Reliable Crowdsourced Event Detection in SmartCities

Ioannis Boutsis, Vana Kalogeraki
Department of Informatics
Athens University of Economics and Business, Greece
{mpoutsis, vana}@aueb.gr

Dimitrios Gunopulos
Department of Informatics & Telecommunications
University of Athens, Greece
dg@di.uoa.gr

***Abstract*—In recent years crowdsourcing systems have shown to provide important benefits to Smartcities, where ubiquitous citizens, acting as mobile human sensors, assist in responding to signals and providing real-time information about city events, to improve the quality of life for businesses and citizens. In this paper we present REquEST, our approach to selecting a small subset of human sensors to perform tasks that involve ratings, which will allow us to reliably identify crowdsourced events. One important challenge we address is how to achieve reliable event detection, as the information collected from the human crowd is typically noisy and users may have biases in the answers they provide. Our experimental evaluation illustrates that our approach works effectively by taking into consideration the bias of individual users, approximates well the output result, and has minimal error.**

***Index Terms*—crowdsourcing; smart cities; bias**

I. Introduction

With overwhelming population growth on the rise, we are moving toward a world where digital technology and intelligent design are harnessed to create smart, sustainable cities that offer creative services to improve the quality of life for their businesses and citizens. Making cities smarter is emerging as a key area of focus for governments and the private sector to address the projected demands of cities in the future. SmartCities use technologies from a wide range of origins, from fixed and mobile sensors to large-scale monitoring infrastructures, and can come from public or private sources. These can provide advanced services such as smarter urban transport networks where sensors, cameras, and global positioning system (GPS) devices provide information on traffic, identifying congestion and recommending alternate routes to improve travel times and reduce emissions, resource management processes (*e.g.*, upgraded water supply and waste disposal facilities), and more efficient ways to light and heat buildings to optimize energy consumption in smart buildings.

One significant tool in SmartCities is the use of ubiquitous citizens, acting as mobile human sensors, that are able to assist in responding to signals and providing real-time information about city events. Recently, several applications driven by city authorities have emerged, such as the JRA Find and Fix app[1] where users report road related defects for the maintenance, repair and development of Johannesburg's road network and storm water infrastructure, and the CrowdAlert app[2] that we have developed (shown in figure 1), where citizens contribute traffic related events (*e.g.,* congestion) through opt-in crowdsourcing mechanisms. Furthermore, CrowdAlert enables users to receive traffic information observed through city-wide heterogeneous sensor network infrastructures that exist in SmartCities such as road sensors, bus sensors, traffic cameras and feedback from the human crowd [1], [2].

SmartCity apps provide important benefits: First, they put new capabilities in the hands of city administrators, where innovative technologies and system infrastructures work in concert to provide insights, identify events of interest, which allow them to effectively cope with emergency situations. Second, they provide a platform and engagement mechanisms (through crowdsourcing, open data, etc.) where citizens actively participate and contribute data into the system toward implementing city wide solutions. Crowdsourcing is the process of soliciting contributions from the human crowd, a large group of self-identified city volunteers, where each contributor can perform a task via a mobile device, this task adds a small portion to the final result. Tasks typically cover a wide variety of domains including traffic monitoring systems where users are asked to identify the volume of the traffic from their corresponding location (such as in Waze[3] or in CrowdAlert), or social feedback applications where users are asked to rate or recommend social venues such as bars and restaurants (*e.g.*, Foursquare[4]), etc.

Thus, one important challenge in SmartCities is how reliable are the responses collected from the human crowd and what process is used to verify the received information. While crowdsourcing has proven to be a cost-effective approach to soliciting input about an event, compared to traditional methods such as employing human experts that check all data manually, humans are prone to errors which can greatly affect the result of a crowdsourcing task. This is attributed to the following two main factors: First, users have different abilities which may be unknown to the task requester a priori, thus selecting the appropriate set of users to perform a task is not

