PSCluster: Differentially Private Spatial Cluster Detection for Mobile Crowdsourcing Applications

Boyang Hu*†, Baojian Zhou*‡, Qiben Yan†, Adil Alim‡, Feng Chen‡, Huacheng Zeng§
†Computer Science and Engineering Department, University of Nebraska-Lincoln, USA Email: yan@unl.edu
‡Computer Science Department, University at Albany SUNY Email: {bzhou6, aalimu, fchen5}@unl.edu
§Electrical and Computer Engineering Department, University of Louisville Email: huacheng.zeng@louisville.edu
* These authors contributed equally to this work.

***Abstract*—Spatial cluster detection has been employed to identify significant connected spatial clusters in a variety of crowdsourcing applications in bioinformatics, social networks, and network security. However, despite of providing tremendous social benefits, the release of crowdsourced data will pose a considerable threat to mobile users' privacy. Differential privacy has been adopted for privacy preservation of crowdsourced data, yet the effective mining of protected data becomes challenging. In this paper, we investigate the problem of spatial cluster detection on privacy-preserving crowdsourced data. Specifically, we propose *PSCluster*, a differentially private spatial cluster detection mechanism to provide services with data protected under a differential privacy model. *PSCluster*'s key components include a hypothesis testing framework for modeling differentially private spatial data, and a hybrid algorithm that jointly conducts expectation maximization (EM)-based parameter estimation and free-form spatial cluster detection. We evaluate *PSCluster* over synthetic data and real data from mobile crowdsourcing applications, and compare *PSCluster* with two state-of-the-art baseline methods. The experimental results show that *PSCluster* improves the utility without sacrificing privacy.**

I. Introduction

Recently, mobile devices have significant improvements in terms of computing power, memory size, and on-board sensor types. These devices are creating an increasing number of mobile platforms with growing computing and sensing capabilities, which have given rise to *mobile crowdsourcing* [1], a newly-emerged paradigm that allows a crowd of users to participate in the collection of valuable data to perform large-scale data analytics for novel applications. *Spatial cluster detection* has become a useful functionality of mobile crowdsourcing applications, which enables the detection of spatial clusters that present meaningful group/cluster behaviors in a certain area or region, such as disease outbreak, traffic congestion, crime hotspot, *etc*. Spatial cluster detection has been extensively studied in the data mining field [2]–[5] with applications in the field of bioinformatics, social networks, and network security.

Mobile crowdsourcing has the potential to bring innovative applications to provide added value for the society and the contributors in real-time. However, in mobile crowdsourcing applications, the data contributed by individuals can be sensitive data, and the data release can lead to serious privacy leakage. In fact, the crowdsourced data are usually published by crowdsourcing platforms to allow third party data analysts to search for insightful patterns [6], e.g., for the purpose of monitoring traffic congestion, search trends or incidence of influenza, *etc*. Surprisingly, sensitive information including daily routines, medical records, location, social relations can be extracted from the data shared by the crowdsourcing platforms [7], which will disincentivize the users from participating in the crowdsourcing tasks.

Counterintuitively, even the published collective statistics of crowdsourced data used for spatial cluster detection can lead to privacy leakage. In other words, the disclosure of users' collective or aggregated statistical behaviors may compromise the privacy of the individuals, such as the type of disease a patient suffers from, the locations a commuter visits, and the social relations of a mobile user. For example, the disease tracking mobile crowdsourcing app [8] is capable of aggregating crowdsourced disease infection data inside a region. However, for a region containing a small block with only a few households, the aggregated disease infection data may disclose the real identify of patients with a specific type of disease, which clearly violates user privacy. Moreover, by observing disease infection statistics over time, the adversary may be able to identify a patient who is newly infected with the disease.

In consideration of preventing the user privacy leakage from collective statistics while performing spatial cluster detection, we propose *PSCluster*, a differentially Private Spatial Cluster detection mechanism to **identify spatial clusters with privacy-preserving crowdsourced data**. The questions that we want to address in this paper are that: *how to design effective and efficient spatial cluster detection over private data with privacy guarantee? PSCluster* can facilitate many real-world applications, for example: 1) mobile crowdsourcing app based on spatial cluster detection has been developed to identify and predict *disease outbreak*, where the spatial cluster denotes the disease outbreak region. However, the disease infection counts over a certain region are sensitive information that needs to be protected; 2) *traffic congestion monitoring* can be enabled by mobile crowdsourcing to aggregate traffic statistics from multiple mobile devices along the road, where the spatial cluster denotes the congestion area. However, the traffic statistics may leak individual user's route information.

Differential privacy [9], which offers provable guarantees on the amount of information been leaked, has emerged as a compelling privacy model. The crowdsourcing applications apply differential privacy by imposing noise to the raw collected statistics from mobile crowdsourcing to perturb the data. We focus on **the problem of spatial cluster detection with differentially private crowdsourced data**. The challenge is to model the differentially private data, incorporate the model into spatial cluster detection mechanism,

and develop approximation algorithms to approach optimal clustering results. To the best of our knowledge, this is the first approach to detecting free-form spatial clusters in a differential privacy-preserving environment. There are three main technical challenges: 1) **Modeling of spatial clusters with differentially private crowdsourced data**. As the crowdsourced data from individual locations is protected by a differential privacy protocol, it is unclear how the detection of spatial clusters can explicitly handle the data variations caused by the privacy-preserving process. 2) **How can the spatial clusters be detected effectively and efficiently?** The proposed method incorporates a joint process of EM-based parameter estimation and connected subgraph detection, and there is no existing solution that can handle the joint process effectively and efficiently. 3) **Effectiveness at different levels of privacy protection.** Different levels of privacy protection are considered for different applications. How accurate is the spatial cluster detection at different privacy levels?

In this paper, we consider a typical scenario that an application server segments an area into multiple disjoint grids and publish the privacy-preserving population statistics of each grid for crowdsourcing data analysts. The data analysts then apply *PSCluster* to identify spatial clusters for different applications. *PSCluster* utilizes a novel hypothesis testing framework to formulate the privacy-preserving cluster detection problem. Then, *PSCluster* jointly conducts EM-based parameter estimation and free-form connected subgraph detection to perform cluster detection.

Contribution. This paper makes the following contributions:

1) We formulate the problem of spatial cluster detection over perturbed crowdsourced data from mobile devices under a differential privacy model, which aims to provide cluster detection while protecting user privacy.
2) We propose *PSCluster*, the first-known differentially private spatial cluster detection mechanism. As the optimized inference of *PSCluster* is analytically intractable, we design an efficient approximation algorithm for *PSCluster* based on a novel hybrid algorithm of the well-known expectation maximization (EM) framework and projected gradient descent optimization techniques, which are tailored to capture the data perturbation.
3) We conduct comprehensive experiments to validate the effectiveness and efficiency of the proposed techniques in terms of cluster detection and runtime performance, based on two real-world benchmark data sets for disease outbreak and traffic congestion detection.

