6) should lie on the curve $\overline{p_2 p_3}$. We draw such circles between r_1 and its neighbors (returned by Step 1), marking the intersection points between any two circles.

Step 3: We detect the qualified combination of regions by going through the intersection points found in Step 2. As illustrated in Figure 7 D), by moving the red broken line from p_1 to p_4, we can find three region sets: (r_1, r_5), (r_1, r_5, r_6), and (r_1, r_6).

We repeat the three steps for each region, generating a collection of region sets that satisfy the spatial constraint. The collections of regions will be used as a basis to generate entry combinations. The computational complexity of this algorithm is $O(n^2)$. As the spatial coordinates of a region does not change over time, this spatial test can be an offline process.

5.2 Pruning Strategy for Entry Combination

In the second component, we find the combinations of entries of consecutive time slots. For example, (r_1, r_5) of two consecutive time slots have 15 combinations: $\{< r_1, t_1 >\}, \{< r_1, t_1 >, < r_5, t_1 >, \{< r_1, t_1 >, < r_5, t_2 >\}, \ldots, \{< r_1, t_1 >, < r_5, t_1 >, < r_1, t_2 >\}, \ldots, \{< r_1, t_1 >, < r_5, t_1 >, < r_5, t_1 >, < r_1, t_2 >\}$. We can prune the unnecessary combinations based on the following

insight: *If a set of entries' upper bound of od is dominated by existing skyline combinations, all the combinations of its subsets will be dominated by the skyline too*. This component is embedded in the anomaly detection process.

We first calculate the upper bound of od for $\boldsymbol{\mathcal{T}} = \{< r_1, t_1 >, < r_5, t_1 >, < r_5, t_1 >, < r_1, t_2 >\}$. An entry's upper bound of od can be computed by assuming the mean (for an underlying Gaussian distribution) or the λ (for an underlying Poisson distribution) is the observed value. The upper bound of a dataset in $\boldsymbol{\mathcal{T}}$ can then be calculated based on Equation 11 or 13. By putting together the upper bound of od for each dataset, we obtain the final vector $\boldsymbol{od}_{up}$ for multiple datasets. If $\boldsymbol{od}_{up}$ is dominated by any item in the skyline, we can prune $\boldsymbol{\mathcal{T}}$ and do not need to check the combination of its subsets. If $\boldsymbol{\mathcal{T}}$'s $\boldsymbol{od}_{up}$ is not dominated by the skyline, we need to calculate the actual od for each dataset and double check if it can be inserted into the skyline buffer. In this case, we need to check its real od and the three-entry combinations of $\boldsymbol{\mathcal{T}}$. If a three-entry combination is dominated by the existing skyline, we can prune it and do not check its subsets, and so on.

Algorithm 3 details *ST_LRT*, which involves the pruning strategy (Line 18-19). The spatio-temporal constraints have been ensured by Line 4 of Algorithm 1.

Algorithm 3: ST_LRT

Input: Data sources $\boldsymbol{S}$, a collection of spatio-temporal entries $\boldsymbol{\mathcal{T}}'$, a list of skyline outlier degrees SLA detected over a period of time
Output: A set of collective anomalies $\mathcal{A}$.