[1]http://www.jra.org.za/index.php/find-and-fix-mobile-app

[2]http://crowdalert.aueb.gr
[3]https://www.waze.com/
[4]https://foursquare.com/

Copyright 978-1-5090-1169-8/16/$31.00 © 2016 IEEE

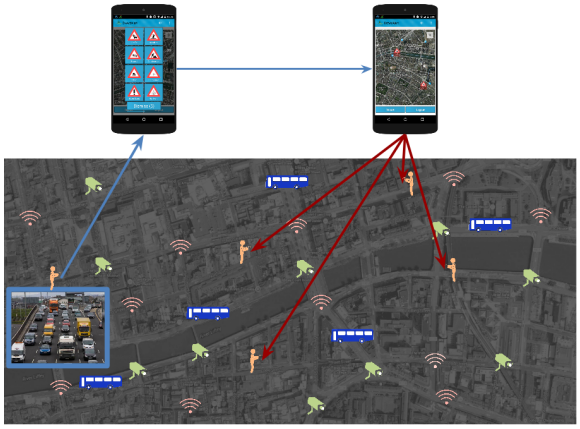

Fig. 1. SmartCity Traffic Monitoring Paradigm.

an easy challenge. Second, verifying user responses obtained from a small set of mobile sensors is not easy since users typically provide only a few responses, as they require human effort, which are subjective and might contain bias.

In this paper we address the problem of reliable crowdsourced event detection in SmartCities. We focus on selecting a small set of human users to perform tasks that involve ratings, such as rating the traffic congestion or air pollution in their geographical area. The challenge is how to select a subset of users whose aggregated responses would closely approximate the final response that the entire set of users would provide. In order to provide accurate results we need to deal with the user bias of the selected users, since many recent examples (*e.g.*, political elections) indicate that the results may contain a lot of noise when the bias is not considered properly, or when the sample is not random. Thus, in our work we focus on taking into account the user bias before aggregating the user answers, and we show that such an approach has a great improvement towards estimating the rating that the whole set of users would provide.

Recent works in the literature aim to identify a sample from the human crowd to determine the most probable answer for each task [3], [4], while other works study the problem of truth discovery in crowdsourcing systems [5], [6], [7]. However, our problem is radically different since we do not aim at determining the true answer among a set of predefined answers. Since we deal with rating systems all answers might be subjective but they are truthful, and our goal is to determine the average rating that we would retrieve if all users participated. Task assignment approaches have also been proposed for crowdsourcing environments [8], [9], [10], including work from our group [11], [12]. Although these works aim at selecting a good set of users to perform tasks based on individual user characteristics, they do not focus on estimating quantitative values, such as ratings, and they do not consider user bias in the responses.

In this paper we present REquEST (Reliable crowdsourcEd Event detection in SmartciTies), our approach to select the most appropriate set of mobile human sensors to perform a task that minimizes the error and allow us to identify crowdsourced events reliably. We summarize our contributions below:

- We present REquEST, our approach that exploits the human crowd to perform crowdsourcing tasks in smart cities that involve ratings. The goal of REquEST is to approximate the rating that all users would provide using only a small set of users.
- REquEST aims to select human workers with minimal bias by examining the behavior of the users in their previously executed tasks. Moreover, it exploits linear regression to estimate user bias in each task and attempts to eliminate it when the user provides her rating, before aggregating the responses.
- We present experimental results to evaluate REquEST and we show that it can approximate the rating that all users would provide with a minimal error, even when the sample size is small.

II. System Model

We assume a crowdsourcing system comprising a set of mobile users, denoted as $u \in U$, that act as human sensors and participate to the system through their mobile devices. Human workers register into the system to receive Crowdsourcing tasks $t \in T$. Each human worker u is associated with the following attributes: $\langle id_u, lat_u, long_u, bias_u, prev_u[] \rangle$, where id_u is the worker's unique identifier in the system, $lat_u, long_u$ correspond to the user's current location in terms of latitude, longitude, $bias_u$ represents the user bias which captures the likelihood of a user to make specific mistakes (this depends on user capabilities and expertise) when estimating the value of a task, and $prev_u[]$ is used to store information about the tasks completed by the worker u.