II. Preliminaries and Problem Formulation

In this section, we introduce some preliminary knowledge on differential privacy, spatial cluster detection, and describe problem settings of our private spatial cluster detection framework. The crowdsourcing platform is shown in Fig. 1, which consists of *mobile users* with crowdsourcing applications, *crowdsourcing data service* infrastructure, and *crowd data analysts*. The mobile users use their mobile devices to participate in the mobile crowdsourcing tasks, contribute data to the crowdsourcing data service infrastructure, and collect data analysis results from the crowd data analysts. Crowdsourcing

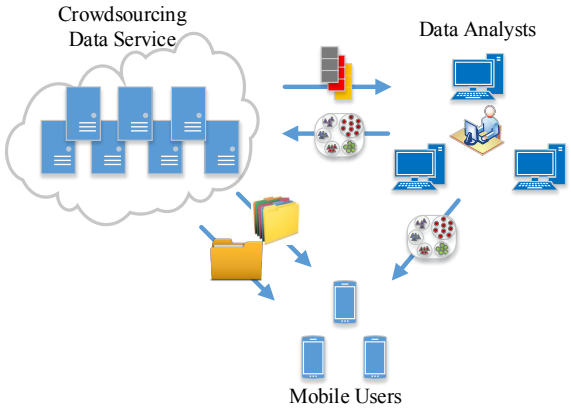


Figure 1: Mobile crowdsourcing platform architecture

data service infrastructure collects data from mobile devices and compute aggregated statistics, and publish the data to crowd data analysts, who perform the identification of significant spatial clusters. To protect user privacy, the aggregated statistics are perturbed by crowdsourcing data service using a differential privacy model.

A. Differential Privacy

Differential privacy has become the standard privacy model for privacy-preserving data analytics, which offers provable guarantees on the amount of information that is leaked. Differential privacy aims to ensure that the output of the algorithm does not significantly depend on any particular individual's data, which provides a quantitative risk assessment for the mobile users to decide whether to participate in the crowdsourcing. A mechanism satisfies ϵ-differential privacy if the output of the mechanism is approximately the same within a ratio e^{ϵ} when any single record in the dataset is removed, added or arbitrarily modified. The ϵ-differential privacy is defined as follows.

Definition 1 (Differential Privacy)**.** *A privacy mechanism $\mathcal{M}$ achieves ϵ-differential privacy, where $\epsilon > 0$, if for any two datasets D and D' differing on at most one record, for all $R \subseteq Range(M)$,*

$$Pr[\mathcal{M}(D) \in R] \leq e^{\epsilon} \cdot Pr[\mathcal{M}(D') \in R]. \tag{1}$$

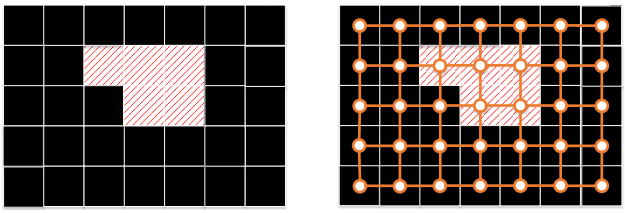
ϵ is the *privacy budget* representing the strength of privacy a mechanism provides. Generally, smaller ϵ represents a stronger privacy strength, which requires a larger perturbation noise. *Laplace mechanism* is the most commonly used mechanism to achieve ϵ-differential privacy [10], which exploits the *sensitivity* of function f, defined as:

$$S(f) = \max_{D,D'} ||f(D) - f(D')||, \tag{2}$$

for all D and D' differing on one record. Intuitively, $S(f)$ captures the maximum changes that could occur in the output of f. The main idea of *Laplace mechanism* is to add noise drawn from a Laplace distribution into the datasets to be published, which is shown in the following definition.

Definition 2 (Laplace Mechanism)**.** *For any function $f : \mathcal{D} \rightarrow \mathcal{R}^d$, the Laplace Mechanism for any dataset $D \in \mathcal{D}$*

$$\mathcal{M}(D) = f(D) + \langle Lap(S(f)/\epsilon) \rangle^d \tag{3}$$



(a) A cluster in a gridded area (b) The grid graph of a cluster

Figure 2: A graph representation of a crowdsourcing map (shaded grids form an arbitrary-shaped spatial cluster with high aggregated statistics)

satisfies ϵ-differential privacy, where the noise $Lap(S(f)/\epsilon)$ is drawn from a Laplace distribution with mean zero and scale $S(f)/\epsilon$.

B. Spatial Cluster Detection

The goal of spatial cluster detection is to identify an optimal cluster of nodes presenting similar cluster features. Spatial cluster detection mechanisms model a spatial data set using a graph. In particular, a graph is given as $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, where $\mathbb{V} = \{1, \cdots, n\}$, n refers to the total number of nodes (spatial regions), and $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$. An edge (i, j) indicates that node i and node j are spatially adjacent. Each node i has a feature that is denoted as z_i. In this paper, the problem of spatial cluster detection is framed as a *hypothesis testing* problem [2], [4], [5] to identify spatial cluster S with arbitrary shapes, which is formulated as follows:

- *Null hypothesis* H_0: $z_i \sim \mathcal{D}(\theta_i)$, where $\mathcal{D}$ refers to a background distribution of this feature when the node i does not belong to a spatial cluster, and θ_i refers to the parameters of the distribution that can be estimated based on historical observations of this feature.
- *Alternative hypothesis* $H_1(S)$: If $i \in S$, then $z_i \sim \mathcal{D}(g(\theta_i, q))$, where $g(\theta_i, q)$ refers to the new parameters of the distribution of z_i that characterize the phenomenon of this spatial cluster S, and q is a multiplicative factor for elevating the mean of the distribution of nodes inside the cluster [2].

The results of the hypothesis testing can be derived by evaluating the Logarithm *Generalized Likelihood Ratio Test* (GLRT) statistic, which can be written in the following form [2]:

$$F(S) = \log \frac{\max_q \prod_{i \in S} \text{Prob}(z_i; g(\theta_i, q))}{\prod_{i \in S} \text{Prob}(z_i; \theta_i)}, \tag{4}$$

where $\text{Prob}(z_i; \theta_i)$ is the *probability density function (pdf)* of z_i given the parameter θ_i. The most significant spatial cluster $S \subseteq \mathbb{V}$ can then be identified by maximizing the above Logarithm GLRT statistic function $F(S)$ over all connected subsets of nodes in $\mathbb{V}$, where the spatial cluster can be arbitrary shape. Subsequently, the decision problem can be formulated as:

$$\max_{S \subseteq \mathbb{V}} F(S) \quad s.t. \quad S \text{ is connected.} \tag{5}$$

We assume that $\mathcal{D}$ refers to a Gaussian distribution[1], and correspondingly, $\theta_i = (\mu_i, \sigma_i)$, where μ_i and σ_i refer to the mean and standard deviation of the Gaussian distribution, respectively; and $g(\theta_i, q) = (q \cdot \mu_i, \sigma_i)$, and $q > 1$. In this scenario, $\mathcal{D}(\theta_i)$ refers to a Gaussian distribution $\mathcal{N}(\mu_i, \sigma_i)$, and $\text{Prob}(z_i; \theta_i)$ refers to the pdf of this Gaussian distribution for z_i given the parameters μ_i and θ_i. Problem (4) then has an analytical form as follows [2]:

$$F(S) = C(S) \log(C(S)/B(S)) + B(S) - C(S), \tag{6}$$

where $C(S) = \sum_{i \in S} \frac{z_i \mu_i}{\sigma_i^2}$ and $B(S) = \sum_{i \in S} \frac{\mu_i^2}{\sigma_i^2}$.