1. $SkyLine \leftarrow \emptyset$; $\mathcal{A} \leftarrow \emptyset$; $\boldsymbol{od} \leftarrow \emptyset$;
2. **While** $\boldsymbol{\mathcal{T}}' \neq \emptyset$ **Do**
3. Select a $\mathcal{T}$ with the maximum entries from $\boldsymbol{\mathcal{T}}'$;
4. For each $s \in \boldsymbol{S}$
5. **If** s varies in $\mathcal{T}$ consistently
6. $od(s) \leftarrow$ LRT($s.Dist, \mathcal{T}.\boldsymbol{v}$); //refer to Section 4.1.2
7. **Else**
8. **For each** $< r_i, t_i > \in \mathcal{T}$
9. $od(\{< r_i, t_i >\}) \leftarrow$ LRT($s.Dist, < r_i, t_i >.\boldsymbol{v}$);
10. $od(s) \leftarrow \sqrt{\frac{\sum_i od^2(\{< r_i, t_i >\})}{m}}$;
11. $\boldsymbol{od} \leftarrow \boldsymbol{od} || od(s)$;
12. **If** od is NOT dominated by $SkyLine$
13. Insert od into $SkyLine$;
14. Remove points dominated by od from $SkyLine$;
15. Remove $\mathcal{T}$ from $\boldsymbol{\mathcal{T}}'$.
16. **If** $|od - SLA.mean| > 3 \times SLA.variance$;
17. Insert $\mathcal{T}$ into $\mathcal{A}$;
18. **Else if** Upper bound $\boldsymbol{od}_{up}$ is dominated by $SkyLine$;
19. Remove all the spatio-temporal entries $\mathcal{T}$ contains from $\boldsymbol{\mathcal{T}}'$;
20. **Return** $\mathcal{A}$;

6. Experiments and Case Studies

6.1 Settings

6.1.1 Datasets

We evaluate our method with five datasets collected in New York City (NYC): (Detailed in Table 1).

1) *POIs*: In NYC, there are 24,031 POIs of 14 categories: "*Arts & Entertainment*", "*Automotive & Vehicles*", "*Business to Business*", "*Computers & Technology*", "*Education*", "*Food & Dining*", "*Government & Community*", "*Health & Beauty*", "*Home & Family*", "*Legal & Finance*", "*Real Estate & Construction*", "*Shopping*", "*Sports & Recreation*", and *others*. Each POI has a name, category, address and geo-coordinates.

2) *Road network data*: Each road segment is associated with two terminal points and a series of intermediate points, as well as some properties, such as level, capacity and speed limit. Road segments with a level from L_1 to L_5 are used as major roads to partition NYC, resulting in 862 regions.

3) *311 data*: 311 is NYC's governmental non-emergency service number, allowing people in the city to complain about everything that is not urgent by making a phone call, or texting, or using a mobile app. When making a complaint, people are required to provide the location, time, and choose from a category of complaints, such as noise, traffic, or construction. The data is very sparse in each region, as people do not complain about the city anywhere and anytime. Sometimes, they are too busy (or lazy) to make a complaint call. Or, we have very few people in a given region.

4) *Taxicab data*: This dataset is generated by over 14,000 taxicabs in NYC, consisting of two types of information: taxi fare data and trip data. The trip data includes: pick-up and drop-off locations and times, the duration and distance of each trip, taxi ID and the number of passengers, etc. The fare data records the taxi fare, tips and tax of each trip.

5) *Bike renting data*: The data is generated by the bike sharing system in NYC, which has 340 bike stations and about 7,000 bikes. Each record in the data includes the time, bike ID, station ID, and an indication of check-out or return. The location of each station is also disclosed to the public.

