Who to query?
A two stage querying algorithm for tracking variant/unknown event distributions

Mai ElSherief
Dept. of Computer Science
UC Santa Barbara
mayelsherif@cs.ucsb.edu

Ramya Raghavendra
IBM T. J. Watson Research Center
rraghav@us.ibm.com

Elizabeth Belding
Dept. of Computer Science
UC Santa Barbara
ebelding@cs.ucsb.edu

ABSTRACT
In this paper, we propose a two-stage node selection algorithm for resource constrained systems based on the nodes location to track incidents in a $2D$ environment. Our algorithm maximizes the dispersion of the locations with a portion of the available resources. Based on the nodes' sensing feedback, it selects the K nearest neighbors for the nodes that provide a positive feedback using the rest of the available resources. We test the two-stage algorithm on three different distributions: uniform, clustered and long-tailed. We then apply the algorithm to a real harassment dataset provided by Hollaback. The proposed algorithm outperforms a random selection policy by up to 63% and a policy that relies on dispersion maximization without incorporating feedback from the nodes by up to 68%.

1. INTRODUCTION
Sensors have become an integral part of daily life. The common smartphone includes a variety of different sensors, such as camera, microphone, GPS, accelerometer, digital compass, light sensor, and Bluetooth as proximity sensor. The ubiquity of sensors is also prevalent in the urban environment. Examples include traffic sensors, agriculture sensors, wireless parking sensors, infrastructure sensors, weather, and pollution sensors. Data analysis of such sensors can hold important observations. For instance, [9] leverages the geographic and temporal data associated with taxis in NYC to gain insight into many different aspects of city life, from economic activity and human behavior to mobility patterns. When combined with crowdsourcing of humans senses, critical data can be generated about surroundings. One example is the application "Waze", where users can report traffic jams, accidents and other road related incidents in real-time. The work in [3] uses local workers to collect data at events and remote workers to curate the collected information and generate event reports.

Despite the ubiquity of human and device sensors, the challenges of energy preservation and resource constraints are always present. The truth is that all systems are bound by a fixed amount of resources. For instance, [18] and [19] focus on eliminating redundancies among correlated sensor measurements.

In this paper, we investigate the problem of how we can probe a limited subset of sensors in a particular environment to either preserve energy or other resources. In particular, we envision a world where users/sensors can be probed to collectively answer some question. An unanswered question can be related to a phenomenon that needs to be tracked under the constraints of N resources. The experiments, analyzing the spatial and temporal characteristics of the twitter feed activity responding to a 5.8 magnitude earthquake which occurred on the East Coast of the United States (US) on August 23, 2011 in [7], support the notion that people act as sensors to give us comparable results in a timely manner. Examples of resource constrained systems include disaster and safety applications or in general terms to track any spatial phenomenon. In an emergency, communication networks tend to fail and available resources, such as bandwidth, are scarce [17]. Under these constrained settings, the two-stage algorithm can be used to probe the crowd/sensors about the current situation of the disaster in their current location. Another example of safety applications is what happens in Tahrir Square during Egyptian revolutions in 2011 and 2012. At that time, women were discouraged from participation due to the high harassment rates [1]. This resulted in movements of men forming protective human shields [2] around female protestors to avoid assault. Our proposed algorithm can be used to query users for safe zones for women and then use the results for safe routing around the square or for identifying zones where women can safely participate in the protests.

Our contribution in this paper is three-fold. First, we propose a two-stage matching algorithm that probes/queries N nodes out of M available nodes to track a real-time phenomenon with no prior information about the event distribution. The algorithm outperforms the random user selection by up to 63% in terms of choosing nodes chosen that are closer to the events and outperforms the dispersion maximization algorithm by up to 68%. Secondly, we study the performance of our proposed algorithm under three different distributions: uniform, clustered and long-tailed. We then test the algorithm on a real dataset that is comprised of harassment cities in three different cities. Third, we discuss how the algorithm can be altered based on trust variations and prior knowledge availability.

The rest of this paper is organized as follows. Section 2 surveys the related work while Section 3 describes the proposed two-stage algorithm. Section 4 experimentally evaluates the proposed algorithm, and Section 5 discusses tradeoffs and variations of the two-stage algorithm. Section 6 concludes the paper.

2. RELATED WORK

Since the introduction of "crowdsourcing" as a modern business term in 2006 [12], a significant body of work has been dedicated to the study and implementation of crowdsourcing in real life applications. In particular, spatial crowd sourcing, where crowd participation is bound by a particular geographic area, has received significant attention [13, 8, 23]. For instance, [16] introduces a location-based real-time social question answering service, where users can ask temporal and geo-sensitive questions and then receive answers that are crowdsourced in a timely fashion. A crowdsensing platform was introduced in [6] to facilitate the collaboration of large groups of people participating in collective actions of urban crowdsourcing.

Using people as sensors, collective sensing and citizen science have opened doors for interesting research problems. Some of these challenges are introduced in [5]. One important challenge in geo-crowd sensing is detecting unusual events. The work proposed in [15] leverages microblogging websites such as Twitter to detect unusual geo-social events by identifying unusually crowded regions. Another challenge is the refinement of crowd sensed data and detection of fake data. Solutions based on a user's history and reputation have been introduced in the literature. The work in [24] proposes a reputation-aware model that balances the workload between users. Another challenge is fusing untrustworthy estimates [22]. Taking into account spatial properties, [21] tackles the problem of merging multiple spatial observations reported by possibly untrustworthy users using a heteroskedastic Gaussian process model.

