Whom to Query? Spatially-Blind Participatory Crowdsensing under Budget Constraints

Mai ElSherief [*], Morgan Vigil-Hayes[‡], Ramya Raghavendra[†], Elizabeth Belding[*]
[*]Dept. of Computer Science, UC Santa Barbara
[‡]School of Informatics, Computing, and Cyber Systems, Northern Arizona University
[†]IBM T. J. Watson Research Center
{mayelsherif, ebelding}@cs.ucsb.edu, morgan.vigil-hayes@nau.edu, rraghav@us.ibm.com

ABSTRACT

The ubiquity of sensors has introduced a variety of new opportunities for data collection. In this paper, we attempt to answer the question: Given M workers in a spatial environment and N probing resources, where $N < M$, which N workers should be queried to answer a specific question? To solve this research question, we propose two querying algorithms: one that exploits worker feedback (DispNN) and one that does not rely on worker feedback (DispMax). We evaluate DispNN and DispMax algorithms on two different event distributions: clustered and complete spatial randomness. We then apply the algorithms to a dataset of actual street harassment events provided by Hollaback. The proposed algorithms outperform a random selection approach by up to 30%, a random selection approach with feedback by up to 35%, a greedy heuristic by up to 5x times, and cover up to a median of 96% of the incidents.

1 INTRODUCTION

The ubiquity of mobile phones and sensors has brought participatory sensing into daily life. Participatory sensing can be defined as "*the process whereby individuals and communities use ever-more-capable mobile phones and cloud services to collect and analyze systematic data for use in discovery*" [5]. In this scenario, data can be continuously collected by leveraging user mobility and phone sensors across a range of applications including mobile sensor networks [8], transportation and traffic monitoring [3], environmental sensors [14] and street safety [13].

Spatial crowdsourcing (SC) [9] provides a framework for the previously mentioned data collection applications where data requesters can create tasks in geographic areas of interest and workers are assigned or voluntarily choose to complete these tasks based on their spatial location. To fulfill an optimization function, such as minimizing distance traveled by workers [18], ensuring data quality [2], or maximizing task assignment [9], the task requester must accurately geolocate the location of task execution by providing geographic coordinates in the request to the SC server. But what happens when the requester is interested in sensing a geographic region, instead of a specific location, because the location of one or more events of interest is not precisely known? One solution would be to use the SC framework by modifying the request sent to the SC server to include a geographic region instead of the precise location. The SC server would then query all workers in this geographic area about the phenomenon of interest. While this solution is viable, it is impractical when it comes to a either a large-scale geographic region e.g., a city, or when the geographic region contains too many individuals to be reasonably queried. A budget constraint is a vital factor to consider in order to (i) save energy for resource constrained systems, e.g., disaster [17] and safety applications, because in an emergency, communication networks tend to fail and resources, such as bandwidth, are scarce [11]; and (ii) prevent users from becoming overwhelmed by queries and reaching a point where they cease using the crowdsensing system.

In our work, we address what we term the "spatially-blind participatory crowd sensing" problem. In this problem, the SC task requester is not able to specify a precise location for a task but instead only a larger geographic region due to a lack of geographically tied distribution information about the phenomenon. In particular, our goal is to answer the following research question: *given the real-time interest of an SC requester in a specific geographic region, and a specific phenomenon of an unknown spatial distribution, who are the workers the SC server should query given a budget constraint of selecting N out of M crowd workers, where N < M, to maximize the probability of coverage for the phenomenon?*

To answer this question, this paper contributes the following:
(i) We define the problem of spatially-blind crowdsensing under budget constraints. To the best of our knowledge, we are the first to study this problem.
(ii) We define two types of queries under the setting of spatially-blind crowdsensing: binary and exploratory queries.
(iii) We propose two novel algorithms, one that does and one that does not rely on worker feedback (DispNN and DispMax, respectively), to select N out of M workers based on their locations, where $N < M$. We compare our algorithms to random selection and a greedy heuristic [6]. We study the performance of our proposed algorithms under two event distributions: clustered and complete spatial randomness. Our algorithms outperform random user selection by up to 30% and the greedy heuristic by up to 5x more detected incidents. We then test the algorithms on a real dataset of street harassment reports in three different cities and show the applicability of DispNN and DispMax in detecting incidents and locating workers close to these incidents without any prior knowledge of the incident distribution. Although we discuss the spatially-blind participatory crowdsending under budget constraints problem under the umbrella

First ACM Workshop on Mobile Crowdsensing Systems and Applications, November 6–8, 2017, Delft, Netherlands

ACM ISBN 978-1-4503-5555-1/17/11...$15.00
https://doi.org/10.1145/3139243.3139249

of crowdsensing, our work could be extended to other communities of artificial sensors, mobile phones or even robotic sensors.