[1]We consider Gaussian distribution in our study because of its popularity in this line of research and its convenience for fast computational inference.

C. Threat Model and Problem Formulation

Each mobile user reports their data to the crowdsourcing data service in real-time. We consider the crowdsourcing data service infrastructure as trusted, but regard the data publishing process as insecure, which is a trust model adopted by other researchers [7], [11]. The adversaries, who can be curious third-party data analysts, can collect the published data and attempt to compromise user privacy, i.e., seek to identify, trace and profile contributors, or link the data records to the corresponding mobile users. The adversaries can launch attacks to identify contributors' sensitive information over a sparse dataset, where only a small number of mobile users inside a grid contribute to the crowdsourcing applications. The adversaries can also keep a database over a long time period to identify the data owner using membership inference attacks [12]. Correspondingly, instead of releasing original crowdsourced data or data statistics, our privacy goal against such adversaries is to provide a sanitized version of data for publishing to achieve ϵ-differential privacy.

In this paper, we consider the popular mobile crowdsourcing applications (such as: disease outbreak prediction, crime hotspot identification, traffic congestion monitoring) where the statistic of each region corresponds to the ***count of users*** inside the region. We segment the area into multiple grids as shown in Fig. 2(a). The two grids are said to be *neighbors* when they are connected by a common frontier. This map of interconnected grids can be further simplified and represented by a graph, where each grid is associated with a node, and when two grids are neighbors, there is an edge connecting the nodes as shown in Fig. 2(b). Every node has a feature corresponding to **count of users/reports** collected by mobile crowdsourcing apps.

The crowdsourcing data service wishes to publish the statistics of a certain crowdsourced data type inside a grid, which can be the number of people who gets a certain type of disease, the number of crimes happened, or the number of cars driving on the road. Let the data be $Z = (z_1, z_2, \cdots, z_n)$, where n is the total number of the grids on a map. To provide mobile user privacy protection, a sanitized version of Z, say $X = (x_1, x_2, \cdots, x_n)$ will be published instead of Z. In order to provide differential privacy, we will add a noise ν_i into the privacy data z_i (i.e., $x_i = z_i + \nu_i$) so that the user can only see the variable x_i instead of z_i, and each noise variable $\nu_i \sim Lap(u_i = 0, b_i = b)$, where $b = S(f)/\epsilon$ is a constant value. The problem we aim to address is:

Problem 1. *Given differentially private crowdsourced data, how to conduct spatial cluster detection with minimum impacts on detection accuracy, effectiveness and efficacy, while satisfying ϵ-differential privacy.*

III. PSCluster: Differentially Private spatial Cluster Detection

In this section, we present *PSCluster*, a private spatial cluster detection scheme to identify significant spatial clusters with differential privacy guarantee. We propose a new hypothesis testing framework for private spatial cluster detection.

A. Hypothesis Testing Framework

Let x_i be the aggregated statistics of all the crowdsourcing data inside grid i after privacy preserving process, i.e. the data to be released, and hence the true feature value z_i is a *latent variable*, which denotes the variable that is hidden from the data analysts to avoid privacy leakage. Considering the statistics as the results of a query Q, since each user can only appear at most one region per time stamp, the sensitivity $S(Q) = 1$. The relation between x_i and z_i can be written as follows: $x_i = z_i + \nu_i$, where ν_i follows a Laplace distribution: $\nu_i \sim \text{Lap}(0, b)$, and $b = \frac{1}{\epsilon}$. Traditional spatial cluster detection techniques are not directly applicable, as none of them has considered the perturbation of the observations. As demonstrated in our experiments in Section IV, a direct application of traditional cluster detection techniques to the perturbed data without specific modeling of the privacy preservation process has limited detection capability.

In this paper, we propose the first-known differentially private spatial cluster detection model using a new hypothesis testing framework:

- Null hypothesis H_0: $x_i|z_i \sim \text{Lap}(z_i, b)$, $z_i \sim \mathcal{N}(\mu_i, \sigma_i)$.
- Alternative hypothesis $H_1(S)$: If $i \in S$, then $x_i|z_i \sim \text{Lap}(z_i, b)$, $z_i \sim \mathcal{N}(q \cdot \mu_i, \sigma_i)$, where the unknown multiplicative constant $q > 1$ indicates a raised mean value of feature distribution for nodes inside the cluster, and needs to be estimated based on the features of nodes in S; otherwise, $x_i|z_i \sim \text{Lap}(z_i, b)$, $z_i \sim \mathcal{N}(\mu_i, \sigma_i)$.

Alternative hypothesis is supported when grid i is included in the statistically significant spatial cluster, while null hypothesis is supported when grid i does not belong to the cluster. According to Eq. (4), the new Logarithm GLRT statistic has the following form:

$$\begin{aligned} F_{DP}(S) &= \log \frac{\max_{q>1} \prod_{i\in S} p(x_i; q)}{\prod_{i\in S} p(x_i)} \\ &= \max_{q>1} \sum\nolimits_{i\in S} p(x_i; q) - \sum\nolimits_{i\in S} \log p(x_i), \end{aligned} \tag{7}$$

where $p(x_i; q) = \int p(x_i|z_i)p(z_i; q)dz_i$. The problem of spatial cluster detection in a privacy preserving environment can then be formulated as:

$$\max_{S\subseteq\mathbb{V}} F_{DP}(S) \quad s.t. \quad S \text{ is connected.} \tag{8}$$

Different from the traditional spatial cluster detection problem in which the Logarithm GLRT statistic function often has an analytical form (e.g., Eq. (6)), the new Logarithm GLRT statistic function as defined above is analytically intractable, and the optimization of Problem (8) is very challenging (to be elaborated below). There is no known algorithm that can be directly applied to solve Problem (8) pertaining to differentially private crowdsourced data. In the following section, we propose an efficient hybrid approximate inference algorithm that integrates a modified EM algorithm for parameter estimation and a graph-structured convex optimization algorithm for free-form cluster (connected subgraph) detection to address the private spatial cluster detection problem.

B. A Hybrid Approximate Inference Algorithm

This subsection first presents a hybrid algorithm that jointly integrates EM-based parameter estimation and connected subgraph detection. In particular, it decomposes Problem (8) to a sequence of subproblems that are easier to solve, then presents an efficient algorithm for solving each subproblem, and finally studies the time complexities of the proposed algorithms.