Another related body of work is sensor networks [4] that include spatially and ubiquitously distributed autonomous sensors used to monitor physical and environmental conditions. Since the sensors are typically small, low-powered nodes, resource-constrained protocols emerged to preserve the energy of these devices. Examples of work targeting energy preservation include, but are not limited to [20], where the authors achieve geographic localization using noise tolerant acoustic ranging mechanism to meet severe resource constraints. In [14], data aggregation methods were introduced and achieved significant performance gains in comparison to end to end routing. The work proposed in [10] implements a system that analyzes sensor behaviors and uncovers misbehavior corresponding to inefficient device usage that leads to energy waste. In contrast, our work focuses on the how to choose sensors to query based on their location while constraining the number of probes to a portion of the total number of sensors hence, preserving energy.

3. RESEARCH QUESTION AND PROPOSED ALGORITHM

In our system, we have a two-dimensional grid and a number of objects that can sense the environment around them. These objects can be humans, artificial sensors, mobile phones or even robotic sensors. If we are interested in answering the question "What is the answer to Question X in this grid?", we can basically ask or query all the objects in the two-dimensional space and aggregate their findings. In this paper, we assume that to answer this question, you can only query N objects. Hence, the question becomes: *Given N resources, who should you select to track a phenomenon?* Answering this question becomes essential in the case of limited resources. This is particularly important in emergency scenarios when a network's performance degrades and preserving energy and other resources become critical.

If we attempt to tackle this question from a probabilistic point of view, then the straightforward answer would be to try to select objects/users with the same probabilistic distribution as the phenomenon. For instance, if we know that a certain phenomenon occurs at different places in the two-dimensional grid uniformly, then we would have no bias in selecting the users to query, i.e. each user/object would have the same probability of selection to be queried. On the other hand, if we know the phenomenon we are interested in is more prevalent in certain areas of the grid as opposed to other areas, we would take that into consideration when we are selecting the users to select more users to query in this area and fewer users 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 time variant? The aforementioned question becomes more interesting in this case and we can then inquire if there is a systematic algorithm that can be used for querying/selecting users to track a phenomenon regardless of the probabilistic distribution or time variation.

3.1 Technique Description

We assume that there are M users in a two-dimensional grid and that the system that selects a user to query is bounded by N resources, where $N < M$. Each of the M users has a specific location in the grid, determined by a two-dimensional system, e.g. (x, y) or a (lat, long). We also assume that the users selected will participate in answering the question of interest to the system and fully co-operate. A pre-selection phase can be used to eliminate users that are not likely to co-operate such as requiring the installation of an app to facilitate querying. Here, we focus on how to select N out of M users to where $N < M$ to keep track of events occurring in the two-dimensional grid.

Our technique combines K nearest neighbor (KNN) queries with querying users to maximize the dispersion of their location in the grid as depicted in Algorithm 1. We divide the selection of users into two stages. In the first stage, our goal is to select users with the goal of maximizing the dispersion of users' locations. We maximize the dispersion by selecting the set of points that maximizes the average distance between each point and its nearest neighbor as follows:

$$\operatorname{argmax} \sum_{i=1}^{N} \|p(i) - NN(p(i))\|^2 \qquad (1)$$

where p represents a point in the $2D$ grid and NN represents the nearest neighbor and the distance is measured as the Euclidean distance. The algorithm attempts to maximize the dispersion up to the number of maximization trials. Based on the crowd feedback in the first stage, we then proceed to a more fine-grained selection. The users that provide a positive feedback (i.e. they witness an event/ emergency in their location) are called the *pivot users*. In the second stage, we want the K nearest neighbors for the pivot users.

This initial algorithm assumes that the first stage users will respond with unfalsified responses. To relax this assumption, we explore dividing the selection of the second phase users into two groups: a group that consists of the KNN of the trusted pivot users, and another group that aims to maximize the dispersion. In this section, we will focus on studying our two-stage querying technique with the assumption of having full trust in the crowd and discuss other variants of the technique in subsequent sections.

Algorithm 1 Two-stage querying algorithm

```
function selectUsersFromGrid (FSP, N)
    selectedUsers = {}
    firstStageCnt = ⌊(FSP * N)⌋
    secondStageCnt = M − firstStageCnt
    firstStageUsers = maximizeDisp(firstStageCnt)
    usersFeedback = feedback(firstStageUsers)
    if usersFeedback.size == 0 then
        selectedUsers = maximizeDisp(secondStageCnt)
    else
        selectedUsers.append(firstStageUsers)
    firstStageQuota = calculateQuota(firstStageUsers)
    for user_i in firstStageUsers do
        selectedUsers.append(KNN(user_i,
    firstStageQuota_i))
    return selectedUsers
```

4. EXPERIMENTS

To quantify the performance of our technique, we perform multiple experiments with three types of data spread: clustered, uniform and real datasets. In our experiments, we compare our algorithm in the selection of users to two policies as follows:

- Random user selection: For this policy, we select N users randomly based on a uniform distribution.
- Dispersion maximization selection: The selection of users in this policy depends on selecting N users from the crowd who maximize the dispersion of their locations.