2 RELATED WORK AND MOTIVATION

Since the introduction of “crowdsourcing" as a modern business term [8], a significant body of work has been dedicated to the study and implementation of crowdsourcing in real life applications. Spatial crowdsourcing (SC), where the information sought is bound to a particular geographic area, has received significant attention [4, 9, 22]. A number of fundamental challenges persist for the design and implementation of SC platforms. Zhao and Han provide a taxonomy of SC, with categories associated with the worker model, task model, response model, and optimization goal of a SC problem [23]. SC problems are split into two categories: problems where servers assign tasks to workers (SAT) and problems where workers select tasks (WST). Each of these two types of problems can be split further based on the worker model used for the problem; reward-based problems and self-incentivized problems. DispNN and DispMax provide a task assignment solution for reward-based SAT problems that seek to generate information about some environment (e.g., neighborhood, city, park, concert) with high coverage of the environmental area.

Beyond our contributions to the general area of SC using reward-based SAT, there is a specific SC problem that we seek to address: event-detection. Kazemi and Shahabi formally propose the *maximum task assignment* (MTA) problem as well as several solutions [9]. While solutions to the MTA problem seek to optimize task assignment given a number of spatially known tasks and workers at a specific time interval, they still require *a priori* information about the location of events and do not incorporate a notion of resource budgeting. Most similar to our work are [16, 20, 21]. In [20, 21], the goal is to maximize the system utility through a focus on task allocation under sensing capability constraints. In contrast, our goal is to maximize spatial situational awareness. In [16], To *et al.* introduce adaptive budget algorithms used to perform real-time task assignment in hyperlocal SC under budget constraints. However, the algorithms introduced require *real-time* information about the location of events of interest. In contrast, we seek to enable detection of events for which hyperlocal spatial information is not previously known. Our solution is particularly important for gathering information about small-scale, ephemeral social events.

As cities become smarter and cyberphysical systems become increasingly pervasive, there is an increasing need for SC platforms that are designed to flexibly collect quality data using methodologies that adapt to the dynamic intersections of human behavior and complex systems. One of the most critical aspects to designing city-scale SC platforms is resource scalability. To leverage the crowd for location-based data collection at a large scale, spatial crowd-sourcing platforms must be able to minimize resource consumption to harvest high quality data. For a SC task, resources may include network bandwidth, energy, user attention, time, and money. In particular, our work focuses on information queries that are best answered via human interpretations of the environment (e.g., *“Are you feeling too cold, too hot, or comfortable right now?”* vs. *‘‘What is the temperature outside today?”* or *“On a scale from 1-10 how safe do you feel right now?”* vs. *“Is your bus stop well lit?”*).

3 PRELIMINARIES

In this section, we introduce relevant definitions and offer examples of motivating queries.

An **incident** is a real-time event or phenomenon that occurs at a particular location. An incident is tied with the specific geographic region around it; any worker in this region is able to sense or detect the incident. We model this region as a circular geographic space centered around the incident location with a specific radius. The larger the radius of the incident, the higher the probability that workers will be able to detect it. For example, the effect of a hurricane can be sensed over an entire city; however a street harassment incident can only be sensed if the worker is within a few meters. In the problem of “spatially-blind participatory sensing,” the location of incidents is not known to the SC server or the requester. It is therefore vital to design a smart algorithm that tries to capture as many incidents as possible in the spatial area of interest. More formally, an incident i of form $< id, l, r >$ is an incident at location l and can be detected by all workers within a circular space centered at l with radius r.

A **worker** is a person or device, i.e. a sensor or node, who can sense an incident in their vicinity. Formally, a worker w, of form $< id, l >$, is a mobile device carrier, or the device itself, who is a subscriber of the crowdsensing application and can report an incident of interest, in their geographic vicinity, to the SC server in real-time.