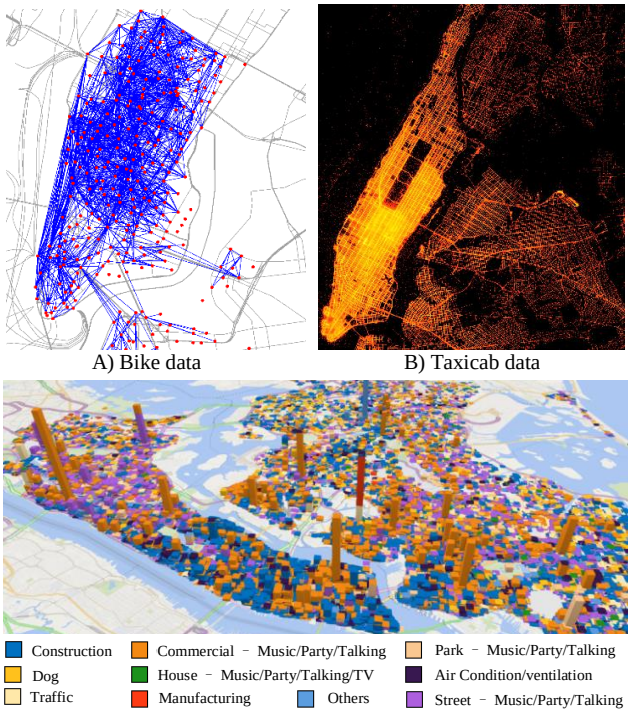
Table 1. Description on datasets

Data sources	Properties	values
Taxicab data 1/1/2014-1/1/2015	number of taxicabs	14,144
	number of trips	165M
	total duration (hour)	36.5M
	total distances (km)	5,671M
Bike Data 1/1/2014-1/1/2015	number of stations	344
	number of bikes	6,811
	number of trips	8,081,216
	total duration (hour)	1.9M
311 Complaints 5/26/2013-12/13/2014	number of categories	10
	number of instances	197,922
Road network 2013	number of nodes	79,315
	number of road segments	32,210
	number of road segments (level>5)	83,655
	number of regions	862
POIs 2013	number of categories	14
	number of instances	24,031

Figure 8 presents the geographical distributions of the taxicab, bike, and 311 data on a digital map. As shown in Figure 8 A), each red point stands for a bike station and a blue edge denotes the aggregation of bike commutes between two stations. To generate a clear graph of stations, we remove the edges with the number of commutes smaller than 700 from 7/1/2013 to 5/31/2014. Figure 8 B) is a heat map of the drop-off and pickup points of all the taxi trips from 1/1/2013 to 12/31/2013. The lighter the denser. As depicted in Figure 8 C), the height of a bar stands for the number of 311 calls that have been made in a particular area. Different colors denote different categories of complaints.

6.1.2 Model Configuration

Settings of MSLT: We project the POIs, bike data, taxicab data and 311 complaints onto the 862 regions; each region is regarded as a document. We aggregate all the 311 data generated on weekdays into a day (and that of weekends into another day), training a different *MSLT* model for them respectively. As illustrated in Figure 9 A), we divide a day into 6 time intervals, counting the number of 311 calls of each complaint category at each time interval in each region. The 311 complaint categories of the six time intervals are regarded as words (i.e. 6×10=60words, $w_1, w_2, \ldots, w_{60}$). The count of 311 calls of a category, at a time interval in a region,



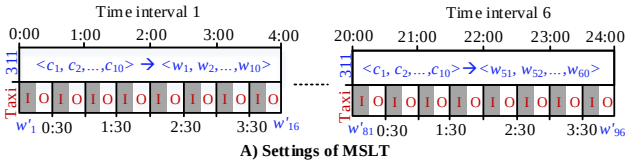
A) Bike data B) Taxicab data

C) 311 complaint data

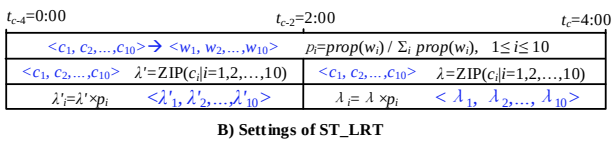
Figure 8. Visualization of the data sources

is deemed as the number of occurrences of a word in a document. Regarding the taxicab data, we count the volume of inflow (I) and outflow (O) in every 30 minutes over a day in a region. So, there are 96 in/out-flows in total, which are deemed to be another word set $(w'_1, w'_2, \ldots, w'_{96})$. The count of each flow in a region is considered a word's number of occurrences in a document. The taxi data on weekdays and weekends are averaged by day respectively. Since the bike stations are mainly located in the south part of Manhattan, we do not involve them in the *MSLT* models for this case study; but, it will be used in the *ST_LRT*. We set the initial value of every entry of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ to 0.1.