4.1 Experiment Variables

There are multiple variables that can be controlled to test the behavior of the two-stage querying technique. Table 1 summarizes the most important. The environment settings

Environment settings: • matrix dimension: the length and the width of the $2D$ spatial matrix. We model the spatial area under investigation as a $2D$ square matrix. • incident count: number of incidents distributed across the cells of the spatial matrix • resources or crowd count: the M resources from which N, where $N < M$, will be chosen to query
Query settings: • N: the number of resources the system is limited by to query/sense • first stage percentage (FSP): the percentage of users/sensors of the N resources that will be selected to query in the first phase. In our analysis, we test the cases of selecting $20\%, 40\%, 60\%$ and 80% of the N resources in the first stage. • k setting: used to identify the KNN crowd individuals/sensors to an incident
Approximation settings: • maximization trials: number of attempts to maximize the dispersion of selected individuals/sensors from the crowd

Table 1: Different parameters of the two-stage querying technique.

are related to the size of the $2D$ matrix, the number of incidents, the distribution of incidents across the matrix, and the number of resources from which to choose. In all of our experiments, except the case study, we set up the $2D$ matrix as a 10 by 10 matrix. We show results for incident count of 50 and number of resources (M) of 100. We varied the environment settings in our experiments and no noticeable differences were observed in performance. Instead, we focus on varying the query settings to better understand the two-stage technique. In this section, we focus on varying the first stage percentage and leave the variation of the k setting to the following section. We also show results for $t\_setting = 30$ which constitutes 30% of the available resources (M). We notice that the gap between the performance of our technique and the other techniques increases when $t\_setting$ decreases and all the techniques converge in performance when $t\_setting$ approaches M.

We compare the performance of our technique to two alternative selection approaches: Random selection and Dispersion Maximization. To do so, we utilize two different metrics: count of nodes queried in the KNN of incidents, and the number of incidents covered by the nodes queried. The two metrics are formally defined as follows.

- Close node count: This is measured as the absolute number of people/resources in the KNN of each incident for all incidents. This is formally represented as follows:

  $$Close\ node\ count = \sum \forall_{incident\ i}\ |(KNN_i \cap QU)| \qquad (2)$$

  where QU (the "Queried Users" set) is the set of users selected for querying.
- Coverage: measured as the number of incidents covered out of the total number of incidents occurring in

the $2D$ matrix. We define an incident as covered if at least one of the nodes in the incident's KNN was queried. This is formally measured as:

$$Coverage = \sum \forall_{incident\ i}\ Coverage_i\ where, \quad (3)$$

$$Coverage_i = \begin{cases} 1, & \text{if } (KNN_i \cap QU) \neq \phi \\ 0, & \text{otherwise} \end{cases}$$

4.2 Clustered data experiments

In this section, we aim to test our technique in a scenario where the events take a clustered form. Geographer Waldo R. Tobler's stated in the first law of geography: "Everything is related to everything else, but near things are more related than distant things." In this subsection, we assume that the incidents are related to each other in a clustered way i.e. they form clusters across the $2D$ spatial matrix as seen in Fig 1.

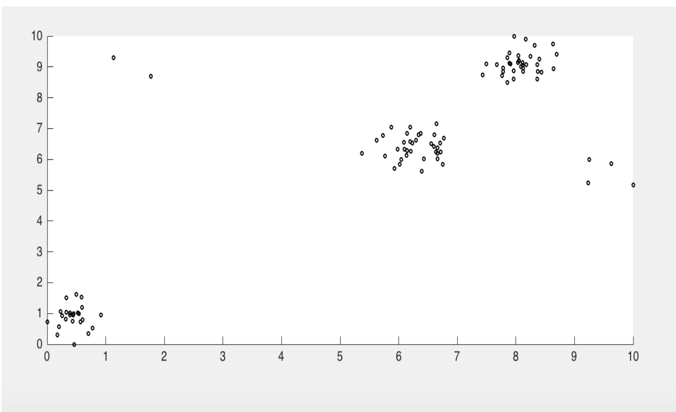


Figure 1: An example of a $2D$ spatial matrix with 5 clusters.

For this experiment type, we vary the number of clusters in our $2D$ matrix from one to ten clusters while fixing the resources or crowd count to be 100. To ensure 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 the crowd count. For each case of number of clusters, we average over 100 different configurations. Our objective is to measure the effect of variation of the first stage percentage on our performance metrics.

Figure 2 depicts the results for Close node count when varying the first stage percentage from 20% to 80%. We notice that our two-stage querying technique always outperforms the Random node selection and the selection based on maximizing the dispersion only. Table 2 depicts the amount of surge in Close node count in comparison to the Random and Dispersion maximization techniques. As the amount of resources queried in the first stage decreases, the close node count increases. This is due to the fact that when the first stage percentage decreases, the second stage resources increase under limited resources constraints, which focuses on resources close to incidents detected in the first stage. On the other hand, incident coverage tends to increase as the first stage count increases. This is depicted in Figure 3. We also notice that both Close node count and Incident coverage tend to increase with the number of clusters until the number of clusters is around four or five. Then, it decreases.

First stage percentage	20%	40%	60%	80%
Surge over Random	62.5%	58.89%	35%	20%
Surge over Dispersion Maximization	67.8%	62.32%	39.8%	20.64%

Table 2: Surge of Two Stage technique in comparison to Random and Dispersion Maximization techniques.