A **real-time information query** is a query sent by the SC server to workers in a spatial region to inquire about one phenomenon of interest in real-time. We envision two types of queries. First, a *binary query*, which requires a yes/no response. As an example, a binary query could be *“Is your location affected by the hurricane?”* or *“Do you feel safe in your location?”*. This query is beneficial to obtain a high-level understanding of the spatial occurrence of the phenomenon of interest. A second type of query, an *exploratory query*, seeks to understand incidents at a more fine-grained level. The objective of this query is to eventually draw an approximate heat map of the phenomenon for the spatial region. Examples of this query include *‘‘On a scale from 1-10, how safe do you feel right now?”* and *“Is your location highly-walkable, somewhat walkable, or car-dependent?”*

Finally, a **spatially blind worker selection algorithm under budget constraints** is an algorithm that runs on the SC server that aims to select workers under a specific budget of N out of M total workers without any prior knowledge of the incident spatial distribution. Since the algorithm is spatially blind to the incident spatial distribution, we cannot model the worker selection as a Maximum Task Coverage problem which is known to be strongly NP-hard [6]. Instead we have to devise a method of worker selection to maximize the spatially unpredicted incident coverage.

4 PROBLEM STATEMENT AND MEASURES

Spatially-blind participatory crowdsensing under budget constraints. In our system, we have a two-dimensional geographic region and a number of online workers (M) that can sense the environment around them. We investigate how to distribute queries within predefined geographic regions in the case of limited resources. To meet this constraint, we bound the system by a specific number of probes per time slot. Hence, the question becomes: *Given M workers and N resources, where $N < M$, which N workers should be queried to sufficiently answer a spatially-constrained query?* In other words, how should the SC server select these N workers?

If we tackle this question from a probabilistic point of view, then the straightforward answer is to try to select workers with the same spatial distribution as the phenomenon in the geographic region. For instance, if we know that a certain phenomenon occurs uniformly in the region, then we would have no bias in selecting the workers to query, i.e. each worker should have the same probability of selection. On the other hand, if we know the phenomenon is more prevalent in certain areas of the region, we should incorporate information when selecting the workers such that more workers are queried in the area of interest, where the phenomenon is likely to occur, and fewer workers in areas where there is a smaller probability of occurrence. The question becomes far more challenging if the distribution is not known or if it is not stationary. In this case, we ask if there is a systematic algorithm that can be used for selecting workers to spatially identify a phenomenon regardless of the probabilistic distribution or time variation.

Measures. To quantify the performance of the different approaches to solve the spatially-blind participatory crowdsensing under budget constraints problem, we propose the following three metrics for the output of the worker selection algorithm, which is the set of N workers that are queried (Queried Workers), denoted by QW. Let $QW = \{qw_1, qw_2, ..., qw_N\}$

- Coverage (COV): the number of incidents covered out of the total number of incidents that occur in the $2D$ geographic region. We define an incident as covered if the algorithm selects at least one worker in the range of the incident to be queried. Let the set of incidents that occur in the geographic region be $\{i_1, ..., i_{\mathcal{I}}\}$ and $Range(i_k)$ denote the set of workers in range of incident i_K, where a worker (w_j) is defined to be in the range of an incident if $dist(w(l)_j, i(l)) \leq i(r)$. Coverage is formally measured as:

$$COV = \sum_{k=1}^{\mathcal{I}} Coverage_k \tag{1}$$

where,

$$Coverage_k = \begin{cases} 1, & \text{if } (Range(i_k) \cap QW) \neq \phi \\ 0, & \text{otherwise} \end{cases}$$

- Close worker count (CWC): the absolute number of workers in the range of each incident for all incidents:

$$CWC = \sum_{k=1}^{\mathcal{I}} |(Range(i_k) \cap QW)| \tag{2}$$

- Redundancy (RED): the average share of workers per covered incident, defined as:

$$RED = CWC/COV \tag{3}$$

5 ALGORITHMS AND METHODOLOGY

We assume that there are M online workers in a two-dimensional geographic area. The server that selects workers to query is bounded by N resources, where N and the geographic region are pre-determined by the SC requester. Each of the M workers has a specific location in the spatial area, determined by a two-dimensional system, e.g. (x, y) or a $(latitude, longitude)$. We assume that the selected workers will respond to the query. If needed, a pre-selection phase can be used to eliminate workers that are not likely to co-operate, such as requiring the installation of an app to facilitate querying. The focus of the worker selection mechanism is how to select N out of M nodes, where $N < M$, to maximize incident detection.

