Proceedings, The Fourth AAAI Conference on
Human Computation and Crowdsourcing

# State Detection Using Adaptive Human Sensor Sampling

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

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

### Abstract

With the massive prevalence of smartphones, mobile social sensing systems in which humans acting as social sensors respond to geo-located crowdsourcing tasks, became extremely popular. Such systems can provide significant benefits particularly during crisis management and emergency situations. However, not only querying users can be extremely costly but also human sensors are mobile, subjective and their response delays can highly vary. In this paper we develop a social sensing system that performs sampling on mobile social sensors to achieve accurate and real-time detection of the state of emergency events. Our contributions are two-fold: (i) our approach can capture well emergencies even in large geographical regions, and (ii) our sampling approach considers the individual characteristics of the social sensors to maximize the probability of receiving accurate responses in a timely manner. We provide comprehensive experiments that indicate that our approach accurately identifies critical real-world events, has low overhead and reduces the classification error up to 90% compared to traditional approaches.

## Introduction

The recent massive prevalence of smartphone devices has established "Social Sensing" as an integral tool for authorities to identify and supervise real-world events in a cost-effective manner. Social Sensing refers to the process of soliciting input from ubiquitous human users who perform crowdsourcing tasks which vary from simple observations to complex tasks that require human intelligence. Extracting information from human sensors can provide great insights to authorities regarding ongoing events, often complementing information received from the static sensor infrastructures (Gu et al. 2014). Several social sensing applications have recently emerged to monitor emergency events in the real world, such as the Disaster Reporter in the FEMA application[1], the Ushahidi application for crisis events[2], and the Waze[3] application for traffic monitoring.



[1]https://www.fema.gov/mobile-app/

[2]http://www.ushahidi.com/

[3]http://www.waze.com/

The importance of social sensing in emergency scenarios has been emphasized in various scenarios such as the post-election violence events that took place in Kenya in 2008. These events created a need to human users to report the places where violent acts took place. This led to the development of the Ushahidi app that collected eyewitness reports through crowdsourcing and placed them on a map. Obviously, even though such information is critical, it cannot be captured by traditional sensor infrastructures. Similar needs are found in many scenarios including terrorist attacks, earthquakes, etc., and the developed maps can be used both by citizens to avoid locations under emergency and by authorities that analyze them to react in a timely manner (*e.g.*, by sending rescue teams or evacuating areas).

Motivated by the above scenario, the problem that we address in this paper is how to use human social sensors to detect the extent and severity of a major emergency event as it unfolds. The problem is challenging as: (1) Asking feedback from every individual human sensor is prohibited due to the volume and costs of crowdsourcing (*i.e.*, rewards for the participating human sensors, communication costs, etc). (2) Querying only a small set of users may not suffice to identify effectively all affected sub-regions is large geographical areas. (3) Human sensors have different perspectives, objectiveness and response delays as their answers depend not only on the human factor but also on their geographical location (in the case of location-based events). Thus, a key aspect is to identify human sensors who will provide useful, timely and relevant information for authorities.

This leads to the following question: Is it possible to use samples from human social sensors to effectively determine the state of the emergency event? Sampling is a well-known method that has been used in a variety of distributed system settings. For instance it has been employed to approximate aggregates in sensor networks (Lin et al. 2008), and to reduce produced social sensing data (Qi et al. 2013). Sampling can reduce the size of the problem by determining a subset of the data to be processed. However, the sample needs to be decided with respect to the available input data and the selection criteria to provide accurate results. Thus, defining the selection criteria is a fundamental task.

In this work we assume emergency scenarios that cover a large geographical area such as a city or a country. Our approach partitions the geographical area into a set of non-overlapping regions and performs systematic sampling within the regions from social sensors that observe the situation and provide feedback from their respective locations. We develop a sampling plan that considers both human factors (*e.g.*, user subjectiveness) as well as the users' geographical location to maximize both the spatial coverage and the knowledge extracted from the samples.

The research in the area involving the problem of sampling is very rich (Cochran 1953; Kong 2008). Stratified sampling (Chaudhuri, Das, and Narasayya 2007) is the closest approach to our scenario. It groups members of the population into homogeneous subgroups and performs random sampling from each subgroup. However, stratified sampling does not consider the individual user characteristics and might select inappropriate users (*e.g.*, spammers). Similarly, sampling approaches that consider traditional sensors instead of social sensors, such as region sampling (Lin et al. 2008) are limited since they do not take human factors into account. Sampling approaches have also been proposed recently to reduce the data produced from the social sensors (Qi et al. 2013; Thejaswini, Rajalakshmi, and Desai 2015). Our work is orthogonal to such approaches since we aim to select a sample out of the available social sensors. Finally, several approaches have also been proposed to determine the users that should respond to a crowdsourcing task (Roy et al. 2015; Ho and Vaughan 2012). However, they either focus on human factors (Boutsis and Kalogeraki 2014; Karger, Oh, and Shah 2011) without considering the spatial dimension of the users or they take user locations into account without considering human factors (Zhang et al. 2014) and typically assign tasks to nearby workers (Kazemi, Shahabi, and Chen 2013).

In this work we propose EViDENCE (EVent DEtectioN using soCial sEnsors), a social sensing system that aims to exploit the collective intelligence of the human crowd to identify the state of a major emergency event. We summarize our contributions below:

- We present EViDENCE, our system that determines the affected regions and their severity during major emergencies. EViDENCE partitions the geographical area into non-overlapping regions and performs sampling to determine the severity of a given event in each region.
- We formulate our sampling problem in order to consider user location, subjectiveness and response times. We prove that the problem is NP-hard and we develop a novel algorithm to solve it in polynomial time. We also show that our algorithm selects users that minimize the error when computing the state of the event.
- We perform extensive experiments using two real world emergency events: (a) the "Winter Snow Thor" snowstorm that took place in the USA on March 5, 2015, and (b) the "German Floods" that occurred in Germany in the period May 1 - July 31, 2013. We validate the benefit of exploiting human social sensors and demonstrate the performance and efficiency of EViDENCE using both a multiple-choice classification of weather conditions, and a numerical value response set where users estimate the precipitation in mm using numerical values.

## System Model

We now formalize our problem. We focus on major emergency events such as natural disasters, terrorist attacks and air pollution which constitute a key challenge in modern societies. Such emergency events have the following properties: (i) they cover large spatial regions, (ii) they are experienced by large populations and, (iii) although they occur sporadically, they require fast and accurate tackling.

An event $e$ in our system is represented by its state, defined as the distribution of a data value $x \in \mathcal{R}$ in a spatial region $\Lambda$. This distribution reveals both the *existence of the event*, when $x$ exceeds the normal values, as well as the *intensity of the event*, determined by the distribution of the values of $x$. We denote the state of the event $e$ for each geographical region $\lambda \in \Lambda$ as $state_e^\lambda$; this is computed by aggregating the social sensors' feedback in location ($lat_i, lon_i \in \lambda$). The distribution of the values $state_e^\lambda, \forall \lambda \in \Lambda$ illustrates the distribution of the target event in the evaluated region.