4.3 Uniformly distributed data experiments

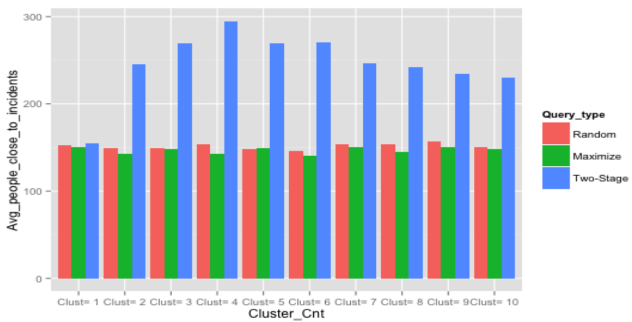
In this type of experiments, the probability of occurrence of incidents is uniform across the grid i.e. $P_I(j) = P_I(k)$ where $j \neq k$ and P_I denotes the probability of an incident occurring at a specific cell. We generate a 100 different matrices. We note that in the case that we know that the distribution of the incidents is uniform, the best we can do is to choose N nodes uniformly. Using the two-stage technique, we select N nodes without assuming any distribution about the incidents and check the performance in comparison to the uniform random selection which in this case is the best we can do. Figure 4 shows that the two-stage technique with $FSP = 20\%$ achieves the highest number of close nodes to the incidents while Figure 5 shows that the two-stage technique with $FSP = 80\%$ achieves higher coverage that the uniform random policy and it is close to the maximum coverage by an average of 1.32 incidents.

4.4 Long Tail Distribution

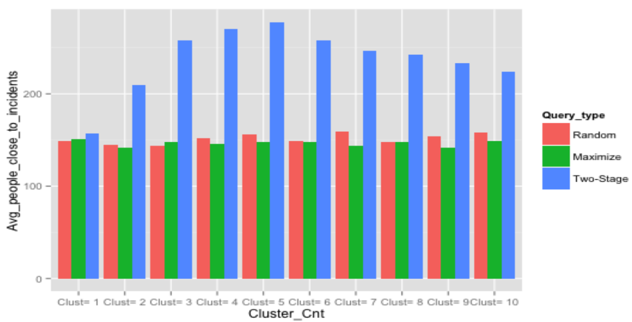
In this subsection, the incidents are generated according to a special case of the Long Tail distribution called the "Pareto principle". According to the Pareto principle, we assume that 20% of the matrix cells are home for 80% of the incidents and 80% of the matrix cells are home for 20% of the incidents. We generate a 100 different matrices applying the Pareto Principle randomly on the cells. We use a random uniform distribution to select 20% of the cells and generate 80% of the incidents uniformly for these cells and vice versa. Figure 6 shows that the two-stage outperforms both the random policy and the dispersion maximization policy by up to 10.2% and 12.1% respectively in terms of close node count. This is probably due to the clustering of events in only 20% of the grid which means that more than one incident is likely to occur in the same cell. So, if the two-stage technique reaches a node close to an incident in one cell, this same node will cover more than one incident in the same cell. Dispersion maximization achieves the best incident coverage as shown in Figure 7. The two- stage technique approaches the maximum incident coverage when $FSP = 80\%$ with a difference of 1.24 incidents on average.

4.5 Case Study: Hollaback harassment data set

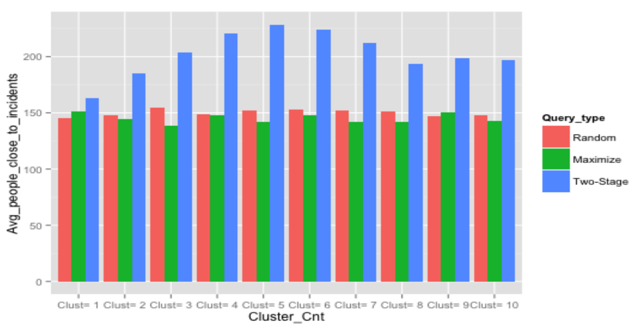
After applying the two-stage querying technique to the previously mentioned three distributions (clustered, uniform and long-tail), we wish to examine the technique under real incident distributions. To do so, we test our querying technique on a global street harassment dataset provided by Hollaback [11].



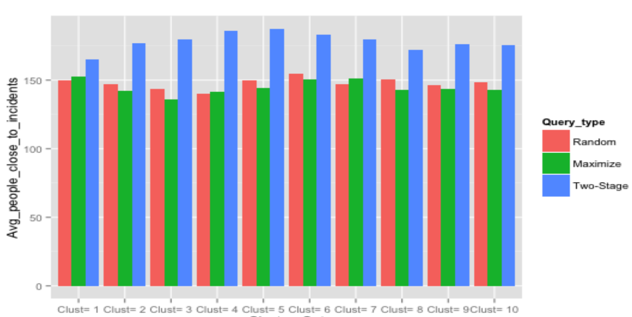
(a) Close node count with $FSP = 20\%$ of available resources.



(b) Close node count with $FSP = 40\%$ of available resources.



(c) Close node count with $FSP = 60\%$ of available resources.



(d) Close node count with $FSP = 80\%$ of available resources.

Figure 2: Average number of people close to the incidents (Close node count) as FSP varies.

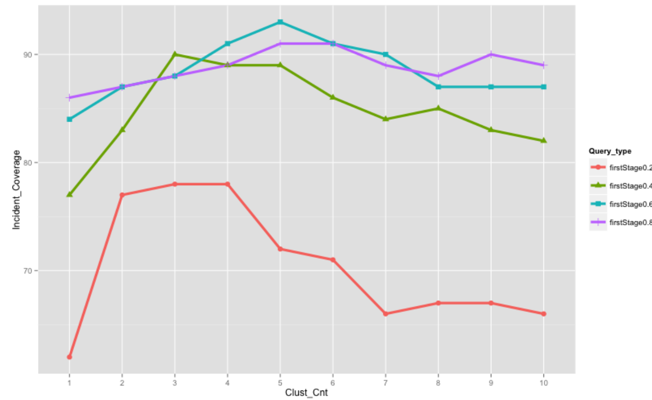


Figure 3: Incident coverage for different values of First stage percentage.