Settings of ST_LRT: We perform anomaly detection every hour, based on the taxicab, bike, and 311 datasets of the past 10 hours. Given a 10-hour time span, we partition it into five 2-hour intervals, as illustrated in Figure 9 B). Based on the 311 data arriving every two hours in a region, we use the *ZIP* model to learn a total arrival rate λ for the region at the time interval (e.g. λ for 2:00-4:00 and λ' for 0:00-2:00). Given a 2-hour time interval, we find the words that the categories of the time interval correspond to. For instance, the ten 311 categories at 2:00-4:00 correspond to words $w_1 \sim w_{10}$. We then retrieve the proportion $prop(w_i)$ of each word $(1 \le i \le 10)$ inferred by the *MSLT* model (see Eq. 16), calculating the proportion of a category at 2:00-4:00 by Eq. 18.



A) Settings of MSLT



B) Settings of ST_LRT

Figure 9. Configurations of models

$$p_i = prop(w_i) / \sum_{1 \le i \le 10} prop(w_i). \quad (18)$$

λ_i of time interval 2-4 is then calculated by $\lambda_i = \lambda \times p_i$. We find that 311 data increases or decreases simultaneously across different regions (i.e. consistently), while the taxicab and bike data do not. So, they are handled by Line 6 and Line 8-10 of Algorithm 3, respectively. The mean and variance of the underlying Gaussian distributions for taxicab and bike data are estimated based on the historical data occurring in a time interval.

6.2 Results

6.2.1 *Evaluation on MSLT*

We select some regions with dense 311 data, calculating the distribution of the data across different 311 categories as a ground truth for each region. We then randomly sample the data down to sparsity. Figure 10 shows the KL-Divergence between the estimated distribution and the ground truth in two regions. The horizontal axis denotes 1/*X* data sampled. Estimating the distribution of the 311 data based on the counts of each category results in an increasing KL-Divergence as the sampling percentage decreases. With the help of the *MSLT* model, we find a much smaller KL-Divergence, thereby estimating the distribution more accurately.

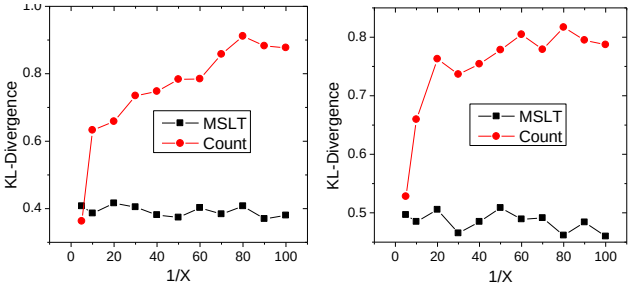


Figure 10. KL-Divergence for the evaluation on the *MSLT*

6.2.2 *Evaluation on ST_LRT*

Evaluating anomaly detection in a real-world setting is an open challenge, as it is impossible to have a full set of ground truth recording all anomalies that have ever happened. In this section, we correlate the anomalies detected by our method with the events that have been reported by *nycinsiderguide.com* over the period from Nov. 1, 2014 to Nov. 30, 2014. Table 2 presents the time and location of the 20 reported events.

We compare our approach with six baselines (shown in Table 3), showing the approach's advantages beyond distance-based methods and those solely using a single dataset. In the distance-based (*DB*) methods, if the distance between an observation and the mean of the data is three times larger than the data's standard deviation, the observation is regarded as an anomaly (for simplicity, we call it the 3-time deviation requirement). *DB-S-Taxi-S* denotes the distance-based (*DB-*) anomaly detection method that identifies an anomaly from a single (*S-*) dataset (i.e. taxi) when either taxi inflow or outflow satisfies the 3-time standard deviation requirement. *DB-S-Taxi-B* means the distance-based method that detects an anomaly from a single dataset (i.e. taxi) but requires both (*-B*) inflow and outflow of taxi data to exceed the 3-time deviation. *DB-M-One* detects an anomaly from multiple datasets (*M-*), as long as one property of a dataset satisfies the 3-time deviation requirement. On the other hand, *DB-M-All* requires all properties of each dataset to satisfy the 3-time deviation requirement. Here, we do not include 311 data as it is very sparse, thus cannot be applied to distance-based anomaly detection methods.