1) A modified Expectation Maximization (EM) algorithm: It is technically challenging to solve Problem (8), as the objective function $F_{DP}(S)$ (i.e. Eq. 7) has a subproblem that involves the estimation of the multiplicative constant parameter q that is analytically intractable due to the mixture of normal and Laplace distributions. The subproblem is:

$$\max_{q>1} \log \prod_{i\in S} p(x_i; q) = \max_{q>1} \sum_{i\in S} \log \int p(x_i|z_i; q)p(z_i; q)dz_i.$$

The above subproblem is called a *Maximum Likelihood Estimation* (MLE) problem that can be solved using the well-known EM algorithm [13]. EM algorithm is an iterative method for finding maximum likelihood estimates of parameters. However, the standard EM algorithm is not directly applicable as the above subproblem is only part of Problem (8) and has an unknown *set variable* S in addition to the parameter q. We propose a novel EM algorithm to solve Problem (8). We denote t as the t-th iteration. There are two basic steps in each iteration t, including the expectation step and the maximization step. Suppose the estimated factor q in the t-th iteration is denoted as $q^{(t)}$. The **expectation (E) step** refers to the calculation of the lower bound function $Q(q|q^{(t)})$:

$$\begin{aligned} \log \prod_{i\in S} p(x_i; q) &= \sum_{i\in S} \log \int p(x_i, z_i; q)dz_i \\ &= \sum_{i\in S} \log \int p(z_i|x_i; q^{(t)}) \frac{p(x_i, z_i; q)}{p(z_i|x_i; q^{(t)})} dz_i \\ &\geq \sum_{i\in S} \int p(z_i|x_i; q^{(t)}) \log \frac{p(x_i, z_i; q)}{p(z_i|x_i; q^{(t)})} dz_i \\ &\propto \sum_{i\in S} \mathbb{E}_{p(z_i|x_i;q^{(t)})} \Big(\log p(x_i, z_i; q) \Big) \\ &= \sum_{i\in S} \mathbb{E}_{p(z_i|x_i;q^{(t)})} \Big(\ln \frac{1}{2b} e^{-\frac{|x_i - z_i|}{b}} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(z_i - \mu_i q)^2}{2\sigma_i^2}} \Big) \\ &\propto \sum_{i\in S} \frac{2\mu_i q \cdot \mathbb{E}_{p(z_i|x_i;q^{(t)})}[z_i] - \mu_i^2 q^2}{2\sigma_i^2} = Q(q|q^{(t)}), \end{aligned} \tag{9}$$

where the first inequality (Line 3) follows from Jensen's inequality [13]. The expected value of the log likelihood function $\mathbb{E}_{p(z_i|x_i;q^{(t)})}[z_i]$ is expanded (Line 5) according to the following distributions: $x_i|z_i \sim \text{Lap}(z_i, b)$, $z_i \sim \mathcal{N}(\mu_i, \sigma_i)$.

The calculation of $\mathbb{E}_{p(z_i|x_i;q^{(t)})}[z_i]$ is analytically intractable due to the fact that x_i is the summation of a Gaussian random variable and a Laplace random variable that are not conjugate [14]. We apply Laplace approximation and obtain an approximate result as:

$$\mathbb{E}_{p(z_i|x_i;q^{(t)})}[z_i] = \frac{x_i\sigma_i^2 + q^{(t)}\mu_i b^2}{\sigma_i^2 + b^2}. \quad (10)$$

The **maximization (M) step** is to find the factor q that maximizes the quantity:

$$q^{(t+1)} = \arg\max_q Q(q|q^{(t)}). \quad (11)$$

An analytical solution of q can be identified such that the gradient of the above objective function is 0:

$$2\sum_{i\in S}\mathbb{E}_{p(z_i|x_i;q^{(t)})}[\frac{u_i z_i}{2\sigma_i^2}] - 2q\sum_{i\in S}\mathbb{E}_{p(z_i|x_i;q^{(t)})}[\frac{u_i^2}{2\sigma_i^2}] = 0, \quad (12)$$

and we obtain the analytic form of $q^{(t+1)}$ as a function of S as below:

$$q(S) = \frac{\sum_{i\in S}\mathbb{E}_{p(z_i|x_i;q^{(t)})}[\frac{u_i z_i}{2\sigma_i^2}]}{\sum_{i\in S}\mathbb{E}_{p(z_i|x_i;q^{(t)})}[\frac{u_i^2}{2\sigma_i^2}]} = \frac{\sum_{i\in S}\frac{u_i}{2\sigma_i^2}\mathbb{E}_{p(z_i|x_i;q^{(t)})}[z_i]}{\sum_{i\in S}\frac{u_i^2}{2\sigma_i^2}}. \quad (13)$$

Based on the result of the above M step, we obtain an approximated version of Problem (8) using the estimated function $q(S)$ as a subproblem in the current iteration:

$$\max_S \Big(\sum\nolimits_{i\in S}\log p(x_i; q(S)) - \sum\nolimits_{i\in S}\log p(x_i)\Big) \quad (14)$$
$$s.t.\ \ S \text{ is connected.}$$

The basic steps of the modified EM algorithm are shown in Algorithm 1. It is a modified version of the standard EM framework because it has an additional step "Estimation of S" in Step 6 in addition to the standard E and M steps in Steps 4 and 5. Recall that the E and M steps both have an unknown set variable S, which motivates the design of this additional step to estimate and refine S in each iteration.

Algorithm 1: A hybrid algorithm for PSCluster

1 $S^\star = \emptyset$;
2 $t = 0$, $q^{(t)} = 1$;
3 **repeat**
4 **E-step:** Compute $Q(q|q^{(t)})$ according to (9);
5 **M-step:** $q(S) = \arg\max_q Q(q|q^{(t)})$ $= \frac{\sum_{i\in S} u_i/2\sigma_i^2 \cdot \mathbb{E}_{p(z_i|x_i;q^{(t)})}[z_i]}{\sum_{i\in S} u_i^2/2\sigma_i^2}$;
6 **Free-form spatial cluster detection (S):** Identify the intermediate cluster $\hat{S}$ by applying Algorithm 2 to solve Subproblem (14);
7 $q^{(t+1)} = q(\hat{S})$; $S^\star = \hat{S}$;
8 $t = t + 1$;
9 **until** $|q^{(t+1)} - q^{(t)}| < \epsilon$;
10 **return** $S^\star$;