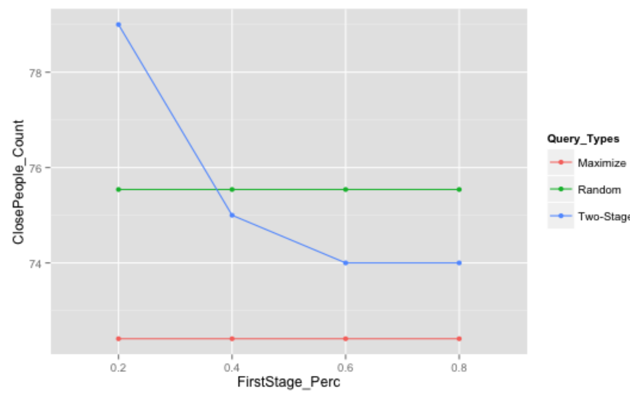


Figure 4: Close node count for different values of First stage percentage.

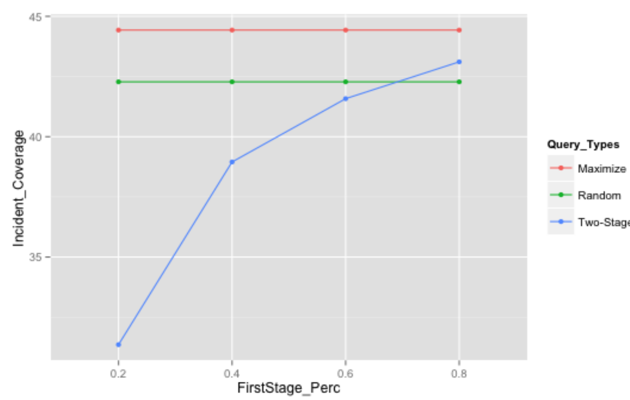


Figure 5: Incident coverage for different values of First stage percentage.

4.5.1 Data Overview

Hollaback is a non-profit movement powered by local activists in 92 cities and 32 countries to end street harassment. The Hollaback project collects data on street harassment events worldwide. Through the Hollaback phone app and the online platform, users can report stories of street harassment to share with the Hollaback community. This

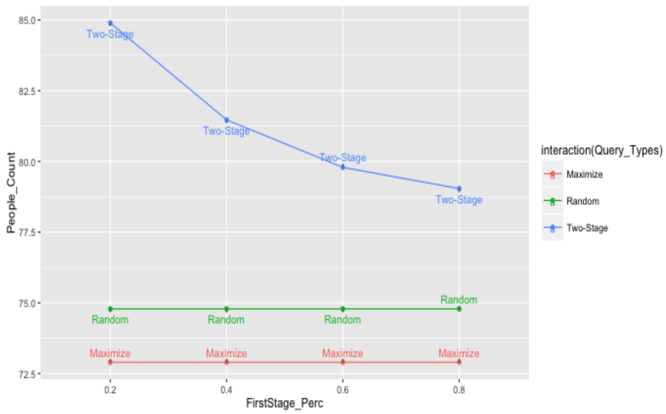


Figure 6: Close node count for different values of First stage percentage in the case of a long tail distribution.

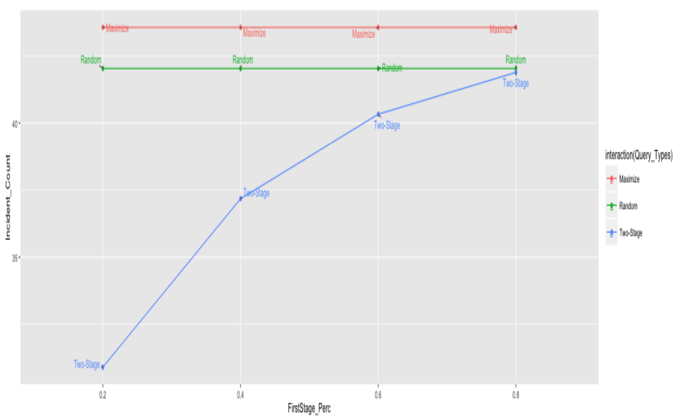


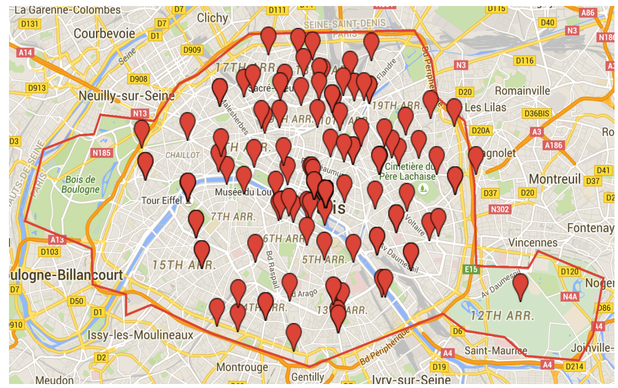
Figure 7: Incident coverage for different values of First stage percentage in the case of a long tail distribution.

empowers victims to speak out about everyday harassment and spread the word about the prevalence of these events. In some communities, local governments are informed in real-time about street harassment so that there is a system-wide level of accountability. In addition, the Hollaback app uses GPS to record a data set that represents the locations of street harassment events as a means of improving the collective understanding of street harassment and how it can be prevented. As of January 2016, over 8000 street harassment incidents have been recorded in their dataset since February 2011. It is on this data set that we wish to test the two-stage querying technique.