DispNN and DispMax algorithms. Suppose a requester wishes to identify unsafe areas in a geographic area (G) using only N worker probes. The requester provides the server with the following information: $< G,\ Q,\ N,\ ANS >$, where Q is the query related to the phenomenon of interest and ANS is the answer to the query for which the server will probe further, e.g., $ANS = No$ for $Q =$ "$Is\ it\ safe\ around\ you?''$. Since the SC server is spatially-blind with respect to the incident distribution, we can envision a solution that tries maximize the spatial variation of N worker locations so that the geographic area is covered. One measure of the degree to which points in a point set are separated from each other is spatial dispersion [1] measured as $tr(cov(P))$, where tr and cov denote the trace and covariance operations. Here, the point set is represented as a matrix P where each row represents a point p. Hence, the crowdsensing problem could be modeled as maximizing the spatial dispersion for the N workers i.e., selecting a set of N workers, $\mathcal{QW} = \{qw(j),\ j \in \{1, ..., N\}\}$, such that $\underset{\mathcal{QW}}{\text{argmax}}\ tr(cov(QW(l)))$ where $QW(l)$ represents the matrix of the locations of the queried workers as follows:

$$QW(l) = \begin{bmatrix} qw_1(l) \\ \vdots \\ qw_N(l) \end{bmatrix}$$

In order to ensure a globally optimal solution, we can compute the dispersion of all $\binom{M}{N}$ worker location combinations and choose the combination with the maximum spatial dispersion as the set of queried workers. Solving $\underset{\mathcal{QW}}{\text{argmax}}\ tr(cov(QW(l)))$ by generating all possible worker location combinations is of a complexity exponential in N. More generally for a fixed N, this yields a complexity of $O(M!/N!(M-N)!)$ which could become unrealistic for real-time applications as M and N increase. Instead, we propose to use Lloyd's K-means clustering algorithm [10], which tries to place the centers of the clusters as far away from each other as possible. We can then apply Lloyd's algorithm by computing the N-means clusters and choosing the workers with the closest locations to the centroid of each of the N clusters as a way of **maximizing the dispersion** of the N workers. Using Lloyd's algorithm yields a complexity of $O(MN)$, assuming constancy of point dimensions and number of iterations needed until convergence [10]. This method represents the core of DispMax and the first stage of DispNN.

Another concept that can be applied to this problem is Thompson sampling, which is a heuristic for choosing actions that address the **exploration-exploitation** dilemma in the multi-armed bandit problem [12]. In our problem, we can design an algorithm that combines the concepts of exploration and exploitation. We define exploration as the process of maximizing the dispersion of worker location so that we can explore the geographic region. On the other hand, the concept of exploitation relates to making use of worker feedback about the incidents in the selection of other workers. For instance, using exploitation, if a worker (w_5) indicates that it is not safe around them by answering "No" to the query "Is it safe around you?", the server could exploit that answer and dedicate a subset of the N probes to some workers close to w_5. Querying the neighboring nodes can provide the requester with information related to spatial correlations and can help the requester bound the region in which the phenomenon occurs.

Based on this prior work, our algorithm, DispNN, selects N of M workers in a geographic region by dividing the selection into two phases: (1) Disp: the dispersion maximization phase (Exploration) and (2) NN: the use of worker feedback to query the nearest neighbors (Exploitation). These two phases work under the total budget constraint N; a percentage (FSP) of N is dedicated to the Disp phase and the percentage $(1 - FSP)$ of N is used to query the nearest neighbors of workers of interest based on the initial query response. If there is not sufficient feedback to locate nearest neighbors, we use the remaining resources towards another round of exploration.

DispNN assumes workers are not malicious and thus it operates under the single task assignment paradigm [9]. A variation of DispNN would be to not rely on user feedback; in this case the algorithm will dedicate all N probes towards the first phase, Disp. We call this algorithm DispMax. DispMax is beneficial to use in two scenarios: when the SC server cannot assume full trust in all workers, and when the SC server receives an exploratory question for which the intent is to build a heat map of the distribution of the answers, e.g., categorizing areas of a city as "extermely-walkable, some-what walkable, or car-dependent".

6 EXPERIMENTS AND RESULTS

Experiment setup. Real world phenomenon rarely follow complete spatial randomness [15]. Hence, we study the performance of DispNN and DispMax under three different event distributions: clustered, random, and real-world datasets. There are multiple variables that can be controlled to test the behavior of DispNN and DispMax. Table 1 summarizes the most important experimental parameters. In all of our experiments, except the case study on real-world data, we use a 10x10 spatial grid and the Euclidean distance to measure the straight line distance between locations. Since it is unrealistic to assume that workers are uniformly distributed across the spatial area, we model the worker location distribution as a mixture of a Poisson point process [15] with $\lambda = \frac{crowd\ count}{2}$ and a cluster process where the other half of the crowd is distributed across a number of clusters that varies between [1, 10] and is chosen randomly. We compare our algorithms for worker selection to three alternative approaches and one optimal approach as follows:

- Random worker selection (Rand): we select N workers randomly based on a uniform distribution, i.e., each worker has the same probability of being selected.
- Greedy worker selection (Greedy): we apply the greedy heuristic proposed in [6] to solve the Maximum Task Coverage problem. At each iteration, the heuristic selects the worker that covers the maximum number of uncovered tasks; however, because the incident distribution is not known, we modify the heuristic. We choose the worker that *is likely* to cover part of the geographic space that is not covered. We start by selecting a worker randomly and then iterating through the rest of the workers and select the worker that will maximize the spatial dispersion. We continue iterating until we have N workers.
- Random with feedback worker selection (Randf): we use random worker selection in the first phase then apply the feedback process similar to the DispNN methodology.
- Optimal coverage (OptCov): we assume full knowledge of incident and worker locations and select N worker locations that maximize the number of incidents covered. We use this as a reference for the maximum coverage obtained if the server was aware of the incident distribution.

incident count: number of incidents distributed across the cells of the spatial matrix.
incident range: the radius of an incident where, if a worker is present within the radius, he/she will be able to detect the incident.
crowd count: the M workers from which N will be chosen to query, where $N < M$.
N: the number of workers the SC server is limited by to query.
first stage percentage (FSP): the percentage of workers of the N resources that will be selected to query in the Disp stage.

Table 1: Parameters used in experiments.

Generic observations. There are many variables that can affect the output of the experiments. For instance, we found that as the range of incidents increases, all approaches tend towards the same performance. The same observation is true as N approaches M. As a result, we stress the different approaches by modeling incidents with smaller ranges. An interesting trade-off for DispNN is related to the choice of FSP. As FSP increases, COV increases but CWC tends to decrease and vice versa, for clustered distributions. We find that a good choice for FSP, that strikes a balance between COV and CWC, is 0.8, i.e., 80% of the probes allocated to the Disp phase and 20% for the NN phase.

6.1 Clustered incident experiments

Geographer Waldo R. Tobler stated in the first law of Geography: "Everything is related to everything else, but near things are more related than distant things" [19]. In this set of experiments, we assume that the incidents are related to each other, i.e. they form clusters across the $2D$ spatial region as shown in Figure 1. Our goal in these experiments is to study the performance of the different query algorithms when the incidents are clustered.

We vary the number of clusters in our $2D$ spatial area from one to ten while fixing the incident count to be 50 with a range of 1 unit distance. We set crowd count to 60 and $N = 30$. To enforce data variability, we model the size of each cluster as a random variable while ensuring that the aggregated size of all the clusters is equal to crowd count. For each number of clusters, we average results over 100 different random configurations.

Figure 2 illustrates COV, CWC and RED aggregated over all random configurations of clustered incidents. DispNN and DispMax outperform Rand, Greedy and Randf. DispNN and DispMax achieve a median coverage of 60% while the median OptCov is 70%. Rand ($\mu = 47.8, \sigma = 18.4$) and Randf ($\mu = 47.7, \sigma = 17.9$) provide a median coverage of 48%, while Greedy ($\mu = 41.9, \sigma = 20$) results in a median of 40% coverage. With respect to coverage, DispNN outperforms Rand, Greedy, and Randf by an average of 22.5%, 39.8% and 22.7%, respectively, and it comes within 13.3% of OptCov. Similarly, DispMax outperforms Rand, Greedy, and Randf by an average of 23.8%, 41.3%, and 24.1%, respectively, and comes within 12.4% of OptCov. Randf and DispNN achieve a higher CWC than Rand and DispMax since they rely on worker feedback; their NN selection phase selects workers that uncover other incidents because of the clustered nature of the incidents. DispMax achieves the lowest median RED of 1.3, since it maximizes the location dispersion of workers without relying on any feedback.

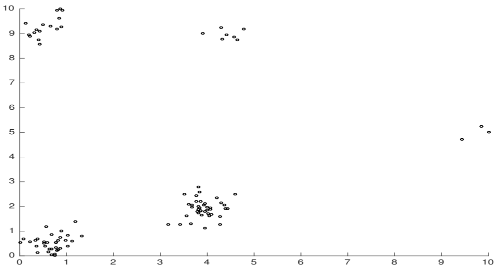


Figure 1: Example of a $2D$ spatial region with five clusters.