2) An efficient algorithm for detecting free-form spatial clusters: Subproblem (14) is simpler than Problem (8) as the objective function of the former only has the discrete variable S but the objective function of the latter has a mixture of discrete variable S and numerical variable q. However, the problem is still difficult as the objective function is highly nonlinear. An exhaust search for solving Subproblem (14) is impractical as the total number of all possible subsets (S) is exponential with respect to the total number of nodes. In fact, the problem is known to be NP-hard and does not admit any constant-factor approximations even when the objective function is a linear function, via a reduction from the net worth prize-collecting Steiner tree problem [15]. To the best of our knowledge, there is no existing discrete optimization algorithm that can be applied to solve this problem subject to a connectivity constraint. Therefore, we explore efficient numerical optimization techniques to approximately solve a relaxed version of this problem. We first represent S using the vector form $y \in \{0,1\}^n$ such that $S = \{i \mid y_i = 1\}$. The function $q(S) = q(y)$ can then be represented as:

$$q(y) = \frac{y^T[\frac{u_1}{2\sigma_1^2}\mathbb{E}_{p(z_i|x_i;q^{(t)})}[z_1], \cdots, \frac{u_n}{2\sigma_n^2}\mathbb{E}_{p(z_i|x_i;q^{(t)})}[z_n]]^T}{y^T[\frac{u_1}{2\sigma_1^2}, \cdots, \frac{u_n}{2\sigma_n^2}]^T}.$$

Given the fact that

$$p(x_i; q(y)) = \int p(x_i|z_i)p(z_i; q(y))dz_i,$$

and $p(x_i|z_i)$ and $p(z_i; q(y))$ are Gaussian and Laplace density functions, respectively, the analytical form of $p(x_i; q(y))$ is known as [14]:

$$\log(p(x_i; q(y))) \quad = -\log(2\sqrt{b}) - \tfrac{\mu_i^2 q(y)^2}{2\sigma_i^2} + C_i(q(y))$$

and

$$\log(p(x_i)) \quad = -\log(2\sqrt{b}) - \tfrac{\mu_i^2}{2\sigma_i^2} + C(1),$$

where

$$C_i(q(y)) = \Big[e^{\frac{x_i}{b} + \frac{(q(y)\mu_i - \sigma_i^2)^2}{2b\sigma_i^2}} \cdot \Phi\Big(-\frac{x_i - q(y)\mu_i + \sigma_i^2}{\sigma_i\sqrt{b}}\Big) + e^{-\frac{x_i}{b} + \frac{(q(y)\mu_i + \sigma_i^2)^2}{2b\sigma_i^2}} \cdot \Phi\Big(\frac{x_i - q(y)\mu_i - \sigma_i^2}{\sigma_i\sqrt{b}}\Big)\Big],$$

and $\Phi(\cdot)$ refers to the cumulative density function of a Gaussian distribution. Also, the objective function of Subproblem (14) can then be reformulated as:

$$f_{DP}(y) = \langle y^{(t)}, [\log(p(x_1; q(y))), \cdots, \log(p(x_n; q(y)))]\rangle - \langle y^{(t)}, [\log(p(x_1)), \cdots, \log(p(x_n))]\rangle, \quad (15)$$

where $\langle\cdot,\cdot\rangle$ is a dot product operator between two vectors, $y^{(t)}$ is the estimate of y in the previous iteration, and its gradient $\nabla f_{DP}(y)$ can be calculated based on the above analytical form of $f_{DP}(y)$. We then focus on the following relaxed version of Subproblem (14):

$$\max_{y\in[0,1]^n} f_{DP}(y) \quad s.t.\ \ \text{supp}(y) \text{ is connected}, \quad (16)$$

where y takes values from the continuous domain $[0,1]^n$ instead of the discrete domain $\{0,1\}^n$, and $\text{supp}(y) = \{i \mid y_i > 0\}$ refers to the indices of non-zero entries. As each entry in y relates to a node in the network, $\text{supp}(y)$ indicates the set of nodes that belong to the spatial cluster, and thus is the expected solution. Given the solution y to the above relaxed problem, the cluster will be identified as $\text{supp}(y)$ that is guaranteed to form a connected subgraph.

We propose an efficient algorithm to solve Problem (16) based on projected (or model-based) gradient descent (PGD) optimization [16]. PGD is different from the traditional gradient descent optimization in that PGD involves an additional projection process that finds the best approximation of an intermediate solution b in the constrained space $\{y \mid \text{supp}(y) \text{ is connected.}\}$:

$$\text{P}(b) = \arg\min_{y \in [0,1]^n} \|y - b\|_2^2 \quad s.t. \ \ \text{supp}(y) \text{ is connected.} \quad (17)$$

As the above projection is NP-hard due to a reduction from the classical Steiner tree problem, two nearly-linear time approximations have been designed to iteratively find the best solution. Curious readers please refer to [17] for detailed description.

- **Tail approximation** ($\text{T}(b)$): Find $y \in [0,1]^n$, such that

$$\|y - b\|_2^2 \leq c_T \cdot \min_{\text{supp}(\hat{y}) \text{ is connected}} \|\hat{y} - b\|_2^2, \quad (18)$$

- **Head approximation** ($\text{H}(b)$): Find $y \in [0,1]^n$, such that

$$y = b_S, \|b_S\|_2^2 \geq c_H \cdot \max_{\hat{S} \text{ is connected}} \|b_{\hat{S}}\|_2^2, \quad (19)$$

where c_H (set as $\sqrt{1/14}$) and c_T (set as $\sqrt{7}$) are arbitrary, fixed constants.

The basic steps are shown in Algorithm 2, which demonstrates an iterative process that bounces between upper bound (Head approximation) and lower bound (Tail approximation) to approach an optimal result for Subproblem (14) [16]. The first step (Line 3) in each iteration, $b = y^i + \text{H}(\nabla f_{DP}(y^i))$, identifies an updated version of y^i along the direction defined by the projected gradient $\text{H}(\nabla f_{DP}(y^i))$, in which pursuing the maximization will be most effective. As the updated version b is not guaranteed to satisfy the constraint that $\text{supp}(b)$ is connected, Step 4 identities an approximation of b in this constrained space $\{y \in [0,1]^n \mid \text{supp}(y) \text{ is connected}\}$ using the tail approximation process. The iterations terminate when the change of the estimated maximum from the previous iteration is less than a predefined small threshold, e.g. 0.0005 (note that any small value would work).

The convergence analysis of Algorithm 2 can be found in [18]. It can be readily shown that the objective value of Subproblem (14) based on the intermediate solution is monotonically increasing at each iteration. The experimental results in [18] demonstrate that this algorithm converges in a small number of iterations for general nonlinear cost functions, and outperforms state-of-the-art algorithms for detecting arbitrary-shape spatial clusters. Our experiments in this paper further confirm the superior performance of this algorithm with our unique score function f_{DP} (Eq. 15).

Algorithm 2: An efficient algorithm for Subproblem (14)

1 $i = 0$, $y^i = 0$;
2 **repeat**
3 $\quad b = y^i + \text{H}(\nabla f_{DP}(y^i))$;
4 $\quad y^{i+1} = \text{T}(b)$;
5 $\quad i = i + 1$;
6 **until** $\|y^{i+1} - y^i\| < \varepsilon$;
7 **return** $\text{supp}(y^{i+1}) = \{j \mid y_j^{i+1} > 0\}$;

C. Time Complexity Analysis

Algorithm 1 is the overall algorithm, and its Step 6 is implemented by Algorithm 2. As we obtain the analytical forms for both the E and M steps in Eq. (9) and Eq. (13), respectively, the overall time cost for these two steps is $O(1)$. Thus, the time cost of Algorithm 1 is mainly decided by the time cost of Step 6 (Algorithm 2). For Algorithm 2, the time cost to calculate the gradient $\nabla f_{DP}(y^i)$ in Step 3 is $O(n)$, where n refers to the total number of nodes in the input graph. The time costs of head approximation in Step 3 and tail approximation in Step 4 are both $O(m \log^3 n)$ [17], where m refers to the total number of edges.