To receive social sensing tasks, human sensors register with the EViDENCE server, thus, the pool of social sensors changes dynamically. We denote each social sensor as $s_i \in S$ and we associate $s_i$ with the following tuple: $\langle lat_i$, $long_i$, $bias_i$, $previous\_tasks_i[]\rangle$, where $lat_i, long_i$ represent the current location (latitude, longitude) of the social sensor. Since we deal with tasks that assume subjective responses, the variation of the responses depends on the human factor, since users make specific types of errors for a specific type of task. Hence, we capture the *user bias*, that we will also refer simply as bias for the rest of the paper, that represents the expected percentage difference of each user responses for the emergency events compared to the real (*i.e.*, objective) values. We denote this metric as $bias_i$ and we update it dynamically based on the user responses as explained in the following section. Finally, we maintain statistics related to the completed tasks in $previous\_tasks_i[]$.

The EViDENCE system issues tasks to identify the state and intensity of an event that occurs. Tasks vary in complexity and users can get compensated for the responses they provide (*e.g.,* through a monetary reward). Each task $t_j \in T$ is associated with the following set of attributes: $\langle description_j$, $reward_j$, $area_j$, $amount_j$, $timewindow_j\rangle$, where $description_j$ describes the task that should be performed by the user (*e.g.*, *"Provide an estimate for the precipitation of rain in mm in your location"*), $reward_j$ refers to the monetary or other form of compensation received by the user that processed the task, which is typically predetermined by the task requester. The $area_j$ defines the geographical area that the task refers to in the form of a polygon, while $amount_j$ represents the maximum amount of human sensors requested to process the task. Every $t_j$ is also characterized by a time interval $timewindow_j$ defined by the task requester; this is the time within which the task responses should be returned.

For each task $t_j$, we seek to identify a subset of the social sensors, denoted as sampling set $Set_{gj}$, $g \subset S$. Each sensor $s_i \in Set_{gj}$ will be asked to process task $t_j$ and provide a response, denoted as $resp_{ij}^{lat,lon}$, coupled with her current location ($lat$,$lon$) so that the system can compute the state of the event $state_e^\lambda$ at this location. We consider as valid responses the ones received within the task's $timewindow_j$. User responses are quantitative and heavily depend on user perspective; that is, typically humans can confidently respond whether it is raining but they will be subjective when asked to estimate the precipitation of rain (Kerman et al. 2009). Thus, estimating the state of an event is considerably more difficult than traditional approaches that provide binary choices (Gu et al. 2014; Hu et al. 2015) or multiple choices from a predefined list of answers (Boutsis and Kalogeraki 2014; Cao et al. 2012), which are known in advance.

We denote as $error_e^\lambda$, the relative error of the event state $state_e^\lambda$ that we estimate using the sensors' sample, compared to the real state value in that geographical region, denoted as $r_e^\lambda$. We focus on the relative instead of the absolute error, as the former is usually a fairer measure, defined as:

$$error_e^\lambda = \min(\frac{|state_e^\lambda - r_e^\lambda|}{r_e^\lambda}, 1) \quad (1)$$

We formulate the error metric based on the difference of the aggregated samples compared to the real value, rather than the value we would receive from all users. This is because EViDENCE aims to maximize the accuracy and human users do not always provide accurate responses (*e.g.*, spammers). Finally, we bound the error within $[0, 1]$.

**Problem Description.** Our problem is defined as follows: *Assume a set of human sensors $S$ in location $\Lambda$. Let sensor $s_i$ in location $lat_i, lon_i \in \Lambda$ be capable of providing response $resp_{ij}^{lat,lon}$ when queried for task $t_j$. The problem we address is how to identify a sample from a subset of the available set of sensors ($Set_{gj}$, $g \subset S$) so that the total error $\sum_{\lambda \in \Lambda} error_e^\lambda$ is minimized.*

## The EViDENCE Approach

In this section we first give an overview of our sampling approach, then we present our method for selecting the samples from the available social sensors per region, and finally we illustrate how we evaluate the received samples to estimate the state of the event.

### EViDENCE Overview

The problem of efficient event detection comes up in several application domains, where statistical, probabilistic, machine learning, or composite techniques can be applied to detect events (Kerman et al. 2009). Several techniques have been proposed in the literature to detect events including particle filtering (Sakaki, Okazaki, and Matsuo 2010; Boutsis and Kalogeraki 2016) and kernel density estimators (Subramaniam, Kalogeraki, and Palpanas 2006). However, existing approaches have two important limitations: (i) they do not focus on human social sensors and, thus, they do not take into account the characteristics of individual human sensors, (ii) the majority of the existing works focus on detecting the location of an event, while we focus on determining not only all locations but also the intensity of the event for all affected regions.

One major challenge to address is the limited amount of social sensors that we can query due to budget constraints, communication costs and potential user denial to respond frequently. To overcome this, we propose a sampling approach, based on previous work on sensor networks (Lin et al. 2008; Willett, Martin, and Nowak 2004), where sampling has been widely used to extract important information with a reduced size of data. However, our sampling plan considers both human characteristics (Roy et al. 2015; Ho and Vaughan 2012) and the location of the social sensors (Zhang et al. 2014) as these factors can highly improve the obtained knowledge for the target event.

Hence, we develop a novel adaptive sampling approach to detect the state of the target event. Since the locations where the emergency takes place is unknown, we systematically investigate the entire and, potentially large, geographical area to detect all locations where the event occurs. We achieve that using an iterative approach as follows:

- ***Step 1:*** We segment the spatial region into $P$ non-overlapping regions ($\lambda = 1, \ldots, P$), whose amount and size can be tuned based on application-specific criteria.
- ***Step 2:*** We perform sampling in each region $\lambda$, as explained in the next section, considering both spatial and human factors to detect the affected regions.
- ***Step 3:*** For each region $\lambda$ which is determined to be under emergency, we further segment it into smaller non-overlapping regions. In our experiments we divide each region to four equal regions. However, different types of segmentation can be incorporated depending on the application scenario.
- ***Step 4:*** We iterate through steps 2 and 3 until the size of the regions becomes small enough to represent its state. That way we can use the available resources to focus on areas under emergency.

### Step 1: Segmenting Spatial Space

In the first step, we segment the spatial space into $P$ non-overlapping regions, similar to previous works that consider event detection in sensor network environments (Lin et al. 2008). This allows us to concentrate on the individual regions during the sampling process, to more efficiently determine the state of the event. In the following steps we explain how the regions are further divided to provide an accurate representation of the state of the event for each location.

### Step 2: Sampling within Regions

The objective of our sampling algorithm is to determine the set of social sensors $Set_{gj}$ that will be asked to provide samples for task $t_j$ for each region (although humans might refuse to respond), so that the sampling error is minimized. We achieve that by selecting the set of sensors that (i) will cover well the entire spatial region, since the potential locations of the event is unknown beforehand, and (ii) are able to provide objective responses in a timely manner. Thus, our problem can be defined as an optimization problem that:

- Maximizes both (1) the spatial coverage of the region, denoted as $Cov_{gj}$, and (2) the probability of providing an unbiased $response_j$, denoted as $P(Obj_{gj})$.
- Sets a constraint for the second objective to ensure that the received responses will fulfill at least a predefined level of objectiveness, denoted as $B$, as: $P(Obj_{gj}) > B$.
- Ensures that the selected individuals will execute the task $t_j$ within a $timewindow_j$ (this is expressed by setting a lower bound threshold $\tau$ on the probability of providing a timely response: $P(exec_{ij}) > \tau, \forall s_i \in Set_{gj}$).