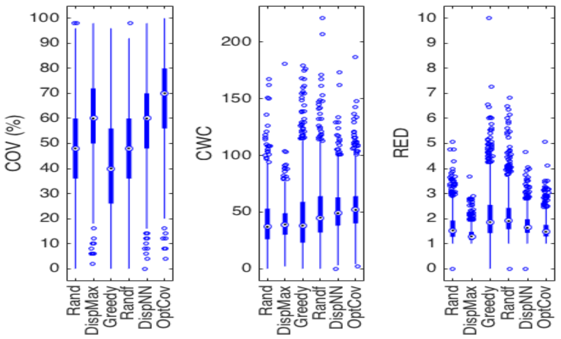


Figure 2: COV, CWC, and RED for distributions of clustered incidents.

6.2 Complete spatial randomness experiments

In the next set of experiments, the probability of occurrence of an incident is uniform across the spatial region. Incident occurrence in the spatial area follows a Poisson point process with $\mu = \lambda = incident\_count$. We randomly generate 100 different spatial region incident configurations. On average, the spatial matrix contains $incident\_count$ incidents. We operate under the same settings where $incident\_count = 50$ with a range of 1 unit distance and $M = 60$, and $N = 30$. Figure 3 shows that DispNN outperforms Rand, Greedy, and Randf in terms of coverage by 18.4%, 62%, and 26.2%, respectively, and comes within 11.7% of OptCov. Similarly, DispMax outperforms Rand, Greedy, and Randf by 26.9%, 73.6%, 35.2%, respectively, and comes within 5.4% of OptCov. We note that DispMax consistently performs closer to OptCov than DispNN. Because of the random distribution of incidents, there are no spatial correlations, unlike in the previous clustered distribution. Hence, there are fewer workers for DispNN to exploit in the NN phase. For the same reason, Randf performs slightly worse than Rand in terms of coverage. Apart from OptCov, DispNN and DispMax achieve the lowest RED since they focus on maximizing the dispersion. The result is higher incident coverage, on average, with workers more geographically dispersed.

6.3 Case study: Hollaback street harassment data

After applying DispNN and DispMax to the previous two distributions, we wish to examine the algorithms under real incident distributions. To do so, we test our algorithm on a global street harassment dataset provided by Hollaback [7].

Data overview. Hollaback is a non-profit movement powered by local activists in 92 cities and 32 countries to end street harassment.

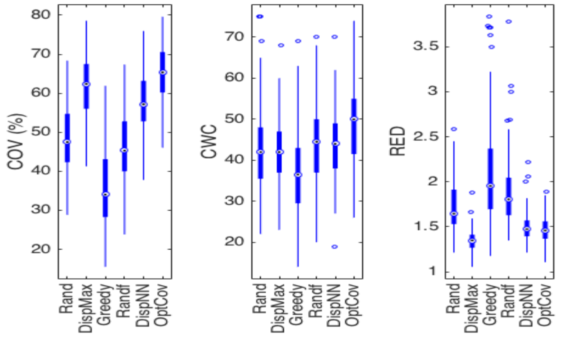
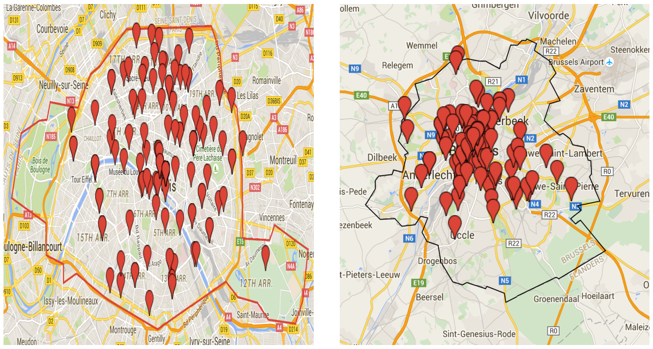


Figure 3: COV, CWC, and RED for incidents that follow complete spatial randomness.