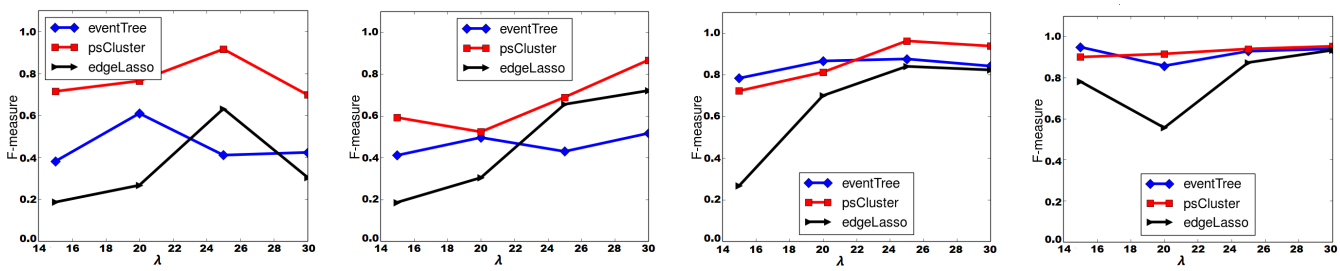
We denote the total number of iterations in Algorithm 1 as L_1 and the total number of iterations in Algorithm 2 as L_2. In mobile crowdsourcing applications, the input graph is defined based on spatial regions (e.g., grid cells) as nodes, and their spatial adjacencies as edges. In this scenario, the input graph is a planar graph and the total number of all possible edges is at most $3n - 6$ [19]. The total time cost of Algorithm 1 can then be calculated as $O(L_1 \cdot L_2 \cdot (n \log^3 n + n))$. In practice, as demonstrated in our experiments, both Algorithm 1 and Algorithm 2 converge fast, and their numbers of iterations L_1 and L_2 are small with respect to n. Considering all the previous practical factors, the time cost of Algorithm 1 is $O(n \log^3 n)$ and scales nearly-linearly with respect to n.

The convergence analysis of Algorithm 1 is complicated due to the design of Step 6, which requires solving a non-convex optimization problem (Problem (14)), in addition to the modified Expectation and Maximization steps in Step 4 and Step 5. We empirically demonstrate the fast convergence of Algorithm 1 using experiments in Section IV-B. In future, we will conduct theoretical analysis to prove its convergence.

IV. Evaluation

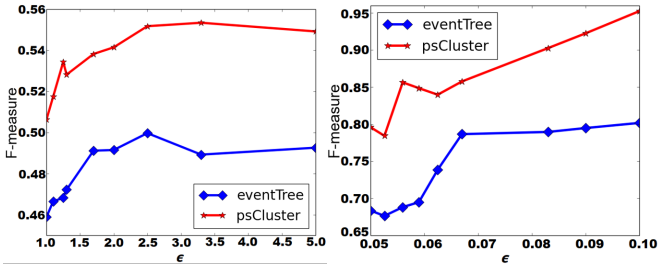
The quality of the significant cluster detection in a graph $\mathbb{G}$ can be measured using three metrics, including precision, recall, and F-measure. Denote the detected cluster as $S \subseteq \mathbb{V}$, and the true cluster as $S^\star$, where the cluster can be disease outbreak region, traffic congestion region, *etc*. The three metrics are defined as:

$$\text{precision}(S) = \frac{S \cap S^\star}{|S|}, \text{recall}(S) = \frac{S \cap S^\star}{|S^\star|},$$
$$\text{F-measure} = 2 \cdot \frac{\text{precision}(S) \cdot \text{recall}(S)}{\text{precision(S)+recall(S)}}. \quad (20)$$



(a) F-measure ($\epsilon = 0.2$) (b) F-measure ($\epsilon = 0.25$) (c) F-measure ($\epsilon = 0.4$) (d) F-measure ($\epsilon = 1$)

Figure 3: Utility comparison with varying λ using synthetic data



(a) Disease Outbreak Data (b) Taxi Traffic Data

Figure 4: Utility comparison using real data

Specifically, F-measure represents a combined score of precision and recall, which we use as a main performance metric. The evaluation results in terms of precision and recall are omitted due to space limitations. Since there is no existing private cluster detection algorithms available, we implement two well-known baseline methods for cluster detection in noisy data for comparison: *EventTree* [20] and *EdgeLasso* [21]:

- *EventTree* reformulates the detection problem as a variant of prize-collecting Steiner tree (PCST) problem and applies fast approximation algorithms of the PCST problem [15] to detect the most significant *compact* spatial cluster in the input graph, where the *compactness* is defined based on a minimum-distance tree of the cluster.
- *EdgeLasso* reformulates the cluster detection problem as a generalized fused lasso problem and applies the state-of-the-art sparse learning techniques to detect the most significant free-form spatial cluster in the input graph.

Based on previous research [20], [21], these two baseline methods outperform existing methods on both running time and accuracy, which also have been tested to work well with noisy data. They simply treat the private data as noisy data to carry out cluster detection. In this section, we evaluate the spatial cluster detection performance of *PSCluster* using synthetic data generated from Poisson distribution, and real data from disease outbreak and taxi traffic applications.

A. Evaluating private spatial cluster detection performance

Evaluation Using Synthetic Data: We generate synthetic data using Poisson distribution (i.e. $f(z;\lambda) = Pr(X = z) = \frac{\lambda^k e^{-\lambda}}{k!}$), since Poisson distribution is one of the most popular distribution for count data in a geographic map [22]. Each data point corresponds to the count of crowdsourced data inside one grid. Then Laplace distributed noise is added to perturb the data point. The normal data is drawn from Poisson distribution with $\lambda = 5$. The data points forming the significant spatial cluster are drawn from Poisson distribution with varying λ in the range of $[15, 30]$. The graph consists of 20×20 grids with 400 nodes. The size of the cluster is set to 5% of the graph size, and the nodes of the cluster are generated using random walk. The performance metrics are computed using Eq. (20).

We evaluate the private spatial cluster detection performance of *PSCluster* in terms of F-measure. We compare the performance against two baseline methods. The F-measure performance with respect to λ is shown in Fig. 3. From Fig. 3(a) to Fig. 3(d), we change the parameter λ with a fixed ϵ. We can see that when ϵ is small, *PSCluster* outperforms two baseline methods. However, as ϵ gets bigger, the F-measure performance of *PSCluster* gets close to that of the *EventTree* baseline method. As mentioned previously, smaller ϵ indicates a stronger privacy strength which translates into a larger perturbation noise. The results show that *PSCluster* performs better than other methods *in case of a larger perturbation noise*, because we take the noise into account in the algorithm design, and strive hard to remove the impacts of noise during spatial cluster detection. As shown from the results, the performance of baseline methods deteriorates when the perturbation noise gets larger. In the case when the perturbation noise is low, the baseline methods show their strengths in detecting significant cluster, and *PSCluster* can also maintain a high cluster detection performance. These results validate *PSCluster*'s superior cluster detection performance under a high level of differential privacy protection. Although the F-measure could drop around 0.6 in some cases, we note that *PSCluster* can be more resilient against perturbed data with large added noise compared with other methods.