Table 4 presents the results of *ST_LRT* and baseline methods, where the first column denotes the average number of anomalies

detected by a method per day; the second column shows the IDs of events that have been retrieved (from Table 2) by a method. A detected anomaly is regarded as a correct recall if the anomaly has an overlap with a reported event in both spatial and temporal spaces. *ST_LRT* detects 9 out of the 20 events, having a much higher recall than other baselines. Moreover, the number of anomalies detected per day is much smaller than *DB-S-Taxi-S* and *DB-M-One*. Overall, *ST_LRT* outperforms all baselines in terms of precision and recall, detecting about 28 anomalies per day in NYC.

Table 2. Events reported by *nycinsiderguide.com*

	Event Name	Address	Start Time	End Time
1	Bowlloween 2014 New York Halloween	624-660 W 42nd St	10/31/2014 9PM	11/1/2014 2AM
2	Largest Halloween Singles Party in NYC	247 West 37th Street	10/31/2014 7AM	11/1/2014 3AM
3	Kokun Cashmere Sample and Stock Sale	237 W 37th Street	11/5/2014 10:30AM	11/7/2014 5:45PM
4	Big Apple Film Festival	54 Varick St	11/5/2014 6PM	11/9/2014 11PM
5	InterHarmony Concert Series: The Soul of	881 7th Avenue	11/6/2014 8PM	11/6/2014 10PM
6	Hiras Master Tailors New York Trunk Show	301 Park Avenue	11/6/2014 9AM	11/9/2014 1PM
7	in Collaboration with Carnegie Halls	881 Seventh Avenue	11/7/2014 6PM	11/7/2014 10PM
8	Thomas/Ortiz Dance Show	248 West 60th Street	11/7/2014 7PM	11/8/2014 9PM
9	Rebecca Taylor Sample Sale	260 5th Ave	11/11/2014 10AM	11/15/2014 8PM
10	The News NYC Sample Sale	495 Broadway	11/13/2014 9AM	11/15/2014 6AM
11	Giorgio Armani Sample Sale	317 W 33rd St	11/15/2014 9:30AM	11/19/2014 6:30PM
12	Get Buzzed 4 Good Charity Event NYC	200 5th Ave	11/15/2014 1PM	11/15/2014 4PM
13	Ment'or Young Chef Competition	462 Broadway	11/15/2014 2PM	11/15/2014 6PM
14	Gotham Comedy Club	208 West 23rd Street	11/17/2014 6PM	11/17/2014 9PM
15	Kal Rieman NYC Sample Sale	265 West 37th Street	11/18/2014 11AM	11/20/2014 8PM
16	Inhabit Cashmere Sample Sale	250 West 39th St	11/18/2014 10AM	11/20/2014 6 PM
17	Shoshanna NYC Sample Sale	231 W. 39th St	11/19/2014 10AM	11/20/2014 6:30PM
18	ICB / J. Press NYC Sample Sale	530 Seventh Avenue	11/19/2014 12AM	11/21/2014 12AM
19	Thanksgiving in New York City 2014	1675 Broadway	11/27/2014 6AM	11/27/2014 10PM
20	Thanksgiving Day Dinner at Croton Reservoir Tavern	108 West 40th St	11/27/2014 12PM	11/27/2014 9PM

Figure 11 A) presents a collective anomaly, which is comprised of two regions r_1 and r_2 at two successive time intervals (t_1:18-20, t_2: 20-22). We find that this anomaly was caused by *the News NYC Sample Sale* (the 10th event in Table 2), which is a two-day event occurring at blue point *A* shown in Figure 11 A). Figure 11 B)–I) present the in/out-flow of taxicabs and bikes in (r_1, r_2) at (t_1, t_2), where the vertical gray range at each time interval denotes the 3-time standard deviation of the base distribution (learned from historical data at the interval). The black points standing in the middle of each range are the mean of the base distribution. The red points are the real observations in each data source.