In this work we focus on crowdsourcing tasks where users respond with a rating based on the issued query. Thus, every crowdsourcing task $t \in T$ has the following attributes: $\langle id_t, lat_t, long_t, description_t \rangle$, where id_t is the identifier of task t, lat_t and $long_t$, represent the geographical location that the task refers to, and $description_t$ contains the textual description of the task, (*e.g.*, Report the level of traffic congestion at your location from 1 (No Traffic) to 10 (Heavy Traffic)). Hence, in this example, each user will respond to the task with a rating denoted as $a_{u,t}$ with a value in the range of $[1, 10]$. We assume that users have their own biases and each bias is independent of other users' biases.

Our goal is to identify the correct value for the task via aggregating the responses received from all users queried about the task. We denote as val_t^X the estimated value for task t, computed based on input from all users in set X. Since we focus on tasks that involve ratings, we assume that user responses can be subjective but truthful. This corresponds to behaviors where users express their own ideas and knowledge when labeling tasks, based on their personal abilities, characteristics and expertise. This occurs because users will be subjective when they provide numerical responses (*e.g.*, when rating traffic congestion or estimating rain precipitation).

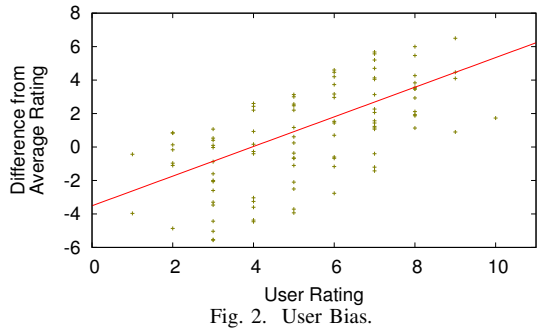

Fig. 2. User Bias.

Problem Definition. In this work we aim at collecting input from multiple human sensors (workers) regarding an ongoing event. Collecting responses mitigates these biases when collecting a large number of user responses and then aggregating them to negate the effect of individual biases. We perform sampling on the user set to retrieve such information. Given the amount of users that we can query, our goal is to determine a set of users to sample that will allow us to estimate the output that we would get if all users were queried. More formally:

Assume a set of human workers U located near the location of task t: lat_t, lon_t. Our goal is to identify an appropriate set of workers $S \subset U$ to provide a response $a_{u,t} \forall u \in S$ for task t, whose location lat_u, lon_u is within a predefined threshold to the location of the queried task. We can then estimate the final result val_t^S taking into account user $bias_u \forall u \in S$ in order to minimize the error compared to the average rating obtained from all users in U: $|val_t^S - \frac{\sum_{u \in U} a_{u,t}}{|U|}| \rightarrow 0$. After estimating val_t^S, the system verifies the user responses and re-estimates the $bias_u$ for each user that participated in the sampling process.

III. The REquEST approach

In this section we present our approach that aims at identifying an appropriate subset of mobile human workers S that will enable us to compute reliably the output of tast t. Once the human sensors are identified, we query them and retrieve their crowdsourcing answers $a_{u,t}$. Finally, we determine the output of the task while considering user biases, and use the estimated result to update the user biases.

User Bias. Recent works have shown that users have bias when responding to crowdsourcing tasks [13] and several approaches have been proposed to eliminate bias from their responses [14], [15], [16], [17]. However, existing approaches are not sufficient since (i) they either focus on binary responses [14], [15] instead of numerical responses, (ii) they use active learning approaches [13] which need several iterations to converge, while in crowdsourcing processes users typically answer sparsely, or (iii) they use hybrid models where the bias may depend either on the worker's confusion matrix or on a population-wide representation that can introduce additional noise [17]. On the contrary, we propose an approach that aims to eliminate user bias from the responses which works efficiently even with a small set of numerical responses obtained from the users. Unlike existing approaches, since we focus on ratings, we are able to model bias as a linear function of the user responses which is more flexible and easier to compute at run-time.

The intuition in our approach is that user ratings have a bias, defined as a linear function of their answers with respect to the difference of their ratings from the average rating when all users in U are considered. We visualize this relationship exploiting our dataset (further discussed in in the experimental evaluation section), that contains ratings regarding the traffic conditions provided by human users. In figure 2 we show for a single user, who has provided the highest amount of ratings in our dataset, the relationship among her ratings compared to the difference of her ratings from the average rating for each task. This relationship can be captured with a linear function, denoted as the user bias. We note that similar behavior exists in all users as well as in other datasets that capture rating that we have examined.