Evaluation Using Disease Outbreak Data: We also evaluate *PSCluster* using real data from disease outbreak research [23] for disease outbreak detection. The data corresponds to New York city benchmark data, which is a collection of public benchmark datasets used for the evaluation and comparison of early disease detection methods. Geographic coordinates (representing the approximate center of each zip code) and population numbers for 176 NYC zip codes are used for these datasets. Among five New York City boroughs, we use Queen Region with 63 zip codes. After a careful initial examination of the data, we find that at a certain disease outbreak day, node count inside outbreak region (Queens) has mean of 2.34, and standard deviation of 1.15. On the other hand, outside the

outbreak region, the node count has mean of 0.52, and standard deviation of 0.74. As the count value is small, the perturbation noise should also be small. Therefore, we let perturbation noise follow $Laplace(0, b)$, where $b \in [0.2, 1]$ (i.e., $\epsilon \in [1, 5]$).

Here, we only compare *EventTree* with *PSCluster* to prevent duplications. Fig. 4(a) shows the F-measure performance of *PSCluster* outperforms that of *EventTree* with varying ϵ. The increasing ϵ (i.e., reducing noise) improves the performance of *PSCluster*. The F-measure curves are not smooth due to the variability of the data outbreak data. The slight decline in F-measure curve of *PSCluster* (when ϵ is 1.4) is caused by the large and randomly generated noise. However, *PSCluster* can always detect disease outbreak clusters more accurately. This indicates that *PSCluster* is able to achieve superior utility using real world data sets even when the added noise is large, by modeling the data perturbation of differential privacy preservation.

Evaluation Using Taxi Traffic Data: The traffic data set is collected for taxi pickup data analytics in 01/01/2010 in New York City [24]. We use spatial cluster detection to detect congestion areas. We randomly select a region in the urban area of New York City with four vertex coordinates. The size of the region is $16,646 \times 6,840\ meters^2$. We evenly separate the region into 100×100 grid cells, with each grid spanning $166 \times 68\ meters^2$. The average count of taxi pickups inside the grid is 22.92, with the standard deviation as 5036.03. Total count of taxi pickups inside the region is 229,230. Inside the cluster grids (i.e. traffic congestion grids), the count value has mean of 202.55 with standard deviation of 140.78, while outside the cluster grids, the count value has mean of 6.04 with standard deviation of 18.22. We allow the perturbation noise to be comparable to the count value, which follows $Laplace(0, b)$, where $b \in [10, 20]$ (i.e., $\epsilon \in [0.05, 0.1]$).

Fig. 4(b) shows the F-measure performance comparison w.r.t. ϵ, where we can see *PSCluster* outperforms *EventTree* significantly with different levels of privacy protections. The high data variability causes a non-monotonic increase of *PSCluster*'s F-measure performance with the increasing ϵ (i.e., reducing noise), and we find that *PSCluster* can even reach 95% of F-measure when ϵ is 0.1. The minor performance drop with small ϵ is again caused by the large added perturbation noise that complicates the spatial cluster detection. Comparing Fig. 4(b) with Fig. 4(a), we can see the cluster detection rate for disease outbreak is much lower, since the disease outbreak data itself is more noisy. In summary, we show that *PSCluster* has superior spatial cluster detection performance with data perturbation at different privacy protection levels, significantly outperforming the state-of-the-art methods.

Next, we evaluate the detection performance of *PSCluster* w.r.t. the grid size. We separate the entire region into different number of grids: 10×10, 25×25, 50×50, and 100×100, corresponding to different grid sizes from large to small. The F-measure performance of *PSCluster* with different grid numbers (grid sizes) is shown in Fig. 5, and we have three major observations. First, F-measure plateaus after ϵ exceeds a certain value, because a larger ϵ represents a smaller noise, which imposes less impacts to the cluster detection performance. Second, a smaller grid number indicates a larger grid size, and thus a larger taxi pickup counts. Therefore, with a smaller grid number, only larger noise (smaller ϵ) will have a negative impact. Third, F-measure improves with the decreasing grid size (or increasing grid number). The reason is as follows: consider a large grid that contains several small grids. In case that only a small number of small grids are included in the cluster, the large grid will be excluded from the cluster, which causes a reduction in recall. Otherwise, in case that this large grid is indeed included in the cluster, it will cause a degradation in precision. Both two cases present a degradation in F-measure performance with large grids. Also, inside small grid size (100×100 grid map), the counts are smaller, which leads to a performance fluctuation when noise is relatively large (i.e., $\epsilon \in [0.01, 0.02]$). Based on all the above experiments, we show that our algorithms converge in all cases.

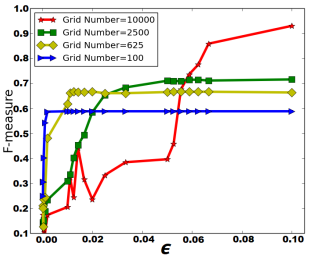


Figure 5: Cluster detection performance w.r.t. grid size

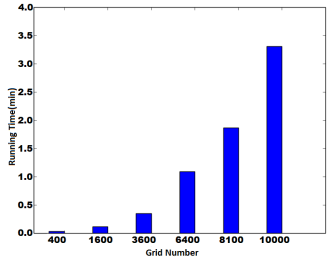


Figure 6: Runtime performance of PSCluster

B. Evaluating Runtime Performance

Table I compares the runtime of our method with two baseline methods based on the synthetic data set in a 20×20 grid map. The average runtime is assessed by calculating the average of computational time consumption of each method over 50 rounds. Among these 50 rounds, we vary ϵ from $[0.2, 1]$, and change λ from $[15, 30]$. It is noteworthy that our method can always converge to yield results, and the runtime also denotes the convergence time. We note that these two baseline methods are among the most efficient methods in the current literature for spatial cluster detection. The results indicate that the runtime of our method is comparable to the baseline methods, although our method achieves significant improvement of cluster detection performance by addressing the spatial cluster detection problem in the privacy-preserving setting that is technically more challenging as discussed in Section III. Fig. 6 shows the runtime of our method w.r.t. synthetic data of different grid sizes. The results indicate that our method can scale to support large grid numbers.

V. Related Work

Private Preserving Mobile Crowdsourcing: Mobile crowdsourcing has attracted attentions of the research community, as it enables mobile users to fully utilize the sensing and computing capabilities of mobile devices to participate and accomplish important crowdsourcing tasks. Recently, privacy-preserving problem has been studied in mobile crowdsourcing systems. SPPEAR [25] uses anonymization networking and group signature schemes to protect user privacy. Miao *et al.* [26] investigate privacy-preserving truth discovery to take user reliability into account, which is complementary to our work.