The problem is that the number of combinations $Set_{gj}$ can be very large (there are $\binom{|S|}{amount_j}$ possible combinations to select $amount_j$ sensors from the set of available sensors $S$). Solving the problem naively by examining all feasible solutions is computationally infeasible for an online system for emergencies that requires fast responses.

In the following we present our approach for estimating the spatial coverage and computing the bias for each set of users, we then prove that the defined optimization problem is NP-hard and finally we present an efficient sampling algorithm that solves the optimization problem.

**Estimating Spatial Coverage.** The challenge is, that, there is no necessarily a priori knowledge regarding the location where an event might occur. In order to cover the entire region, and given that its size can be large, our goal is to retrieve samples evenly from all locations within the region and then try to focus on the sub-regions where the event occurs. Thus, we develop a grid structure, similar to (Van Dyke Parunak and Brueckner 2001), by dividing the region $\lambda$ into a number of smaller locations $\lambda_1, \lambda_2, \ldots, \lambda_k \in \lambda$, and we exploit the Entropy (Shannon 2001) to evaluate the distribution of the amount of selected sensors in each location.

We compute the entropy for the group of sensors $Set_{gj}$ as follows: for each location $\lambda_1, \lambda_2, \ldots, \lambda_k \in \lambda$ we compute the probability $Pr(\lambda_k) = \frac{sel_{\lambda_k}}{total}$, as the number of the sensors from $Set_{gj}$ located in $\lambda_k$, denoted as $sel_{\lambda_k}$, over the amount of sensors in $Set_{gj}$ in the entire region ($total$). Thus, we denote the entropy H for the set of sensors $Set_{gj}$ as:

$$H(Set_{gj}) = -\sum_{\lambda_k \in \lambda} Pr(\lambda_k)\log(Pr(\lambda_k)) \tag{2}$$

Entropy provides a measure of the information contained in that distribution. Intuitively, the Entropy increases when the amount of the evaluated items becomes more similar, thus, selecting the same amount of sensors from every location for $Set_{gj}$ will maximize $H(Set_{gj})$. Our goal is to choose sensors that maximize the spatial coverage by tuning the number of sensors selected for each location $\lambda_1, \lambda_2, ..., \lambda_k$. The maximum entropy for a given set of locations $\lambda_k$ can be computed by the logarithm of the number of locations as $MaxH(\lambda) = \log(|\lambda_k|)$, where $|\lambda_k|$ denotes the amount of locations $\lambda_k$ in region $\lambda$. Thus, we compute the spatial coverage for which we acquire information as:

$$Cov_{gj} = \frac{H(Set_{gj})}{MaxH(\lambda_k)} \tag{3}$$

**Lemma 1** *The function $Cov_{gj}$ for a specific amount of selected sensors $amount_j$ increases, if we replace a sensor from location $\lambda_k$ that contains $x$ sensors for a sensor from a location that contains $y$ sensors, if $y < x$.*

*Proof.* Since the total $amount_j$ of sensors remains the same and thus the maximum entropy $MaxH(\lambda_k)$ remains the same, it is sufficient to show that $H(Set_{gj})$ increases when we perform that change. Since all the terms from other locations in $-\sum_{\lambda_k \in \lambda} Pr(\lambda_k)\log(Pr(\lambda_k))$ remain the same we need to show that altering the number of sensors in these locations increases the entropy, and so: $-\frac{x}{k}\log\left(\frac{x}{k}\right) - \frac{y}{k}\log\left(\frac{y}{k}\right) \leq -\frac{x-1}{k}\log\left(\frac{x-1}{k}\right) - \frac{y+1}{k}\log\left(\frac{y+1}{k}\right)$. This can be expressed as: $\log\left(\frac{(x/k)^{(x/k)} * (y/k)^{(y/k)}}{((x-1)/k)^{((x-1)/k)} * ((y+1)/k)^{((y+1)/k)}}\right) \geq 0$ and using the properties of the k-th root we need to show that $\frac{(x)^{(x/k)} * (y)^{(y/k)}}{(x-1)^{((x-1)/k)} * (y+1)^{((y+1)/k)}} \geq 1$, which is equal to $x * (\frac{x}{x-1})^{x-1} \geq (y+1) * (\frac{(y+1)}{y})^{y}$. This is always true since we assume that $y < x$.

**Estimating User Bias.** Our second goal is to select sensors that are able to provide objective responses since users have bias when responding to crowdsourcing tasks (Ouyang et al. 2015). Several approaches have been proposed recently to eliminate user bias from the respective responses but these approaches have several limitations: (i) they focus on the simplest case of using only binary responses (Zhuang et al. 2015; Zhuang and Young 2015) instead of numerical responses, (ii) they use active learning approaches (Ouyang et al. 2015) requiring too many iterations to converge, while they do not consider the fact that in crowdsourcing users typically answer sparsely, or (iii) they use hybrid models to resolve the issue of the large amount of iterations by using a population-wide representation when the user has not provided enough responses (Kamar, Kapoo, and Horvitz 2015).

In EViDENCE we update the user $bias_i$ adaptively, whenever a human sensor has been selected from the sampling approach and has successfully performed the task. As mentioned above, $bias_i$ refers to the percentage difference that we expect to retrieve from a sensor $s_i$, compared to the real (*i.e.*, objective) value when providing an answer for $t_j$. We infer the user bias using an online Expectation Maximization algorithm that alternates between computing an expectation of the parameters' values, by taking into account the observations and the current estimates, and updates the values by maximizing this expectation. We use a variable $b_i$ to estimate the user bias as follows. We update $b_i$ whenever the system retrieves an answer from $s_i$ as follows: For each response $resp_{ij}^{lat,lon}$ retrieved for $t_j$, we evaluate the percentage bias injected in the response by the user, denoted as $\beta(resp_{ij}^{lat,lon})$. To compute $\beta(resp_{ij}^{lat,lon})$, since the real value of the location is unknown, we retrieve all responses within a spatial distance $r$ from the location of the sensor that provided $resp_{ij}^{lat,lon}$ and we compare the percentage difference of the response compared to the median $\mu_{1/2}^{lat,lon}$, to exclude outliers:

$$\beta(resp_{ij}^{lat,lon}) = \frac{|resp_{ij}^{lat,lon} - \mu_{1/2}^{lat,lon}|}{\mu_{1/2}^{lat,lon}} \tag{4}$$