According to the data, we find that the event attracted many people from r_1 and r_2 to go shopping after work. That is the reason why the anomaly was detected after 6pm. As the two regions are very close to point *A*, people can travel on foot without taking a taxi or riding a bike as usual. Consequently, the outflows of taxi and bike in the two regions does not increase at the two time intervals t_1 and t_2. Instead, these flows decrease after 8pm, since people can choose to leave for home from place *A* after shopping (without returning to place *A*). In other words, excluding the people going shopping at place *A*, r_1 and r_2 have fewer people departing from there after 8pm.

Table 3. Baseline methods

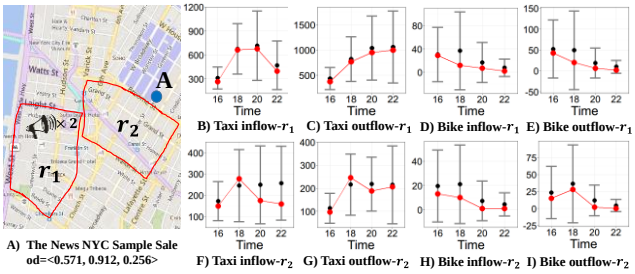
	Taxi Inflow	Taxi Outflow	Bike Inflow	Bike Outflow
Single Dataset	*DB-S-Taxi-S*: one property		*DB-S-Bike-S*: one property	
	DB-S-Taxi-B: both properties		*DB-S-Bike-B*: both properties	
Multi-Datasets	*DB-M-One*: one of the properties satisfying the 3-time deviation			
	DB-M-ALL: all the properties need to satisfy the 3-time deviation			

Table 5 presents the od of each dataset in each spatio-temporal entry for the example illustrated in Figure 11. For example, the od of taxi inflow is 0.274 in spatio-temporal entry <r_1, t_1> and 0.593 in <r_1, t_2>. Calculated by Equation 13, the aggregated od(s) of the taxicab data is 0.404 in r_1 and 0.571 in the two regions. There are two 311 complaints that occur in r_1, resulting in a collective od of 0.256 for the two regions. The final anomalous degree vector $\boldsymbol{od}$ across the three datasets is <0.571, 0.912, 0.256>. Learning from this case, we make three discoveries:

Table 4. Detected anomalies and events hit

Methods	Detected Anomalies/day	Hit Event IDs
DB-S-Taxi-S	336.3	1, 9, 19, 20
DB-S-Bike-B	25.7	9, 19, 20
DB-S-Taxi-S	18.1	4, 19
DB-S-Bike-B	1.83	None
DB-M-One	353.2	1, 4, 9, 19, 20
DB-M-ALL	0.12	None
ST_LRT	28.5	1, 3, 9, 10, 11, 13, 15, 16, 20

1) *Beyond a single dataset*: If checking each single data source individually based on LRT, none of the two regions' od is greater than 0.95 (a threshold for χ^2 test). So, they cannot be detected as an anomaly. After putting the three od(s) in a vector $\boldsymbol{od}$ and applying a skyline detection, we found no other skyline points that can dominate $\boldsymbol{od}$. In short, it is rare to see that the three datasets be that anomalous simultaneously.



A) The News NYC Sample Sale od=<0.571, 0.912, 0.256>
B) Taxi inflow-r_1 C) Taxi outflow-r_1 D) Bike inflow-r_1 E) Bike outflow-r_1
F) Taxi inflow-r_2 G) Taxi outflow-r_2 H) Bike inflow-r_2 I) Bike outflow-r_2

Figure 11. An anomaly caused by The News NYC Sample Sale

2) *Beyond single regions*: If checking r_1 individually, its taxi and bike flows are not that anomalous as compared to its normal patterns. As shown in Table 5, its total od of taxi flow in <r_1, t_1> and <r_1, t_2> is 0.404, which is dominated by other skyline points too. After checking together with r_2, we find the anomalous degree of taxi flow in the two regions becomes larger. In other words, one can barely see the changes in traffic flow of two nearby regions to that extent simultaneously. Detecting anomalies across multiple regions helps identify the spatio-temporal scope impacted by an event. In this case, (r_1, r_2) are affected by the event from 6pm-10pm.