Table I: Runtime performance (average runtime over 50 rounds)

EventTree (seconds)	*EdgeLasso* (seconds)	*PSCluster* (seconds)
0.901	1.376	1.642

Kellaris *et al.* [11] propose w-event privacy, which is based on differential privacy, to prevent any event sequence disclosure occurring over a time-series data stream. RescueDP [7] builds upon w-event privacy to design a private crowdsourced data publishing mechanism on spatial-temporal data. In this paper, we consider a new problem of spatial cluster detection over differentially private data generated by crowd users and published by crowdsourcing applications.

Spatial Cluster Detection: Exhaustively searching spatial clusters is impractical, as the total number of possible spatial clusters scales exponentially with respect to the total number of spatial regions (n). To avoid this computational bottleneck, a number of methods are designed to identify relatively simple clusters of fixed shapes, such as rectangles and circles, and the total number of possible spatial clusters scales quadratically with respect to n [2], [4], [5]. In recent years, more advanced methods have been proposed to detect free-shape spatial clusters, which model the spatial regions using a generalized graph, and represent a spatial cluster as a connected sub-graph. In particular, Speakman *et al.* [27] propose a heuristic algorithm to identify the highest scoring connected subgraphs based on shortest paths in a given graph structure with real-valued weights at each node. Sharpnack *et al.* [21] consider a generalized likelihood ratio function, a type of parametric scan statistic, as the score function, and present *EdgeLasso*, a sparse learning method based on edge-lasso regularization. Rozenshtein *et al.* [20] present *EventTree*, a fast algorithm based on approximation algorithms of the prize-collecting Steiner tree problem for detecting *compact* spatial clusters. However, none of these methods has considered privacy protection of spatial data.

VI. Conclusion

In this paper, we propose *PSCluster*, a privacy-preserving spatial cluster detection scheme under a differential privacy protection model. We design a framework for *PSCluster* consisting of scan statistics, and a novel hybrid algorithm integrating the EM algorithm and projected gradient descent optimization. We apply *PSCluster* to detect clusters with two different types of real-world mobile crowdsourcing application data. Extensive experiments show that *PSCluster* outperforms state-of-the-art methods and improves the utility of spatial cluster detection with privacy guarantee.

Acknowledgement

This work was supported in part by the US National Science Foundation under grants CNS-1566388, CNS-1717898, and CNS-1731833. The work of Feng Chen was partially supported by the following grants: NSF IIS-1750911, NSF IIS-1441479, and ARO W911NF1720129. The views and opinions of the author(s) do not reflect those of the Department of Defense or Assistant Secretary of Defense for Research and Engineering.

References

[1] J. Ren, Y. Zhang, K. Zhang, and X. Shen, "Exploiting mobile crowdsourcing for pervasive cloud services: challenges and solutions," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 98–105, March 2015.

[2] D. B. Neill, "Fast subset scan for spatial pattern detection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 74, no. 2, pp. 337–360, 2012.

[3] D. B. Neill and A. W. Moore, "Rapid detection of significant spatial clusters," in *Proc. of SIGKDD*, 2004.

[4] D. B. Neill, "Expectation-based scan statistics for monitoring spatial time series data," *International Journal of Forecasting*, vol. 25, no. 3, pp. 498–517, 2009.

[5] ——, "An empirical comparison of spatial scan statistics for outbreak detection," *International Journal of Health Geographics*, vol. 8, no. 1, p. 1, 2009.

[6] G. Acs and C. Castelluccia, "A case study: Privacy preserving release of spatio-temporal density in paris," in *Proc. of SIGKDD*, 2014.

[7] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, "Rescuedp: Real-time spatio-temporal crowd-sourced data publishing with differential privacy," in *Proc. of INFOCOM*, April 2016.

[8] "Healthmap," http://www.healthmap.org/, Accessed on July 29, 2017.

[9] C. Dwork, "Differential privacy," in *Proc. of ICALP*, 2006, pp. 1–12.

[10] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the Third Conference on Theory of Cryptography*, ser. TCC'06, 2006, pp. 265–284.

[11] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias, "Differentially private event sequences over infinite streams," *Proc. of VLDB Endow.*, vol. 7, no. 12, pp. 1155–1166, Aug 2014.

[12] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*, May 2017, pp. 3–18.

[13] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal processing magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[14] S. Nadarajah and S. Kotz, "On the linear combination, product and ratio of normal and laplace random variables," *Journal of the Franklin Institute*, vol. 348, no. 4, pp. 810–822, 2011.

[15] D. S. Johnson, M. Minkoff, and S. Phillips, "The prize collecting steiner tree problem: theory and practice," in *SODA*, vol. 1. Citeseer, 2000, p. 4.

[16] C. Hegde, P. Indyk, and L. Schmidt, "Approximation algorithms for model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 61, no. 9, pp. 5129–5147, 2015.

[17] ——, "A nearly-linear time framework for graph-structured sparsity," in *Proc. of ICML*, 2015, pp. 928–937.

[18] F. Chen and B. Zhou, "A generalized matching pursuit approach for graph-structured sparsity," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16)*, 2016.

[19] T. Nishizeki and N. Chiba, *Planar graphs: Theory and algorithms*. Elsevier, 1988, vol. 32.

[20] P. Rozenshtein, A. Anagnostopoulos, A. Gionis, and N. Tatti, "Event detection in activity networks," in *Proc. of SIGKDD*, 2014.

[21] J. Sharpnack, A. Singh, and A. Rinaldo, "Sparsistency of the edge lasso over graphs." in *AISTATS*, vol. 22, 2012, pp. 1028–1036.

[22] D. A. Griffith and R. Haining, "Beyond mule kicks: The poisson distribution in geographical analysis," *Geographical Analysis*, vol. 38, no. 2, pp. 123–139, 2006.

[23] M. Kulldorff, Z. Zhang, J. Hartman, R. Heffernan, L. Huang, and M. F., "Evaluating disease outbreak detection methods: Benchmark data and power calculations," *Morbidity and Mortality Weekly Report*, vol. 53, pp. 144–151, 2004.

[24] B. Donovan and D. Work, "New york city taxi data," https://uofi.app.box.com/v/nyctaxidata, Accessed on July 29, 2017.

[25] S. Gisdakis, T. Giannetsos, and P. Papadimitratos, "SPPEAR: Security & privacy-preserving architecture for participatory-sensing applications," in *Proc. of WiSec*, 2014, pp. 39–50.

[26] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, and K. Ren, "Cloud-enabled privacy-preserving truth discovery in crowd sensing systems," in *Proc. of SenSys*, 2015, pp. 183–196.

[27] S. Speakman, Y. Zhang, and D. B. Neill, "Dynamic pattern detection with temporal consistency and connectivity constraints." in *Proc. of ICDM*, 2013, pp. 697–706.