Then, we update $b_i$ for each participant $s_i$ using a variable $\gamma_i$, which is assigned with a small value as:

$$b_i = (1-\gamma_i)b_i + \gamma_i\big(\beta(resp_{ij}^{lat,lon})\big) \tag{5}$$

Thus, $b_i$ will converge to the expected user bias. However, in order to be able to consider the variations in terms of subjectiveness, we compute $bias_i$ by adding to $b_i$ the standard deviation $\sigma^2$ of the percentage bias injected in the responses of the specific user $\beta(resp_{ij}^{lat,lon})$:

$$bias_i = b_i + \sigma^2(\beta(resp_{ij}^{lat,lon}), \forall j) \tag{6}$$

This allow us to bound user subjectiveness as the user will typically respond more objectively than her $bias_i$:

$$|(resp_{ij}^{lat,lon} - r_e^{lat,lon})/r_e^{lat,lon}| \leq bias_i \tag{7}$$

Finally, we compute the probability that the sampling set $Set_{gj}$ will provide an objective response, denoted as $(P(Obj_{gj}))$, as the product of the objectiveness of the selected sensors, where objectiveness is defined as $1 - bias_i$, that we bound in the $[0,1]$ region:

$$P(Obj_{gj}) = \prod_{s_i \in Set_{gj}} (1 - \min(bias_i, 1)) \tag{8}$$

**Lemma 2** *The objective $P(Obj_{gj})$ is a monotonic and increasing function of* $1 - \min(bias_i, 1)$.

*Proof.* The function $P(Obj_{gj})$ depends on the product of the worst case probability to provide an objective response for each individual sensor $(1-\min(bias_i, 1))$ which is bounded in the $[0,1]$ range. Thus, selecting a sensor $k$ with individual probability $1 - \min(bias_k, 1) \in g$, instead of a sensor $k'$ with individual probability $1 - \min(bias_{k'}, 1) \in g'$, where $1 - \min(bias_k, 1) \geq 1 - \min(bias_{k'}, 1)$ and $g \setminus \{k\} = g' \setminus \{k'\}$, results in $P(Obj_{gj}) \geq P(Obj_{g'j})$. ■

**Sampling with EViDENCE.** Next, we prove that the problem of sampling within regions is difficult as it can be reduced from the Knapsack problem and we present an efficient polynomial algorithm that is able to maximize both objectives to select the best set of users to perform sampling.

**Lemma 3** *Our sampling problem is NP-hard since it can be reduced from the Knapsack problem.*

*Proof.* The Knapsack problem states, that, given a set of objects with utility $z_i$ and cost $c_i$, and a bound $C$, the goal is to find a set of $x$ objects that maximizes $\sum_{i=1}^{x} z_i$, while $\sum_{i=1}^{x} c_i \leq C$. Assuming an instance of the Knapsack problem we can create an instance of our problem. Let us consider a simplified version of our problem, where we aim at optimizing the objective $Cov_{gj}$, subject to the constraint $P(Obj_{gj}) > B$. We create an instance of our problem where for each object we have a $s_i$ whose utility depends on the sensor's spatial location with a respective objectiveness $1 - \min(bias_i, 1)$. Compared to the original problem, our objective function $Cov_{gj}$ is more complex than the summation of values of the selected items. Our defined constraint $P(Obj_{gj}) = \prod_{s_i \in Set_{gj}} (1 - \min(bias_i, 1)) > B$ is equivalent to $\sum_{s_i \in Set_{gj}} (-\log(1 - \min(bias_i, 1))) < -\log(B)$ and so the cost for each $s_i$ is $-\log(1 - \min(bias_i, 1))$ and the bound $C = -\log(B)$. Thus, we reduce the Knapsack problem to our problem, and so, our problem is NP-hard. ■

To solve the NP-hard problem, we propose a polynomial algorithm to identify the sensors that maximize both objectives and fulfill the constraints. To achieve that we determine the solution that maximizes $P(Obj_{gj})$ and we traverse through the set of feasible solutions, to determine the solution that maximizes $Cov_{gj}$ for the defined constraints.

Our algorithm is summarized in Algorithm 1. To determine the $Set_{gj}$ of sensors that will process $t_j$ we first extract the list of available sensors $\mathcal{L}$ that reside in the area of the task $(boundingbox_j)$ and we filter out the sensors with low probability to provide an answer within $timewindow_j$ $(P(exec_{ij}) \leq \tau)$. This can be estimated by the Cumulative Distribution Function of the Power Law distribution based on the sensor's profile (we follow the observation that in social sensing systems the execution times of the sensors follow a Power Law distribution (Ipeirotis 2010)). Next, we sort the list $\mathcal{L}$, based on the sensor objectiveness and we add the top $amount_j$ sensors from the sorted list $\mathcal{L}$. This ensures that we will retrieve a feasible solution, if it exists, that maximizes $P(Obj_{gj})$, due to Lemma 2 and the sorting.

Then, we continue iterating through the sorted list $\mathcal{L}$ to evaluate all sensors that have not been selected in $Set_{gj}$: $\mathcal{L} \setminus Set_{gj}$. For each $s_i \in \mathcal{L} \setminus Set_{gj}$, we investigate if we can substitute the sensor for another sensor in $Set_{gj}$ to provide a feasible solution that increases the spatial coverage. We only consider a substitution with the sensor determined from the function $senToRepl(Set_{gj})$. This function returns the most subjective sensor in $Set_{gj}$ from the location with the maximum amount of sensors. We choose this approach that exploits Lemma 1 since computing the entropy has a high complexity. If such a substitution increases the spatial coverage and provides a feasible solution we accept it. We stop the iterations when the spatial coverage $Cov_{gj}$ has converged to one, or when the evaluated sensor produces a group reliability $(P(Obj'_{gj}))$ which is less than the predefined bound. In both cases, due to the sorting we would not be able to find a feasible solution that increases one of the objectives.

**Algorithm 1** Sampling Algorithm

$\mathcal{L}$= available sensors in $boundingbox_j$ with $P(exec_{ij}) > \tau$
Sort($\mathcal{L}$) by $(bias_i)$ in ascending order
**for** $(s_i \in \mathcal{L})$ **do**
  **if** $(Set_{gj}.amount < amount_j)$ **then**
    $Set_{gj} = Set_{gj} \cup \{s_i\}$
  **else**
    $Set'_{gj} = (Set_{gj} \cup \{s_i\}) \setminus \{senToRepl(Set_{gj})\}$
    **if** $(Cov'_{gj} > Cov_{gj}$ && $P(Obj'_{gj}) > B))$ **then**
      $Set_{gj} = Set'_{gj}$;

**Lemma 4** *In every iteration, the sampling process produces* *a solution that improves $Cov_{gj}$ and ensures that no other solution can improve $P(Obj_{gj})$ for the sensors considered.*

*Proof.* For each $s_i$ that we evaluate, the sensor's objectiveness is worse or equal to the ones in $Set_{gj}$ due to the sorting. However, we choose to substitute $s_i$ with another sensor to maximize the spatial coverage, while still providing a solution in the feasible region. Thus, we ensure that for the set of sensors $Set_{gj} \cup \{s_i\}$ there is no other feasible solution with $amount_j$ sensors and highest $Cov_{gj}$.■

**Minimizing sampling error.** We argue that EViDENCE effectively provides an approach to minimize the sampling error for each individual region $\lambda$. Assume that objective users exist in each location $\lambda_1, \lambda_2, ..., \lambda_k \in \lambda$. EViDENCE will maximize the spatial coverage and will select the most objective users in each of these locations. Due to equations (1), (7), the expectation for the $error_e^{\lambda}$ will be minimized for the available set of users. On the other hand, in the case that users exist in a subset of these locations but all of them are too subjective to be considered, due to the bound ($P(Obj_{gj}) > B$), then we expect that these users would only inject noise to the aggregation of the result for the $state_e^{\lambda}$. However, EViDENCE is flexible enough to consider these users as well by setting $B = 0$.