3) *Beyond distance-based method*: As depicted in Figure 11 B)-I), all spatio-temporal entries w.r.t. t_1 and t_2 have a value within the gray ranges (i.e. smaller than the 3-time standard deviation). Thus,

applying distance-based methods to a single dataset cannot identify the anomaly. Simply putting together the values of different datasets into a vector and then applying the Mahalanobis distance cannot help either, because of the different scales and distributions of data. For instance, the 311 data follows a *ZIP* model rather a Gaussian distribution.

Table 5. Computing anomaly degrees for Figure 11

Data sources	Properties	r_1		r_2		od(s)
		t_1	t_2	t_1	t_2	
Taxicab Data	In flow	0.274	0.593	0.822	0.932	0.571
	Out flow	0.383	0.282	0.612	0.202	
	Total	0.404		0.700		
Bike Data	In flow	0.796	0.901	0.932	0.901	0.912
	Out flow	0.872	0.953	0.983	0.987	
	Total	0.882		0.940		
311 Data	Complaint	\	\	\	\	0.256

6.2.3 Efficiency

Using a single core of a server (with a 2.93GHZ CPU and 8GB memory) and the configurations introduced in Section 6.1.2, we can train the *MSLT* model in 21 minutes and perform the circle-based search algorithm in 0.5 seconds. Figure 12 shows the operating time of *ST_LRT* changing over the spatial threshold δ_d (δ_t=4hours). When setting δ_d to 600 meters, we can detect all the collective anomalies in NYC in 3 minutes. The skyline-based pruning further saves about 20% of computational workloads.

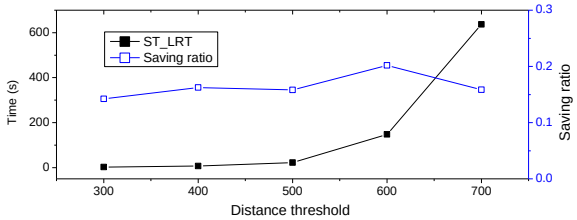


Figure 12. Efficiency study of ST_LRT

7. RELATED WORK

Anomaly detections have been studied extensively over the past decades [2]. In this section, we only study the relevant research that detects anomalies from spatio-temporal data, e.g. detecting outlier trajectories [5][15], identifying traffic anomalies based on trajectories [10][11], diagnosing traffic anomalies [3], and gleaning problematic design in urban planning [6][18]. As one dataset only describes an event from one point of view, many underlying problems cannot be found based on a single source. These techniques cannot be directly adapted to datasets across different domains either, as they cannot handle different distributions and scales of data. E.g. though *LRT* has been used in [11][12] to detect spatial anomalies, these techniques are not concerned with data sparsity and data fusion issues. In our method, data sparsity is handled by the *ZIP* model and the *MSLT* model. Data fusion is implemented by the *MSLT* model and skyline detection. A survey on data fusion methodologies can be found at [16].

Recently, a few research projects have started combining multiple spatio-temporal datasets to detect anomalies [8][9]. For example, Pan et al. [9] first detect the spatio-temporal scope of a traffic anomaly based on vehicles' GPS trajectories and then try to describe the anomaly by using social media that have been generated in the spatial and temporal scope. In the research, the two datasets are used successively rather than simultaneously. There is no organic integration between different datasets. To the best of our knowledge, our method is the first research that detects anomalies from multiple datasets across different regions.