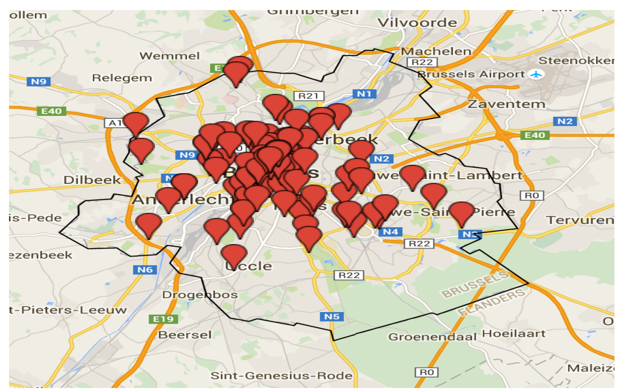
4.5.2 *Analysis*

From the Hollaback dataset, we select multiple cities for which we have enough harassment samples for statistical significance (i.e. more than 30 samples). We test the performance of Random selection, Dispersion maximization selection, and the two-stage querying on six different cities: Paris, Brussels, Berlin, Baltimore, Buenos Aires and Istanbul. These cities were in the top ten cities with respect to the number of harassment reports in this dataset. In this paper, we show results for Paris, Brussels, and Istanbul. The results for Berlin, Baltimore, and Buenosaires were consistent with the results shown in this paper.

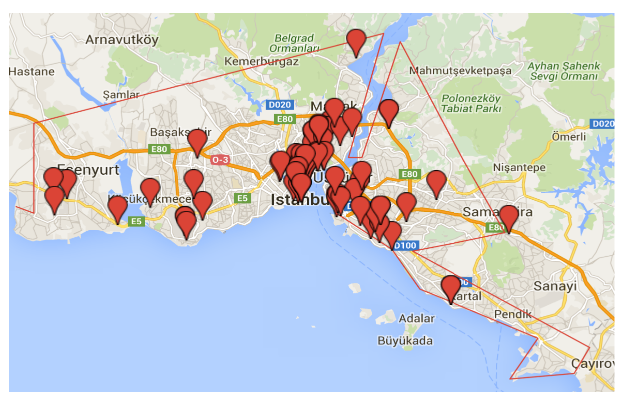
As a first step, we must parse the Hollaback dataset such that incidents reports are grouped by city. To do so, we use bounding box coordinates. We then draw the border lines for the different cities and remove any outliers from our datasets. Figure 8 shows the resulting distribution of events for the different 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. Istanbul had 87 reported incidents covering an area of 138 mi^2 on the left of Bosporus Strait and 69 mi^2 on the right.



(a) Paris.



(b) Brussels.



(c) Istanbul.

Figure 8: Distribution of harassment incidents across Paris, Brussels and Istanbul.

For each of the cities, we generate different variations of uniformly distributed crowd ($M = 100$) across the city. In this kind of analysis, the parameters, matrix dimension and incident count, are not generated by our analysis but rather taken from the dataset. In this case, we update the distance metric in Equation 1 and use the Haversine formula to calculate the great-circle distance between two points as follows:

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

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 (mean radius = 6,371km). We measure the Close node count and the Incident Coverage for all three querying techniques and plot the results in Figures 9 and 10, respectively. We notice that the Two-stage technique outperforms both the Random and Dispersion Maximization in terms of Close node count for all three cities. In terms of incident coverage, Figure 10 shows that dispersion maximization achieves maximum incident coverage. The figure also shows that the two-stage technique can achieve this maximum by setting the first stage percentage to be 80%. Figures 9 and 10 suggest that there is an inherent tradeoff between accuracy and coverage under constrained resources which we discuss in detail in later sections. The figures also suggest that the two-stage technique under setting the first stage percentage to be 80%, can achieve a balance between accuracy and coverage.

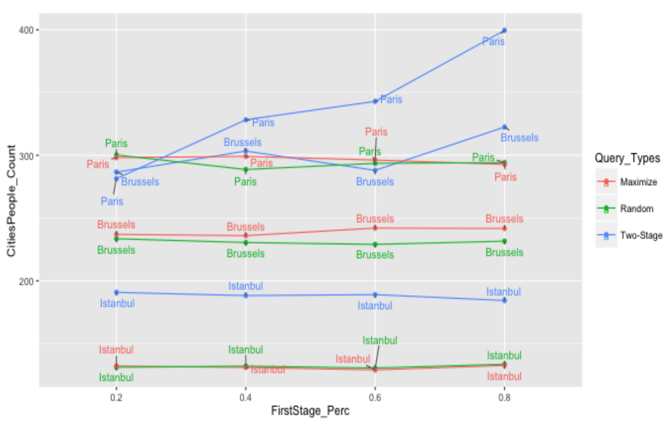


Figure 9: Close node count for different values of First stage percentage.

4.6 Stressing the two-stage querying technique (k=1)

After applying the two-stage technique to different datasets, we wish to look at how the different selection policies, in terms of identifying the first nearest neighbor to the different incidents. This is beneficial in case of targeting first responders in case of an emergency scenario or just in case of spatial task distribution where you want to select the nearest neighbors to maximize spatial task assignment. This can be viewed as stressing the selection policies in order to determine which one achieves a higher number of first nearest neighbors.