**Worst-Case Complexity.** Assuming $n$ sensors, we first iterate through the list of sensors to exclude the ones who will fail to provide an answer within $timewindow_j$ or belong outside $boundingbox_j$, that costs $\mathcal{O}(n)$. Then we sort the sensors which costs $\mathcal{O}(nlogn)$. In order to compute the entropy, we iterate through the selected group of sensors that costs $\mathcal{O}(amount_j)$ and compute the frequency of each location. However, we can do this only once, and then update the respective counters when a sensor is added or removed. Finally, we iterate through the list of available sensors, which costs $\mathcal{O}(n)$, and each of the operations to decide whether to accept the sensor costs $\mathcal{O}(1)$. Thus the worst case complexity of EViDENCE is $\mathcal{O}(n) + \mathcal{O}(nlogn) + \mathcal{O}(amount_j) + \mathcal{O}(n) = \mathcal{O}(nlogn)$.

## Step 3: Evaluating Sensor Responses

In this section we present how EViDENCE evaluates the samples retrieved from the sensors, selected by our sampling plan and continues with the iterations to divide the regions.

**Eliminating Outliers.** Although we have selected objective users to sample from, we evaluate their responses and we eliminate the user responses which are far off their neighbors as outliers. To achieve that we use the Gaussian distribution function for the responses provided by other sensors, within a spatial distance $r$ and we compare each response $resp_{ij}^{lat,lon}$ provided by $s_i$ compared to the mean. Assuming that there exist at least $k$ responses by sensors within $r$, we set a threshold $\rho$ for bounding the probability of accepting responses to a specific level and we accept $resp_{ij}^{lat,lon}$, only if $e^{-(resp_{ij}-\mu)^2/(2\sigma^2)} > \rho$. Thus, we only eliminate responses when their values are distant from the rest of the sensors' responses in that location, to ensure that this would not result on failing to capture an emergency event.

**Segmenting Regions.** In order to decide whether a region should be segmented into smaller regions we use the most extreme response, that has not been eliminated as an outlier, among the ones provided by the sensors in all iterations. Although we assume that this is the highest reported value our formulation can easily be transformed to support applications where the lowest value should be used (*i.e.*, food shortage scenario). Thus, if the most extreme response $\max(resp_{ij}^{lat,lon})$ sets the region under emergency we further divide it into smaller regions to perform sampling.

This approach enables us to further investigate the possibility that an emergency event occurs and accurately determine all locations involved in the event. Assuming that an outlier falsely reports a high value and there are several sensors nearby that truthfully report the opposite, the false report would not be considered. However, if there are no nearby sensors the location will be considered as being in an emergency, which will trigger in retrieving more data in that region as the iterations continue. This enables us to either determine that the report was false or to accurately define the sub-locations affected by the event.

## Step 4: Determining the State of the Event

After the defined amount of iterations of EViDENCE has been reached we can compute the state of the event $state_e^{(\lambda)}$, for every location $\lambda$. Hence, we extract the set of accepted responses in each region and we compute the median to determine the most probable answer, so as to exclude outliers:

$$state_e^{(\lambda)} = \mu_{1/2}(resp_{ij}^{lat,lon}), \forall resp_{ij}^{lat,lon} \in \lambda \quad (9)$$

The benefit is that the regions are constantly divided into smaller areas, we fine-tune the sizes of these areas and determine their data values. This way we can better capture an event even when the data values change (*e.g.*, water levels).

## Tracking events over time

Tracking events over time can be achieved easily by: (i) updating the values in the defined affected locations, and (ii) updating the boundaries of the affected spatial region when nearby locations get affected. Given that we have identified a closed shaped spatial area $A$, where all locations $\lambda$ within that area are affected by an event $e$, captured from the $state_e^{(\lambda)}$, we can track the event as follows: We develop a new spatial area $A'$ where its boundaries are extended with a distance $v$, which depends on the possible spatial evolution of the event: $dist(p, p') < v$, $\forall p \in A$ and $\forall p' \in A'$. For this area ($A'$) we perform sampling and re-evaluate the state using the data retrieved within a time window, to determine the updated borders and state of the evolving event.

## Real-world Applicability

EViDENCE has already been employed in our real-world urban monitoring system (Panagiotou et al. 2016). It is triggered when an emergency event occurs to obtain additional information by querying citizens to extract real-time feedback about the event. EViDENCE can easily be incorporated in existing social sensing systems, such as Waze and Ushahidi that typically rely on reports provided voluntarily by the users. This would allow them to extract information from the ubiquitous users dynamically to determine the state of events in large geographical regions in real-time.

# Experiments

We run experiments using a real-world dataset (obtained from WUnderground)[4] that represents the weather conditions for two major events: (i) the "Winter Storm Thor" that occurred in the South, Midwest and East USA, focusing on March 5, 2015, which was one of the most severe days, and (ii) the "2013 German Floods" during the period of major European floods between May 1 - July 31, 2013.

## Winter Storm Thor

The first scenario enables us to understand how humans, acting as social sensors, contribute data during emergency events, and also evaluate the accuracy of the data they report. In this scenario we extracted human responses from the mPING app[5] that includes approximately 5,000 weather reports from users provided on March 5, 2015. Users report weather conditions (*e.g.*, rain, fog, drizzle) by selecting among multiple choices, but are not able to provide a numerical quantification about the amount of rain. In this scenario, human reports are obtained from the mPING app and *there is no way to control the sampling process*, as there is no user identification in the reporting app.

Figure 1a illustrates in a heatmap the categorization of the weather conditions derived from WUnderground, that uses a binary categorization for rain and snow in every location. The value of one (grey) represents rainy conditions, while the value of two (black) represents snowy or both snowy and rainy conditions. In figure 1b we present the respective heatmap for the reports provided from the social sensors as derived from mPING. Since mPING provides a more fine grained categorization, to be consistent with figure 1a, we group user reports that refer to rain (*e.g.*, rain, drizzle, etc.) with the value of one, and reports that refer to snow (*e.g.*, snow, ice pellets, etc.) with the value of two. As can be seen, the social sensors provide a quite accurate representation of the extreme weather event as the majority of the areas are categorized correctly, and especially the overland areas.