(a) Paris

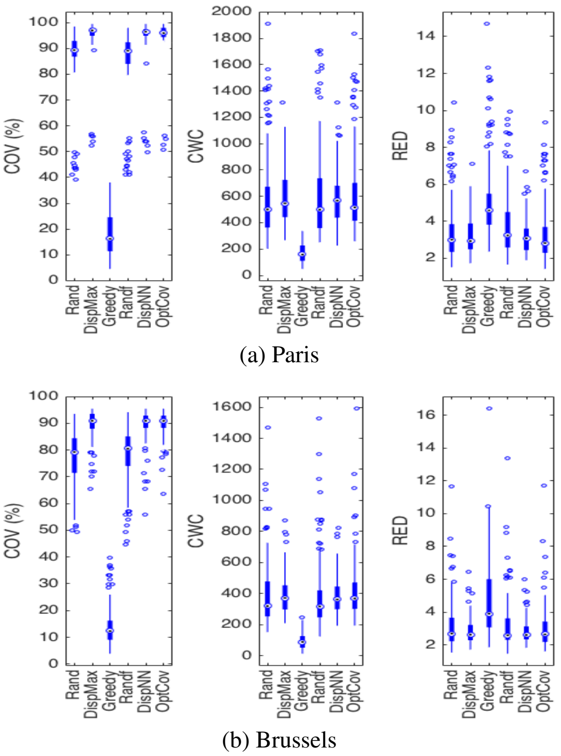
(b) Brussels

Figure 4: Distribution of harassment incidents across representative city datasets.

Through the Hollaback phone app and the online platform, users worldwide can report stories of street harassment to share with the Hollaback community. In some communities, local governments are informed in real-time about street harassment. The Hollaback app uses GPS to record a data set of street harassment event locations as a means of improving the collective understanding of street harassment. As of January 2016, over 8000 street harassment incidents have been recorded in the dataset since February 2011. It is on this data set that we test DispNN and DispMax.

Analysis. From the Hollaback dataset, we select two cities (Paris, and Brussels) for which we have enough harassment samples for statistical significance (i.e. more than 30 samples). We test the performance of the six querying approaches on these cities. As a first step, we parse the Hollaback dataset such that incident reports are grouped by city. To do so, we use bounding box coordinates and shape files for each city to determine incidents bounded by the city borders and we remove any outliers. Figure 4 shows the resulting distribution of events for the two cities. The Paris dataset contains 197 harassment incidents and covers an area of 28.2 mi^2, while the Brussels dataset contains 154 incidents covering a geographic area of 28.4 mi^2.

For each of the cities, we generate 100 different variations of crowd locations ($M = 1000$) and set $N = 500$. In this analysis, $incident\_count$ is taken directly from the Hollaback dataset. We update the distance metric and use the Haversine formula to calculate



(a) Paris

(b) Brussels

Figure 5: COV, CWC, and RED for Paris, and Brussels.

the great-circle distance between two points as follows:

$$d = 2R * atan2(\sqrt{a}, \sqrt{1-a}) \qquad (4)$$

where a is calculated as $\sin^2((\Delta\phi)/2) + \cos(\phi_1)\cos(\phi_2) * \sin^2((\Delta\lambda)/2)$; $\Delta\phi$ and $\Delta\lambda$ are calculated as the radian difference between the latitudes and longitudes, respectively; and R is the Earth's radius. Since a harassment incident cannot be witnessed unless a worker is very close, we adjust the incident range to 5 meters. We measure COV, CWC and RED aggregated over all random configurations of worker distributions for all six querying approaches and plot the results in Figure 5. DispNN and DispMax achieve close to optimal coverage in the case of Paris and Brussels. The median coverage using DispNN and DispMax for Paris and Brussels was found to be 96.4 and 90.1, respectively. We note that Greedy performs poorly for all cities. The reason is that at each step, Greedy chooses the point that maximizes the dispersion. The result is it selects the majority of the workers around the borders of the geographic region where the number of harassment incidents are minimal.

7 CONCLUSION

This paper proposes DispNN and DispMax, spatial querying algorithms that select workers to discover randomly placed events within a $2D$ spatial environment through intelligent probing of worker resources. While the experimental evaluation confirms the applicability of proposed approaches, the algorithms could be adjusted to accommodate prior information about the nature of the events. If an approximate spatial distribution is known, we can use weights to reflect the probability of occurrence in each spatial sub-region and then apply DispNN and DispMax on each of the sub-regions. On the other hand, knowledge of spatial correlations and event stationarity could be used to manipulate worker selection. Our work is applicable in numerous scenarios, particularly when resource preservation is important and when querying all nodes will cause too large a disturbance or a response implosion.

8 ACKNOWLEDGMENTS

This work was funded in part by the NSF Graduate Research Fellowship Program under Grant No. DGE-1144085, and in part by US Army Research laboratory and the UK Ministry of Defence under Agreement Number W911NF-15-R-0003. The authors would like to thank Hollaback for sharing their dataset.