In order to study first nearest neighbors, we examine the

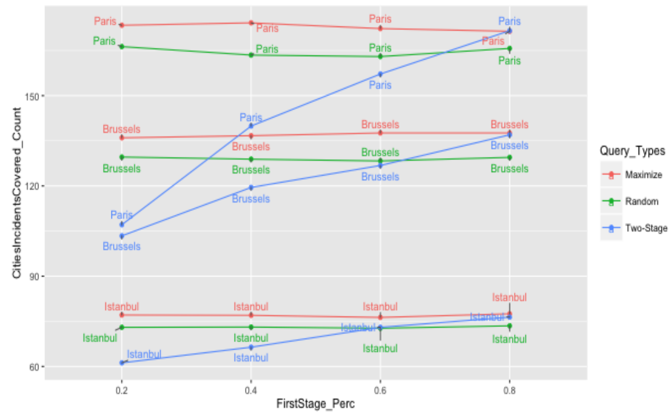


Figure 10: Incident coverage for different values of First stage percentage.

real Hollaback datasets for Paris, Brussels, and Istanbul. We examine the total close node count and for each incident, we look if we selected the first nearest neighbor in the queried users set or not. These results can be shown in Table 3, where NN denotes nearest neighbors, CNC denotes close node count, TS denotes the two-stage querying technique, Bruss denotes Brussels, and Istan denotes Istanbul. For Paris, the two-stage technique achieves a 9.5% surge on average in selecting nearest neighbors in comparison to dispersion maximization and 19% surge in comparison to random selection. For Brussels, the surge is 14.2% and 21.35% in comparison to dispersion maximization and random selection respectively. For Istanbul, the surge percentages were 26.7% and 36.5%.

City	Random	Maximize	TS ($FSP=0.2$)	TS ($FSP=0.4$)	TS ($FSP=0.6$)	TS ($FSP=0.8$)
Paris-NN	58	63	54	69	69	84
Paris-CNC	300	298	281	328	343	399
Bruss-NN	48	51	55	58	55	65
Bruss-CNC	233	237	286	303	288	322
Istan-NN	26	28	36	37	35	36
Istan-CNC	131	132	190	188	188	184

Table 3: Nearest neighbors and close node count for Paris, Brussels, and Istanbul.

5. DISCUSSION

5.1 Tradeoff

In the previous section, we examined the performance of the two-stage querying technique and we observed that as FSP decreases, the number of close node count tend to increases. We also observed that in this case the same node can be in the K nearest neighbors for multiple incidents. This means that the algorithm tends to select central nodes that are in proximity with other incidents. This is beneficial in cases where the centrality of nodes is important to the problem, e.g. minimizing trip costs to these incidents and maximizing task assignments. This observation ensures diversity of the feedback i.e. instead of relying on a small number of nodes close to the incidents, we have a greater sample that can contribute to the measurement. On the other hand, as FSP increases so does the probability of catching more incidents in the spatial environment which is crucial in applications where coverage is important and where a false positive is less expensive than a false negative. This is due to the fact that more nodes are selected in the first stage and fewer nodes in the second phase. The conclusion is that under resource constrained conditions, there is a tradeoff between accuracy and coverage.

5.2 Technique variants

5.2.1 *Trust based responses*

In our second stage of our algorithm, we select users based on the pivot nodes that provided a positive feedback in the first stage. To incorporate trust into the algorithm, trust-based algorithms can provide feedback about certain nodes and their feedback. If some of the nodes queried in the first stage of the algorithm were deemed trust unworthy, the second stage can be divided among two querying parts. The first part is the KNN for the trustworthy-nodes and a second attempt of dispersion maximization.

5.2.2 *Prior knowledge availability*

The two-stage querying technique does not assume any knowledge about the distribution of events. Given some prior info about the distribution, the algorithm can be tailored to take the prior distribution into account. The idea is to divide the spatial area into bounding regions and for each region we give a specific weight that reflects the probability of occurrence in that bounding regions. For example, Figure 11 shows Baltimore divided into four bounding regions denoted as br1, br2, br3, and br4. Using the knowledge that br3 has more incidents than br2 which has more incidents than br1 and br4, a higher weight should be given to br3. In particular, $br3_w > br3_w > br4_w > br1_w$ where the subscript w denotes the weight assigned to the bounding region. The next step would be applying the algorithm on the different bounding regions taking into account the weights assigned when allocation resources as shown in 2.

Algorithm 2 Bounding regions two-stage variation.

```
function selectUsersBRs (BRs[], w[], N, FSP)
    selectedUsers = {}
    BRQuota = calculateQuota(BRs[], w[], N)
    for br in BRs do
        brQuota = BRQuota(br)
        brUsers = selectUsersFromGrid(FSP, brQuota)
        brUsers.append(brUsers)
    return selectedUsers
```

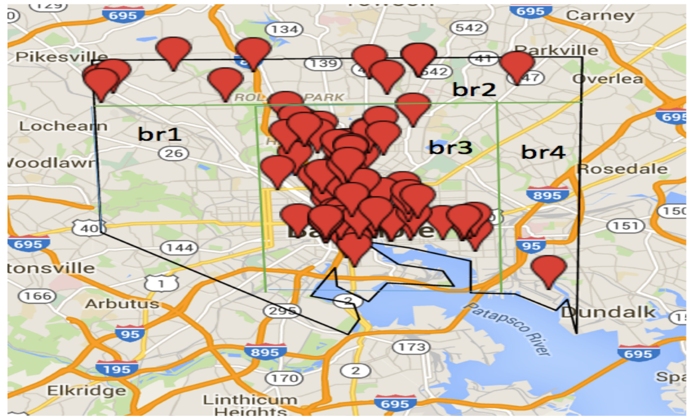


Figure 11: Baltimore divided into bounding regions depending on prior knowledge of harassment occurrence.