**Accuracy.** Figure 1c presents the accuracy of the social sensors when categorizing weather conditions. We classify each area as **T**ruly or **F**alsely considered as being under emergency (**P**ositive) or not (**N**egative). We assume that a location is under emergency when there are snowy conditions. We vary the number of total acquired reports (500 - 5000) and the maximum number of users per location (1-100) to capture the fact that users heavily report from a location when an event occurs. True Negative and False Positive represent areas that weather sensors have marked as not snowing. Social sensors report for up to 13.5% of these areas that it snows (FP). Such reports should be investigated by the authorities as they can be vital (*e.g.*, they may correspond to reports provided by humans in streets not covered by weather sensors or they may capture faulty sensors). True Positives and False Negatives represent areas that the weather sensors report snow. The social sensors fail to categorize all of them correctly, mainly because they are not able to cover all respective areas. Varying the maximum amount of reports per location has a small impact, since users can accurately answer binary questions. However, increasing the total amount of reports increases True Positives and reduces False Negatives, as they cover larger spatial area.

## German Floods

In our second scenario we deal with answers that contain numerical values and we aim to determine accurately the rain precipitation that indicates the state of the floods that took place in Germany. In order to develop real user profiles we exploit the findings of the CrowdFlower case studies reported in (Boutsis and Kalogeraki 2014; Boutsis, Kalogeraki, and Gunopulos 2016). As we obtained from the studies each user has a $bias_i$ in the range of [0.0,0.93], with the average value of 0.35. Hence, we assume that users respond with the same behavior, as in the case studies, but we also inject some random noise in their responses. The response delay of the users also depends on their behavior and ranges from 9 to 90 seconds, while the time window is set to 70 seconds; this enables us to model the behavior of providing delayed answers or rejecting tasks; in our experiments we assume a fixed $reward_j$ for all tasks. Finally, the sensors are assigned with a location within the spatial region but we bias some locations to include more users, as happens in real environments where humans are located densely in large cities.

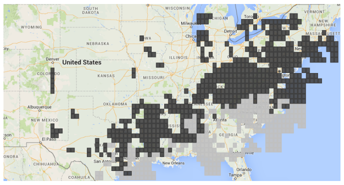
In the following experiments we have set the bound $B$ to $0.5^{amount_j}$ and the execution time threshold $\tau$ to 0.5. Each response is evaluated against the ones provided within 500 meters and we set the acceptance threshold $\rho$ to 0.9. Note, that, in this scenario we consider that the areas are under emergency when the precipitation is over 3mm; such values can be provided by a local authority. Finally, we state that EViDENCE iterates up to six times to sample from sensors.

**Classification Accuracy.** In figure 2a we present the accuracy of EViDENCE in terms of categorizing the spatial areas for being under an emergency for various numbers of total sensors and sample size. For a sensor size of 1000 sensors the percentage of False Negatives is high ranging among 7 and 8 percent; this is because the sensor size is small and thus we are not able to retrieve data from all the needed regions. On the other hand the False Positives are minimal. We state, however, that False Negatives are more critical than False Positives, when making decisions for helping regions under emergency. Selecting more samples per iteration increases the categorization accuracy only slightly. On the other hand, when we increase the amount of total users in the region, EViDENCE reduces the False Negatives which are practically eliminated for selecting more than 100 samples per iteration. Keep in mind that increasing the total users only increases the amount of "good" sensors that we can sample from based on their location and objectiveness.

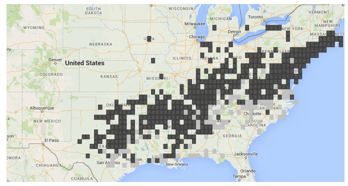
**State Accuracy.** Figure 2b presents the accuracy of the state value for all regions which are actually under emergency. We focus on the regions which are under emergency since mistakes in these regions (misclassification or underestimation of the situation) can lead to fatal results. Thus, we capture the average sampling error $error_e^\lambda$ for the affected areas using our Critical Error metric: $\frac{\sum_{\lambda \in TP}(error_e^\lambda) + \sum_{\lambda \in FN}(1)}{|TP|+|FN|} *$

[4] http://www.wunderground.com/weather/api/

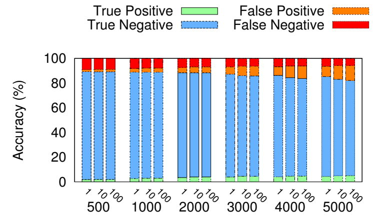
[5] https://mping.ou.edu/



(a) Weather Sensors

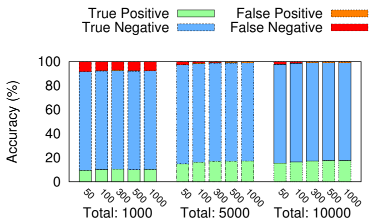


(b) Human Sensors

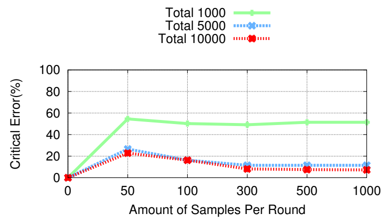



(c) Accuracy

Figure 1: HeatMap for Winter Storm Thor (Rain=Grey, Snow=Black) and Accuracy Evaluation for the Social Sensors




(a) Accuracy




(b) Critical Error

Figure 2: EViDENCE Evaluation for Varying Numbers of Samples

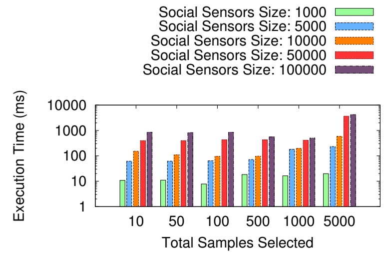



Figure 3: Execution Time for the Sample Selection

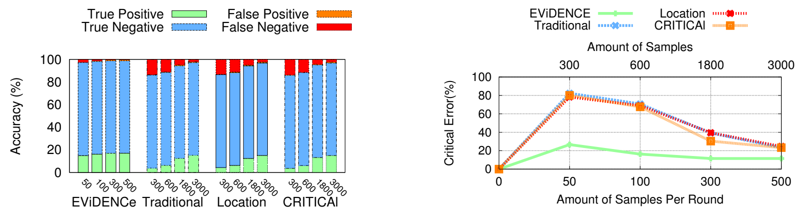
100 which represents the average of: (1) the sampling error for each correctly assigned region (TP), and (2) the error of one for all falsely evaluated regions (FN), multiplied with 100. Similar to figure 2a, we observe that the amount of total users in the social sensing system plays an important role at the accuracy of the results, compared to the sample size per round. Hence, we observe that for 1000 sensors in total the critical error to capture the state of the event ranges within 50-55% mainly due to the amount of false negatives. Respectively when the total sensors are increased to 10000 the critical error reduces to 8% for sampling from more than 300 sensors. Moreover, we report that if we consider only the true positives, the respective critical error was approximately 14% for 1000 sensors, and 7% for 10000 sensors, no matter how many samples we retrieve per round.

**Scalability.** In figure 3 we present the execution time of the sampling process on a single EViDENCE node, under various sensor sizes, for a defined location. As the figure shows, the execution time is slightly affected by the amount of samples, compared to the total sensor size. This happens since the complexity is highly affected by the sorting performed. The low complexity of EViDENCE is also depicted, since the execution time even for selecting 5000 samples from 100000 sensors is 4.5 seconds. However, as we have shown previously selecting 300 samples from a total set of 10000 is sufficient to provide an accurate prediction for large areas.