In our setting we assume that each user $u \in S$, selected from the sampling process provides an answer $a_{u,t}$ with a bias $b(a_{u,t})$ and thus: $a_{u,t} = \frac{\sum_{u \in U} a_{u,t}}{|U|} + b(a_{u,t})$. The bias $b(a_{u,t})$ is defined as a linear function of the user response. This enables us to estimate the difference from the average value for each user response $a_{u,t}$.

REquEST exploits linear regression to adjust the user bias estimation whenever the user provides a response. Linear regression is a useful tool in many applications to find the hedge ratio between two assets. In our scenario these assets are defined from the user response $a_{u,t}$ and its difference from the average value of the task. Thus, we record the response $a_{u,t}$ provided for each task and the respective difference from the average rating and we define the ratio among these two dimensions. This is computed easily using simple linear regression[18] that produces a linear function: $b(a_{u,t}) = \mu * a_{u,t} + \nu$ where μ is the slope and ν is the interval of the line, which are estimated from the linear regression. Thus, we can estimate the difference of each user rating compared to the average rating from all users by computing $b(a_{u,t})$ for each rating $a_{u,t}$.

Selecting users for the task. For each task we select among the nearby workers with minimal bias in their responses. Thus, we first extract the set of available users whose distance is smaller than a predefined threshold from the location of the event. Then, we select among those users that will provide answers with minimal bias. Assuming that the list of possible ratings is defined as R, we compute for each user the score $\sum_{\forall a_{u,t} \in R} |b(a_{u,t})|$, that accumulates the absolute difference of the user rating compared to the average rating retrieved from all users, for each possible answer $a_{u,t}$. Hence, the score provides an estimation of the bias that the user may introduce in her ratings and a small score implies minimal bias. Thus, we select the top-K users with the smallest scores.

Computing the output of the task. The next step is to determine the output of the task. As mentioned above, human users have biases in their responses [13], and thus, we need to

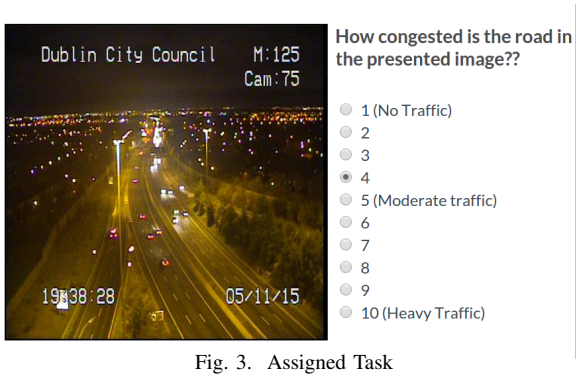

Fig. 3. Assigned Task

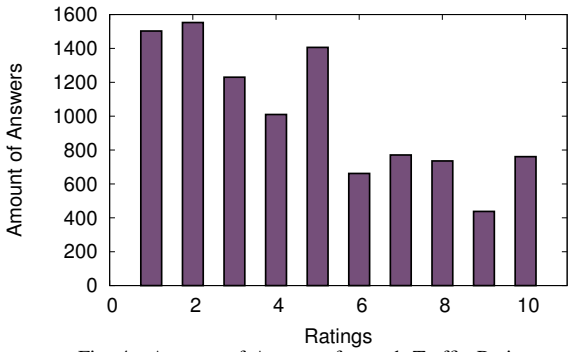

Fig. 4. Amount of Answers for each Traffic Rating

compute the rating that all users would provide, subject to user biases. We compute the output of the task using the following equation:

$$val_t^S = \frac{\sum_{u \in S}(a_{u,t} - b(a_{u,t}))}{|S|} \quad (1)$$

Thus, we compute the average value of the retrieved ratings after eliminating the estimated bias that each individual user introduces in her rating.