REFERENCES

[1] L. Anselin and A. Getis. 1992. Spatial Atatistical Analysis and Geographic Information Systems. *The Annals of Regional Science* 26, 1 (1992), 19–33.

[2] P. Cheng, X. Lian, Z. Chen, R. Fu, L. Chen, J. Han, and J. Zhao. 2015. Reliable Diversity-based Spatial Crowdsourcing by Moving Workers. *Proceedings of the VLDB Endowment* 8, 10 (2015), 1022–1033.

[3] U. Demiryurek, F. Banaei-Kashani, and C. Shahabi. 2010. TransDec: A spatiotemporal query processing framework for transportation systems. In *IEEE 26th International Conference on Data Engineering (ICDE)*. 1197–1200.

[4] D. Deng, C. Shahabi, and U. Demiryurek. 2013. Maximizing the Number of Worker's Self-selected Tasks in Spatial Crowdsourcing. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. Orlando, FL, USA, 324–333.

[5] D. Estrin, K. M. Chandy, R. M. Young, L. Smarr, A. Odlyzko, D. Clark, V. Reding, T. Ishida, et al. 2010. Participatory Sensing: Applications and Architecture [Internet Predictions]. *IEEE Internet Computing* 14, 1 (2010), 12–42.

[6] D. S. Hochbaum. 1996. Approximating Covering and Packing Problems: Set Cover, Vertex Cover, Independent Set, and Related Problems. In *Approximation Algorithms for NP-hard Problems*. PWS Publishing Co., 94–143.

[7] Hollaback. 2015. Read and Share Stories. When it comes to street harassment, you are not alone. http://www.ihollaback.org/share/. (2015).

[8] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden. 2006. CarTel: A Distributed Mobile Sensor Computing System. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*. 125–138.

[9] L. Kazemi and C. Shahabi. 2012. GeoCrowd: Enabling Query Answering with Spatial Crowdsourcing. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. Redondo Beach, CA, 189–198.

[10] S. Lloyd. 1982. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory* 28, 2 (1982), 129–137.

[11] B. S. Manoj and A. H. Baker. 2007. Communication Challenges in Emergency Response. *Commun. ACM* 50, 3 (2007), 51–53.

[12] H. Robbins. 1985. Some Aspects of the Sequential Design of Experiments. In *Herbert Robbins Selected Papers*. 169–177.

[13] Safetipin. 2014. Safetipin – Supporting Safer Cities. http://safetipin.com/. (2014).

[14] Scientific American. 2013. 8 Apps That Turn Citizens into Scientists. https://www.scientificamerican.com/article/8-apps-that-turn-citizens-into-scientists/. (2013).

[15] M. Sherman. 2011. *Spatial Statistics and Spatio-temporal Data: Covariance Functions and Directional Properties*. John Wiley & Sons.

[16] H. To, L. Fan, L. Tran, and C. Shahabi. 2016. Real-time Task Assignment in Hyperlocal Spatial Crowdsourcing Under Budget Constraints. In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*. Kona, Big Island, HI, USA.

[17] H. To, S. H. Kim, and C. Shahabi. 2015. Effectively Crowdsourcing the Acquisition and Analysis of Visual Data for Disaster Response. In *IEEE International Conference on Big Data*. 697–706.

[18] H. To, C. Shahabi, and L. Kazemi. 2015. A Server-assigned Spatial Crowdsourcing Framework. *ACM Transactions on Spatial Algorithms and Systems* 1, 2 (2015), 29–56.

[19] W. R. Tobler. 1970. A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography* 46, 1 (1970), 234–240.

[20] J. Wang, Y. Wang, D. Zhang, F. Wang, Y. He, and L. Ma. 2017. PSAllocator: multi-task allocation for participatory sensing with sensing capability constraints. In *Proceedings of the ACM CSCW*. 1139–1151.

[21] J. Wang, Y. Wang, D. Zhang, L. Wang, H. Xiong, A. Helal, Y. He, and F. Wang. 2016. Fine-Grained Multitask Allocation for Participatory Sensing With a Shared Budget. *IEEE Internet of Things Journal* 3, 6 (2016), 1395–1405.

[22] H. Yu, C. Miao, Z. Shen, and C. Leung. 2015. Quality and Budget Aware Task Allocation for Spatial Crowdsourcing. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*.

[23] Y. Zhao and Q. Han. 2016. Spatial Crowdsourcing: Current State and Future Directions. *IEEE Communications Magazine* 54, 7 (July 2016), 102–107.