**Comparison.** Since there is no other technique that we know that exploits social sensing to detect emergency events, based on quantitative data, we compare EViDENCE with (i) the *Traditional* approach, where the users select tasks at will which is similar to Random Sampling, (ii) with *Location*, an approach that considers only the spatial coverage objective and (iii) *CRITICAl* (Boutsis and Kalogeraki 2014), a state-of-the-art approach that considers assigning tasks based on the reliability of the sensors and real-time constraints. However, we extend CRITICAl, which considered qualitative data to consider quantitative data as well. We perform these experiments with a sensor set size of 5000. EViDENCE iterates up to six times and samples from the available sensors. Thus, to make the comparison fair with the rest of the approaches, which are non-iterative, we select six times more samples, based on their strategy, to extract the same amount of samples with EViDENCE.

In figure 4a we compare the categorization accuracy of the approaches. As can be observed, EViDENCE has a superior performance when categorizing the regions, especially for the critical true positives and false negatives. EViDENCE performs within 15-17% for the true positives while the other approaches do not exceed 6.2% when they sample from 600 sensors. For the same amount of users the false negatives of these approaches is 11% while EViDENCE has only 1%. This is mainly because our iterative sampling approach eliminates the regions which are not under emergency and focuses on the important ones. However, as the amount of samples increases the difference is reduced. Respectively, figure 4b illustrates that EViDENCE has a great advantage on the critical error, mainly because the other approaches cannot compete with EViDENCE in the categorization of the regions. Although the critical error of EViDENCE ranges between 11-27% depending on the amount of sensors, the rest performed within 23-82%.

We also employed the iterative sampling approach to the Location and CRITICAl strategies, to compare how the individual objectives of EViDENCE perform. Figure 5 shows that the Location objective that aims to maximize the entropy has the worst performance on the False Negatives, since it selects too many sensors that produce noisy responses. On the other hand, CRITICAl selects high quality




(a) Accuracy (b) Critical Error

Figure 4: Comparison with state-of-the-art approaches

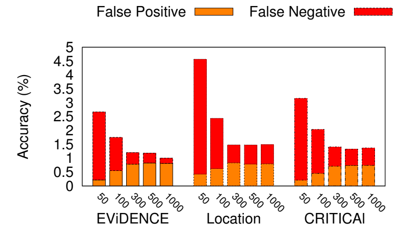



Figure 5: Accuracy evaluation using individual objectives

users, but it cannot capture the whole region. Thus, it manages to categorize less regions correctly. Nevertheless, that difference is reduced as the amount of samples increases, since inevitably such sensors will exist in all regions.

## Related Work

There has been a large body of work on sampling to reduce the size of the data in several application domains (Kong 2008). Sampling approaches that consider traditional sensor networks typically aim to reduce the energy consumption of the sensors (Lin et al. 2008) but they do not consider human factors which is fundamental in social sensing as humans responses can not be trusted. Other sampling approaches aim to reduce the data produced from the individual social sensors (Qi et al. 2013). On the other hand our goal is to reduce the amount of sensors that will provide the samples.

Recent works have identified that humans can act as social sensors and provide great benefit in ascertaining the correctness of the collected data (Wang et al. 2013) and several approaches have been proposed to determine the most suitable users for each task based on human factors (Roy et al. 2015; Ho and Vaughan 2012). Most of these works focus on assigning tasks based on user reliability (Karger, Oh, and Shah 2011), user "quality" (Khazankin et al. 2011), both user reliability and real-time requirements (Boutsis and Kalogeraki 2014) or to minimize the error subject to a budget constraint(Cao et al. 2012). Unlike EViDENCE, none of the these approaches deals with the problem of detecting the state of real-world emergency events in a wide geographical area and, thus, none of them considers the spatial dimension which is fundamental for emergency events.

On the other hand, task assignment approaches that take user locations into account also exist in the literature. A number of works have been proposed to maximize the spatial coverage in social sensing (Weinschrott et al. 2011; Zhang et al. 2014; Zhao, Li, and Ma 2014; Li, Li, and Wang 2015; Han, Zhang, and Luo 2014), but these approaches fail to consider important human factors. Another approach that considers human factors (Kazemi, Shahabi, and Chen 2013) aims to assign tasks to trustworthy users who are located near the task, but it does not focus on event detection and thus, it does not consider that the task assignment should cover a wide spatial area rather than a single location.

Detecting spatial phenomena has been widely studied in traditional sensor networks (Keally et al. 2014). However, exploiting social sensors has important differences as they: (1) provide subjective answers, (2) have different response criteria, (3) their location can change dynamically. Most of the sensor network works deal with static sensors (Wang and Cheng 2008), and they focus on determining the event boundaries (Subramaniam, Kalogeraki, and Palpanas 2006) rather than identifying the state and intensity of the event. In (Krause et al. 2008) they propose community sensing methods for sensing applications. However, they extract data from sensors instead of humans and the samples are selected based on the demand for each location while we extract samples from all "active" areas evenly, as emergency events may occur everywhere. Moreover, our problem is more complex as we maximize two objectives while their goal is to maximize a function that quantifies the expected information gain. Authors in (Sakaki, Okazaki, and Matsuo 2010) exploit Twitter as a geolocated sensor data source to detect real-time events. However, their approach has no influence on the sensor selection, and they focus on detecting the focal point of the event. Authors in (Wang et al. 2015) propose a scalable social sensing approach that exploits dependencies between observed variables to increase fact-finding accuracy. In contrast to our setting, they assume that random users will provide data voluntarily. In (Chu et al. 2011) they propose different social sensing strategies for disaster situations, but they do not focus on the user selection process or on detecting the state of the event. Authors in (Gu et al. 2014) study the data extrapolation in social sensing systems for disaster response, while in (Hu et al. 2015) they consider logical dependencies to minimize the network bandwidth in crowdsourcing environments, during emergencies. However, both approaches focus on binary tasks and they do not take the human factor into account. In our previous work (Boutsis and Kalogeraki 2016), we exploited particle filters to identify regions under emergency. However, this work did not consider the human characteristics that improve the results dramatically compared to random sampling.

## Conclusions

In this paper we presented our sampling approach that seeks to exploit human sensors to achieve robust, real-time detection of the state of emergency events. We draw the following conclusions from our experiments: (1) Users can contribute accurate and critical information during emergency scenarios, (2) EViDENCE, by taking into consideration user abilities and location, can reduce the classification error up to 90% during emergencies, compared to state-of-the-art approaches, and (3) we illustrated that EViDENCE can work robustly for wide areas with large amounts of users.

## 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

Boutsis, I., and Kalogeraki, V. 2014. On task assignment for real-time reliable crowdsourcing. In *ICDCS, Madrid, June 2014*, 1–10.

Boutsis, I., and Kalogeraki, V. 2016. Using human social sensors for robust event location detection. In *DCOSS, Washington, 2016*.