Updating user bias. Once we estimate the aggregate values, we update the function $b(a_{u,t})$ for the users that participated in the crowdsourcing task. This is achieved for each user $u \in S$ that provided an answer $a_{u,t}$ by inserting the new rating and the respective difference from the estimated value val_t^S in the set of user responses and perform linear regression to update the parameters μ, ν.

Since our approach assumes that we should estimate the correlation among users, we may face bootstrapping issues. Thus, in the first iterations of assigning crowdsourcing tasks, we can select some users randomly in order to train the system with their answers.

IV. Experiments

For our experiments we have developed a dataset that includes user ratings for a number of images related to traffic events. The images are extracted from traffic cameras provided by the Dublin City Council (DCC), the local authority in Dublin that manages traffic.

DCC maintains approximately 270 traffic cameras located throughout the city that provide images every 10 minutes. Thus, it is impossible for a human operator to manually check all images to identify whether there are traffic events and also classify the types of traffic events. On the other hand, users can easily tag the traffic in their own location and automate the procedure for DCC. However, as users have bias we cannot trust each individual response, but given a sample of ratings we are able to compute the traffic rating that all users would provide. We have implemented our REquEST approach in Java. We performed the experiments on an Intel Core i7 PC with 16GB of RAM, that provides a controlled environment.

In order to extract traffic ratings from real human users we performed the following experiment in the CrowdFlower platform [19] that employs users to perform crowdsourcing tasks. We extracted 287 individual images from the traffic cameras and we asked from users to classify traffic. An example of such a task as shown in figure 3. Thus, users are presented with an image out extracted from one of the 287 cameras and the following question "How congested is the road in the presented image?", and then users respond with a rating from 1 (No Traffic) up to 10 (Heavy Traffic). We received answers from 157 individual users that tagged from 1 up to 100 individual images (different users can rate the same image). Hence, our dataset consists of 10,070 individual traffic ratings for the 287 traffic images.

In figure 4 we present the total amount of answers that we retrieved for each of the ratings for all tasks. As can be observed, "2" was the most common traffic rating, while most of the ratings were between 1 (No Traffic) and 5 (Moderate Traffic). This is because the majority of the traffic images do not capture heavy traffic or congestion.

Additionally, we present in figure 5 the average rating, provided by the users, for each individual task. As can be observed, 38% of the traffic images is rated within Moderate and Heavy Traffic; these are the images that the city personnel should take into account when managing the city traffic.

Figure 6 illustrates the average rating for the tasks performed by each individual user. As the figure shows, the majority of the users provide ratings with a small bias on average (users with identifiers between 30 and 120), while other users seem to underestimate traffic (120-157) or overestimate traffic (0-30). These users prove that bias should be taken into account when aggregating user ratings since their responses are different from the majority of the users.

In the following we evaluate our approach in terms of efficiency to accurately determine the response that users from a small sample would provide. In order to achieve that, we use the root mean square error (RMSE) to evaluate the accuracy which is defined as:

$$RMSE = \sqrt{\frac{1}{|T|}\sum_t (val_t^S - \frac{\sum_{u \in U_t} a_{u,t}}{|U_t|})^2}$$

where U_t represents the complete set of users in our dataset that have rated task t. We use the RMSE metric as it penalizes large errors more and a smaller RMSE indicates better performance.