8. CONCLUSION

In this paper, we propose a method to detect collective anomalies from multiple spatio-temporal datasets with different distributions, densities and scales, which is comprised of the *MSLT* model, *ST_LRT* and a candidate generation algorithm. The *MSLT* model fuses multiple datasets to learn the latent functions of a geographical region, which helps in turn to estimate the distribution of sparse datasets. *ST_LRT* first learns a proper distribution to model different datasets, measuring the value of a dataset against its underlying distribution to generate an anomaly degree. *ST_LRT* then uses a skyline-based detection algorithm to identify the final anomalies. Besides the anomalies caused by a single dataset, *ST_LRT* also captures the anomalies where a few datasets have changed for a certain degree but not that tremendously. The detected skyline also helps prune the combinations of spatio-temporal entries. We evaluated our method based on five datasets in NYC, finding the anomalies that can be correlated with public events. The results showcase the advantages of our method beyond approaches using a single dataset in a single region, or using distance-based metrics. With a single machine, we can detect all possible anomalies (with a 600-meter spatial constraint and a 4-hour temporal constraint) in NYC in 3 minutes from data collected over the previous 10 hours.

9. REFERENCES

[1] S., Borzsony, D., Kossmann, K., Stocker. The skyline operator. In *Proc. of ICDE 2011*, pp. 421-430.

[2] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, *ACM Computing Surveys*, volume 41, pp. 1–58.

[3] S. Chawla, Y. Zheng, and J. Hu, Inferring the root cause in road traffic anomalies, In *Proc. of ICDM'12*, pp. 141-150, 2012.

[4] L., Diane. Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics* 34(1) (1992): 1-14.

[5] J. Lee, J. Han, and X. Li, "Trajectory Outlier Detection: A Partition-and-Detect Framework," In *Proc. of ICDE'08*, pp. 140-149, 2008.

[6] W. Liu, Y. Zheng, S. Chawla, J. Yuan, X. Xie. Discovering Spatio-Temporal Causal Interactions in Traffic Data Streams, In *Proc. of KDD'11*, pp. 1010-1018, 2011.

[7] D. Mimno and A. McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. *Uncertainty in Artificial Intelligence*, pages 411–418, 2008.

[8] Y., Matsubara, Y., Sakurai, C. Faloutsos. FUNNEL: automatic mining of spatially coevolving epidemics. *In Proc. of KDD'14*.

[9] B. Pan, et al. Crowd Sensing of Traffic Anomalies based on Human Mobility and Social Media, In *Proc. of GIS'13,* pp. 334-343, 2013.

[10] L. X. Pang, S. Chawla, W. Liu, and Y. Zheng. On Mining Anomalous Patterns in Road Traffic Streams, In *Proc. ADMA'11*, pp. 237-251, 2011.

[11] L. X. Pang, S. Chawla, W. Liu, Y. Zheng. On Detection of Emerging Anomalous Traffic Patterns Using GPS Data, *Data & Knowledge Engineering* 87 (2013): 357-373.

[12] M. Wu, X. Song, C. Jermaine, et al, A LRT framework for fast spatial anomaly detection, *In Proc. of KDD '09*, pp. 887-896.

[13] N. J. Yuan, Y. Zheng, X. Xie. Segmentation of Urban Areas Using Road Networks. MSR-TR-2012-65. 2012.

[14] J. Yuan, Y. Zheng, X. Xie. Discovering regions of different functions in a city using human mobility and POIs. In *Proc. of KDD '12*, pp. 186-194. 2012.

[15] D. Zhang, N. Li, Z. Zhou, et al. iBAT: detecting anomalous taxi trajectories from GPS traces, In *Proc. of UbiComp'11*, pp.99-108.

[16] Y. Zheng, Methodologies for cross-domain data fusion: an overview, ACM Transactions on Big Data, vol. 1, no. 1, 2015.

[17] Y. Zheng, L. Capra, O. Wolfson, H. Yang. Urban Computing: Concepts, Methodologies, and Applications, *ACM Trans. Intelligent Systems and Technology*, vol. 5, no. 3, pp. 38-55, 2014

[18] Y. Zheng, Y. Liu, J. Yuan, X. Xie, Urban Computing with Taxicabs, In *Proc. of UbiComp'11*, pp.89-98, 2011.