Boutsis, I.; Kalogeraki, V.; and Gunopulos, D. 2016. Reliable crowdsourced event detection in smart cities. In *SCOPE, Vienna, Austria, April 2016*.

Cao, C. C.; She, J.; Tong, Y.; and Chen, L. 2012. Whom to ask?: jury selection for decision making tasks on micro-blog services. *PVLDB* 5(11):1495–1506.

Chaudhuri, S.; Das, G.; and Narasayya, V. 2007. Optimized stratified sampling for approximate query processing. *ACM Transactions on Database Systems (TODS)* 32(2):9.

Chu, E. T.-H.; Chen, Y.; Liu, J.; and Zao, J. 2011. Strategies for crowdsourcing for disaster situation information. *WIT Transactions on the Built Environment* 119:257–269.

Cochran, W. G. 1953. Sampling techniques.

Gu, S.; Pan, C.; Liu, H.; Li, S.; Hu, S.; Su, L.; Wang, S.; Wang, D.; Amin, T.; Govindan, R.; Aggarwal, C.; Ganti, R.; Srivatsa, M.; Barnoy, A.; Terlecky, P.; and Abdelzaher, T. 2014. Data extrapolation in social sensing for disaster response. In *DCOSS, May 2014*.

Han, K.; Zhang, C.; and Luo, J. 2014. Bliss: Budget limited robust crowdsensing through online learning. In *SECON, Singapore, Malaysia, June 2014*.

Ho, C.-J., and Vaughan, J. W. 2012. Online task assignment in crowdsourcing markets. In *AAAI, Toronto, Canada, July 2012*.

Hu, S.; Li, S.; Yao, S.; Su, L.; Govindan, R.; Hobbs, R.; and Abdelzaher, T. F. 2015. On exploiting logical dependencies for minimizing additive cost metrics in resource-limited crowdsensing. In *DCOSS, Fortaleza, Brazil, June 2015*, 189–198.

Ipeirotis, P. G. 2010. Analyzing the amazon mechanical turk marketplace. *XRDS* 17(2):16–21.

Kamar, E.; Kapoo, A.; and Horvitz, E. 2015. Identifying and Accounting for Task-Dependent Bias in Crowdsourcing. In *HCOMP, San Diego, CA, November 2015*.

Karger, D. R.; Oh, S.; and Shah, D. 2011. Iterative learning for reliable crowdsourcing systems. In *NIPS, Granada, Dec 2011*.

Kazemi, L.; Shahabi, C.; and Chen, L. 2013. Geotrucrowd: trustworthy query answering with spatial crowdsourcing. In *SIGSPATIAL, Orlando, FL, November 2013*, 304–313.

Keally, M.; Zhou, G.; Xing, G.; Nguyen, D. T.; and Qi, X. 2014. A learning-based approach to confident event detection in heterogeneous sensor networks. *ACM ToSN* 11(1):10.

Kerman, M. C.; Jiang, W.; Blumberg, A. F.; and Buttrey, S. E. 2009. Event detection challenges, methods, and applications in natural and artificial systems. Technical report, DTIC Document.

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

Kong, X. 2008. *Sampling Strategies in Sensor Networks*. ProQuest.

Krause, A.; Horvitz, E.; Kansal, A.; and Zhao, F. 2008. Toward community sensing. In *IPSN, St. Louis, MO, April 2008*, 481–492.

Li, H.; Li, T.; and Wang, Y. 2015. Dynamic participant recruitment of mobile crowd sensing for heterogeneous sensing tasks. In *MASS, Dallas, TX, October 2015*, 136–144.

Lin, S.; Arai, B.; Gunopulos, D.; and Das, G. 2008. Region sampling: Continuous adaptive sampling on sensor networks. In *ICDE, Cancun, Mexico, April 2008*, 794–803.

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

Panagiotou, N.; Zygouras, N.; Katakis, I.; Gunopulos, D.; Zacheilas, N.; Boutsis, I.; Kalogeraki, V.; Lynch, S.; and O'Brien, B. 2016. Intelligent urban data monitoring for smart cities. In *ECML-PKDD, Riva del Garda, Italy, September 2016*.

Qi, X.; Keally, M.; Zhou, G.; Li, Y.; and Ren, Z. 2013. Adasense: Adapting sampling rates for activity recognition in body sensor networks. In *RTAS, Philadelphia, PA, April 2013*, 163–172.

Roy, S. B.; Lykourentzou, I.; Thirumuruganathan, S.; Amer-Yahia, S.; and Das, G. 2015. Task assignment optimization in knowledge-intensive crowdsourcing. *The VLDB Journal* 24(4):467–491.

Sakaki, T.; Okazaki, M.; and Matsuo, Y. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW, Raleigh, NC, April 2010*, 851–860.

Shannon, C. E. 2001. A mathematical theory of communication. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* 5(1).

Subramaniam, S.; Kalogeraki, V.; and Palpanas, T. 2006. Distributed real-time detection and tracking of homogeneous regions in sensor networks. In *RTSS, Rio de Janeiro, Dec 2006*, 401–411.

Thejaswini, M.; Rajalakshmi, P.; and Desai, U. B. 2015. Novel sampling algorithm for human mobility-based mobile phone sensing. *Internet of Things Journal, IEEE* 2(3):210–220.

Van Dyke Parunak, H., and Brueckner, S. 2001. Entropy and self-organization in multi-agent systems. In *AGENTS, May 2001*.

Wang, T.-Y., and Cheng, Q. 2008. Collaborative event-region and boundary-region detections in wireless sensor networks. *IEEE Transactions on Signal Processing* 56(6):2547–2561.

Wang, D.; Abdelzaher, T.; Kaplan, L.; and Aggarwal, C. 2013. Recursive fact-finding: A streaming approach to truth estimation in crowdsourcing applications. In *ICDCS, Philadelphia, July 2013*.

Wang, S.; Su, L.; Li, S.; Hu, S.; Amin, T.; Wang, H.; Yao, S.; Kaplan, L.; and Abdelzaher, T. 2015. Scalable social sensing of interdependent phenomena. In *IPSN, Seattle, WA, April 2015*, 202–213.

Weinschrott, H.; Weisser, J.; Durr, F.; and Rothermel, K. 2011. Participatory sensing algorithms for mobile object discovery in urban areas. In *PerCom, Seattle, USA, March 2011*, 128–135.

Willett, R.; Martin, A.; and Nowak, R. 2004. Backcasting: adaptive sampling for sensor networks. In *IPSN, Berkeley, CA, Apr 2004*.

Zhang, D.; Xiong, H.; Wang, L.; and Chen, G. 2014. Crowdrecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint. In *UbiComp, Seattle, WA, September 2014*, 703–714.

Zhao, D.; Li, X.-Y.; and Ma, H. 2014. How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint. In *INFOCOM, Toronto, Canada, Apr 2014*.

Zhuang, H., and Young, J. 2015. Leveraging in-batch annotation bias for crowdsourced active learning. In *WSDM, January 2015*.

Zhuang, H.; Parameswaran, A.; Roth, D.; and Han, J. 2015. Debiasing crowdsourced batches. In *KDD, Sydney, Australia, Aug 2015*.