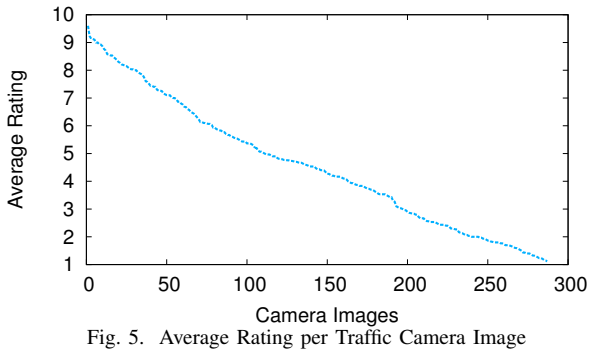

Fig. 5. Average Rating per Traffic Camera Image

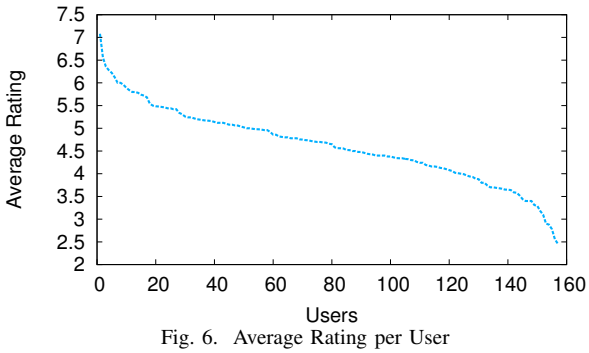

Fig. 6. Average Rating per User

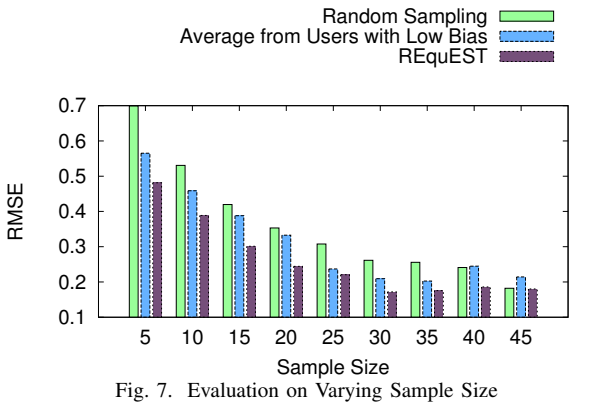

Fig. 7. Evaluation on Varying Sample Size

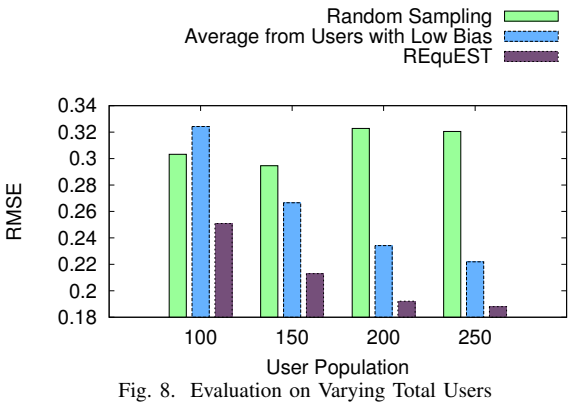

Fig. 8. Evaluation on Varying Total Users

We compare REquEST in terms of RMSE with the following baseline approaches: (i) **Random Sampling**, that queries a random set of users and computes their average rating, (ii) **Average from Users with Low Bias**, that exploits our approach to select the set of users to be queried and computes their average value without considering regression to eliminate bias. The experimental evaluation focuses on evaluating the benefit of our approach compared to the baselines, when varying the Sample size and the Set of Total Users.

Figure 7 presents the RMSE score for the traffic images under various numbers of sample size (5-45). Obviously, the RMSE decreases as we increase the sample size for all approaches. However, as can be observed, our approach manages to improve the RMSE in all cases, especially when the sample size is small. Thus, for a sample size of 5 users the Random Sampling performed a value of RMSE of 0.69, the Average from Users with Low Bias performed 0.56, while REquEST performed 0.48.

In figure 8 we illustrate how RMSE behaves when we keep the sample size to 25 users, but we vary the amount of total users from 100 to 250 out of 287 users who are available in total. As can be observed, our approach outperforms the two baselines in all cases. Moreover, it is clear that the Random Sampling approach is only slightly affected by the amount of total users, while the other two approaches that aim to select good workers improve their performance as more users become available. Hence, the approach that computes the Average from Users with Low Bias improves the RMSE from 0.32 (100 users) to 0.22 (250 users), while REquEST reduces the RMSE from 0.25 for 100 users to 0.18 for 250 users.