6. CONCLUSION

This paper proposed a two-stage spatial querying technique that attempts to probe nodes that are closer to the incidents in a $2D$ spatial environment. The algorithm maximizes the dispersion of location in the first stage and selects the K nearest neighbors in the second stage based on nodes' feedback. The experimental evaluation confirms the applicability of proposed approach. Important aspects that need to be taken into consideration include the sensing range and the sensing accuracy of the nodes. Nodes with accurate sensing and larger sensing ranges will provide a more robust effect on the two-stage technique. Another important parameter that can impact the performance of the technique is the distribution of the available nodes in the spatial environment. To ensure an optimal performance, nodes should be placed uniformly in the environment if the incident distribution is not known or following an approximate distribution of the incidents. Another important factor when choosing the number of nodes to query is the granularity of the incidents. For instance, when using human as sensors, incidents such as disasters can be sensed/detected by a lot of people in the range of the incident. On the other hand, incidents such as harassment require the presence of the human being at the same exact place of the incident otherwise, the incident would not be detected.

7. ACKNOWLEDGMENTS

The authors would like to thank Hollaback for sharing their collected dataset and taking the time to answer questions.

8. REFERENCES

[1] Eighty sexual assaults in one day - the other story of Tahrir Square. `http://www.theguardian.com/world/2013/jul/05/egypt-women-rape-sexual-assault-tahrir-square`, 2016. [Online; accessed March-2016].

[2] Human Shield Formed In Tahrir Square To Protect Women From Sexual Assault . `http://www.huffingtonpost.com/2013/07/03/human-shield-tahrir-square-egypt-sexual-violence_n_3540970.html`, 2016. [Online; accessed March-2016].

[3] E. Agapie, J. Teevan, and A. Monroy-Hernández. Crowdsourcing in the field: A case study using local crowds for event reporting. In *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications magazine*, 40(8):102–114, 2002.

[5] T. Blaschke, G. J. Hay, Q. Weng, and B. Resch. Collective sensing: Integrating geospatial technologies

to understand urban systems-an overview. *Remote Sensing*, 3(8):1743–1776, 2011.

[6] G. Cardone, L. Foschini, P. Bellavista, A. Corradi, C. Borcea, M. Talasila, and R. Curtmola. Fostering participaction in smart cities: a geo-social crowdsensing platform. *IEEE Communications Magazine*, 51(6):112–119, 2013.

[7] A. Crooks, A. Croitoru, A. Stefanidis, and J. Radzikowski. # earthquake: Twitter as a distributed sensor system. *Transactions in GIS*, 17(1):124–147, 2013.

[8] D. Deng, C. Shahabi, and U. Demiryurek. 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*, pages 324–333. ACM, 2013.

[9] N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2149–2158, 2013.

[10] R. Fontugne, J. Ortiz, N. Tremblay, P. Borgnat, P. Flandrin, K. Fukuda, D. Culler, and H. Esaki. Strip, bind, and search: a method for identifying abnormal energy consumption in buildings. In *Proceedings of the 12th international conference on Information processing in sensor networks*, pages 129–140. ACM, 2013.

[11] Hollaback. Read and Share Stories. When it comes to street harassment, you are not alone. `http://www.ihollaback.org/share/`, 2015. [Online; accessed July-2015].

[12] J. Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.

[13] L. Kazemi and C. Shahabi. Geocrowd: enabling query answering with spatial crowdsourcing. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 189–198. ACM, 2012.

[14] B. Krishnamachari, D. Estrin, and S. Wicker. The impact of data aggregation in wireless sensor networks. In *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops, 2002*, pages 575–578. IEEE, 2002.

[15] R. Lee and K. Sumiya. Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection. In *Proceedings of the 2nd ACM SIGSPATIAL international workshop on location based social networks*, pages 1–10. ACM, 2010.

[16] Y. Liu, T. Alexandrova, and T. Nakajima. Using stranger as sensors: temporal and geo-sensitive question answering via social media. In *Proceedings of the 22nd international conference on World Wide Web*, pages 803–814. International World Wide Web Conferences Steering Committee, 2013.

[17] B. S. Manoj and A. H. Baker. Communication challenges in emergency response. *Communications of the ACM*, 50(3):51–53, 2007.

[18] D. Marco, E. J. Duarte-Melo, M. Liu, and D. L. Neuhoff. On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data. In *Information Processing in Sensor Networks*, pages 1–16. Springer, 2003.

[19] S. Pattem, B. Krishnamachari, and R. Govindan. The impact of spatial correlation on routing with compression in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 4(4):24, 2008.

[20] J. Sallai, G. Balogh, M. Maroti, A. Ledeczi, and B. Kusy. Acoustic ranging in resource-constrained sensor networks. In *International Conference on Wireless Networks*, page 467. Citeseer, 2004.

[21] M. Venanzi, A. Rogers, and N. R. Jennings. Crowdsourcing spatial phenomena using trust-based heteroskedastic gaussian processes. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[22] M. Venanzi, A. Rogers, and N. R. Jennings. Trust-based fusion of untrustworthy information in crowdsourcing applications. In *Proceedings of the 2013 international conference on autonomous agents and multi-agent systems*, pages 829–836. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[23] H. Yu, C. Miao, Z. Shen, and C. Leung. Quality and budget aware task allocation for spatial crowdsourcing. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1689–1690. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[24] H. Yu, Z. Shen, C. Miao, and B. An. A reputation-aware decision-making approach for improving the efficiency of crowdsourcing systems. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1315–1316. International Foundation for Autonomous Agents and Multiagent Systems, 2013.