V. Related Work

Several approaches have been proposed in the literature for task assignment in crowdsourcing environments, including our previous work [11] [12] that aims to assign tasks to humans in order to satisfy reliability and real-time requirements. Other approaches that consider user reliability have focused on minimizing the number of task assignments to fulfill an overall reliability constraint [8], selecting users based on their quality [9] and considering human factors, such as expertise, wage requirements and availability [10]. However, in our setting we assume tasks where the answers can be subjective and thus all user responses can be considered as reliable, although they may variate a lot. Moreover, while these works focus on selecting a good (reliable) sample to perform the tasks, they do not focus on estimating quantitative values, such as the rating that the whole set of users would provide for a data item, and also they do not consider the user bias in their responses.

Existing works have also studied the problem of selecting a sample of users. Authors in [3] aim at selecting a sample of reliable users to approximate the crowd's majority vote by collecting opinions from a subset of the crowd. However, they do not consider the user bias and our goal is different as we aim at approximating the solution of the aggregation based on all user responses. Daly *et al.* in [4] also select a small subset of the users to respond based on their individual characteristics (reputation, mobility, etc.) but they do not focus on representing the whole set of human users.

Active learning approaches have also been proposed for truth discovery in crowdsourcing systems. Authors in [5] capture the sources of bias by describing labelers influenced

by random effects and propose an active learning approach to learn the model. Similarly, in [6] they aim to learn the expertise and reliability of each user based on Gibbs Sampling to estimate the true answer of a task, while in [20] they propose an approach, based on Gibbs Sampling, to determine whether an event occurs in a spatial area, according to crowdsourcing reports. However, all previous approaches have no control on the user selection and they need a lot of iterations until the model converges which is unrealistic for crowdsourcing environments where most of the users are transient.

In [21] they aim to estimate the reliability of crowd-workers, the difficulty of the different tasks, and the probability of the true labels using a unified model. However, they focus on labeling items rather than ratings which implies subjective but truthful responses. Authors in [22] integrate machine learning techniques into crowdsourced databases to minimize the number of questions asked to the crowd, allowing crowdsourced applications to scale. However, in case of inappropriate workers the variance can be high, causing the system to ask too many questions or provide erroneous responses.

Techniques that aim to estimate user biases in crowdsourcing environments have recently been proposed in the literature. In [14], [15] they study the data annotation bias when data items are presented as batches to the workers. However, they focus on binary answers and their goal is to correctly categorize each data item instead of estimating the aggregated response that all users would provide. Authors in [13] show that crowdsourcing users have both bias and variance and they propose an approach to recover the true quantity values for crowdsourcing tasks. However, their approach needs several iterations (tasks) to converge which is a strong assumption for crowdsourcing and they focus on debiasing existing responses rather than estimating the response that would be provided from all users. Authors in [17] aim to solve the above problem by using a hybrid approach where the user bias depends heavily on a unified population-wide representation for workers with small number of reports and uses an accurate worker confusion matrix for each worker with a large number of reports. On the contrary, in REquEST, user bias depends on the user responses even for a small number of responses, due to our flexible representation of user bias, since a population wide representation might introduce additional noise. Das *et al.* in [16] focus on debiasing crowdsourcing answers to estimate the average innate opinion of the social crowd with a small number of samples. However, their approach depends on the social dependency among users that does not exist in our setting. In [23] they investigate a game-theoretic scheme that motivates users with monetary rewards to counter bias, but they assume that bias is introduced when users adopt heuristic strategies to solve the task, while we assume that users answer subjectively.

VI. Conclusions

The paper presents REquEST, our approach to reliable crowdsourced event detection in smart cities. We present a methodology that selects a sample of mobile sensors from the human crowd to acquire their responses for a specific event, and processes them to estimate the response that we would get if all users would participate. Our experimental results show that our approach is effective and has minimal error.

Acknowledgment

This research has been financed by the European Union through the FP7 ERC IDEAS 308019 NGHCS project and the Horizon2020 688380 VaVeL project.

References

[1] A. Artikis, M. Weidlich, F. Schnitzler, I. Boutsis, T. Liebig, N. Piatkowski, C. Bockermann, K. Morik, V. Kalogeraki, J. Marecek, A. Gal, S. Mannor, D. Gunopulos, and D. Kinane, "Heterogeneous stream processing and crowdsourcing for urban traffic management," in *EDBT*, Athens, Greece, March 2014, pp. 712–723.

[2] N. Zacheilas, V. Kalogeraki, N. Zygouras, N. Panagiotou, and D. Gunopulos, "Elastic complex event processing exploiting prediction," in *Big Data*, Santa Clara, CA, Oct 2015, pp. 213–222.

[3] Ş. Ertekin, C. Rudin, and H. Hirsh, "Approximating the crowd," *Data Mining and Knowledge Discovery*, vol. 28, no. 5-6, 2014.

[4] E. Daly, M. Berlingerio, and F. Schnitzler, "Crowd sourcing, with a few answers: Recommending commuters for traffic updates," in *RecSys*, Vienna, Austria, September 2015, pp. 253–256.

[5] F. L. Wauthier and M. I. Jordan, "Bayesian bias mitigation for crowdsourcing," in *NIPS*, 2011, pp. 1800–1808.

[6] F. Ma, Y. Li, Q. Li, M. Qiu, J. Gao, S. Zhi, L. Su, B. Zhao, H. Ji, and J. Han, "Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation," in *KDD*, Sydney, Australia, August 2015, pp. 745–754.

[7] S. Zhi, B. Zhao, W. Tong, J. Gao, D. Yu, H. Ji, and J. Han, "Modeling truth existence in truth discovery," in *KDD*, Sydney, Australia, Aug 2015.

[8] D. R. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *NIPS*, Granada, Spain, December 2011.

[9] R. Khazankin, H. Psaier, D. Schall, and S. Dustdar, "Qos-based task scheduling in crowdsourcing environments," in *ICSOC*, Paphos, Cyprus, December 2011, pp. 297–311.

[10] S. B. Roy, I. Lykourentzou, S. Thirumuruganathan, S. Amer-Yahia, and G. Das, "Task assignment optimization in knowledge-intensive crowdsourcing," *The VLDB Journal*, pp. 1–25, 2015.

[11] I. Boutsis and V. Kalogeraki, "On task assignment for real-time reliable crowdsourcing," in *ICDCS*, Madrid, Spain, June 2014, pp. 1–10.

[12] ——, "Crowdsourcing under real-time constraints," in *IPDPS*, Boston, MA, May 2013, pp. 753–764.

[13] R. W. Ouyang, L. Kaplan, P. Martin, A. Toniolo, M. Srivastava, and T. J. Norman, "Debiasing crowdsourced quantitative characteristics in local businesses and services," in *IPSN*, Seattle, WA, April 2015.

[14] H. Zhuang, A. Parameswaran, D. Roth, and J. Han, "Debiasing crowdsourced batches," August 2015.

[15] H. Zhuang and J. Young, "Leveraging in-batch annotation bias for crowdsourced active learning," in *WSDM*, Shanghai, China, Jan 2015.

[16] A. Das, S. Gollapudi, R. Panigrahy, and M. Salek, "Debiasing social wisdom," in *KDD*, Chicago, IL, August 2013, pp. 500–508.

[17] E. Kamar, A. Kapoo, and E. Horvitz, "Identifying and Accounting for Task-Dependent Bias in Crowdsourcing," in *HCOMP*, San Diego, USA, November 2015.

[18] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*. John Wiley & Sons, 2015.

[19] "CrowdFlower." [Online]. Available: http://crowdflower.com/

[20] R. W. Ouyang, M. Srivastava, A. Toniolo, and T. J. Norman, "Truth discovery in crowdsourced detection of spatial events," in *CIKM*, Shanghai, China, November 2014, pp. 461–470.

[21] J. Whitehill, T.-f. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *NIPS*, December 2009.

[22] B. Mozafari, P. Sarkar, M. Franklin, M. Jordan, and S. Madden, "Scaling up crowd-sourcing to very large datasets: a case for active learning," *Proceedings of the VLDB Endowment*, vol. 8, no. 2, pp. 125–136, 2014.

[23] B. Faltings, P. Pu, B. D. Tran, and R. Jurca, "Incentives to Counter Bias in Human Computation," in *HCOMP*, Pittsburgh, USA, 2014, pp. 59–66.