Truth Discovery on Crowd Sensing of Correlated Entities

Chuishi Meng[1], Wenjun Jiang[1], Yaliang Li[1], Jing Gao[1], Lu Su[1], Hu Ding[1], Yun Cheng[2]
[1]SUNY Buffalo, Buffalo, NY USA [2]Air Scientific, Beijing, China
{chuishim,wenjunji,yaliangl,jing,lusu,huding}@buffalo.edu,chengyun.hit@gmail.com

ABSTRACT

With the popular usage of mobile devices and smartphones, crowd sensing becomes pervasive in real life when human acts as sensors to report their observations about entities. For the same entity, users may report conflicting information, and thus it is important to identify the true information and the reliable users. This task, referred to as truth discovery, has recently attracted much attention. Existing work typically assumes independence among entities. However, correlations among entities are commonly observed in many applications. Such correlation information is crucial in the truth discovery task. When entities are not observed by enough reliable users, it is impossible to obtain true information. In such cases, it is important to propagate trustworthy information from correlated entities that have been observed by reliable users. We formulate the task of truth discovery on correlated entities as an optimization problem in which both truths and user reliability are modeled as variables. The correlation among entities adds to the difficulty of solving this problem. In light of the challenge, we propose both sequential and parallel solutions. In the sequential solution, we partition entities into disjoint independent sets and derive iterative approaches based on block coordinate descent. In the parallel solution, we adapt the solution to MapReduce programming model, which can be executed on Hadoop clusters. Experiments on real-world crowd sensing applications show the advantages of the proposed method on discovering truths from conflicting information reported on correlated entities.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Human Factors, Performance

SenSys'15, November 1-4, 2015, Seoul, South Korea.
© 2015 ACM. ISBN 978-1-4503-3631-4/15/11 ...$15.00.
DOI: http://dx.doi.org/10.1145/10.1145/2809695.2809715.

Keywords

Crowd Sensing; Truth Discovery; Correlation

1. INTRODUCTION

Crowd sensing has emerged as a new way of gathering information about entities by collecting observations from humans. With the ubiquitous mobile sensing devices (e.g., smartphones), it becomes easier and easier for the population to sense and share the information they perceived. Specifically, users can report certain conditions of their environment, such as traffic conditions, broken public utilities, gas prices, weather conditions and air quality, to central servers. On the central server, user-contributed information is aggregated to obtain useful information. Much attention is now contributed to the development of crowd sensing systems for various application domains [4, 7, 14, 21, 32, 38].

An important task in the crowd sensing system is the aggregation of user-contributed information. Users may provide conflicting and noisy information on the same entity, and then how to discover the true information (i.e., the truth) among these conflicting observations is the key question. Especially when most users report false information, the true information is unable to be discovered by voting or averaging. Intuitively, we should trust information from reliable users, i.e., users who often report true information. However, users' reliability degrees are not known *a priori.* To tackle the challenge of discovering both true information and user reliability, truth discovery methods [26, 27, 28, 44, 47] have been proposed based on the principle that reliable users tend to report true information and truth should be reported by many reliable users.

Although truth discovery techniques can be applied to crowd sensing applications to extract true information, existing methods do not take into consideration the correlations among entities in truth discovery. In fact, correlations exist ubiquitously among entities. For example, nearby segments of the same road may share similar traffic conditions, locations in the same area may have similar weather conditions, and a station may have similar gas prices during a short period. Such correlation information can greatly benefit the truth discovery process—information obtained from reliable sources can be propagated over all correlated entities, such that the aggregated information is more trustworthy. Especially, taking correlations into consideration will be more helpful when the coverage rate is low, i.e., users only provide observations for a small portion of the entities. In such cases, many entities may receive observations from unreliable users, and thus reliable information borrowed from correlated entities is important. In contrast, if we regard entities as independent ones and ignore their correlations, the truths on entities that are not observed by reliable users cannot be discovered.

In this paper, we propose an effective truth discovery framework for crowd sensing of correlated entities. In the crowd sensing systems, users make their observations on some entities among a set of correlated entities, and report the observations to a central server via smartphones. We propose an approach that jointly infers user reliability and truths among correlated entities. We formulate the task as an optimization problem, in which truths and user reliability are unknown variables, and correlations are modeled as regularization terms. The regularization terms make the optimization difficult to solve due to the correlations among unknown variables. To tackle this challenge, we propose to partition the variables into disjoint independent sets, and conduct block coordinate descent to iteratively update truths and user reliabilities. To process large-scale data, we further propose a parallel solution implemented on Hadoop cluster. Experiments on three crowd sensing applications show the effectiveness and efficiency of the proposed approaches.

In summary, we make the following contributions:

- We propose an effective optimization-based framework to solve the problem of truth discovery for crowd sensing of correlated entities. The proposed objective function measures the differences between the user-input observations and the unknown truths, and integrates users' reliabilities as unknown weights. The correlation regularization terms punish the deviations in the truths between correlated entities.
- To tackle the challenge introduced by the regularization terms, we propose to partition entities into disjoint independent sets, and then develop an iterative solution based on block coordinate descent with convergence guarantee. We introduce an approach that partition entities into disjoint independent sets for general cases, and propose effective ways to construct disjoint independent sets for the crowd sensing task with spatial and temporal correlations.
- We further propose a MapReduce based solution for the truth discovery of correlated entities. This is implemented on a Hadoop cluster and we show its ability of processing large-scale data.
- We conduct experiments on three crowd sensing datasets, i.e., air quality sensing, gas price inference and weather condition estimation datasets. Results demonstrate that the proposed approaches outperform existing methods in discovering true information for correlated entities.

The rest of the paper is organized as follows. We describe the system overview in Section 2. The proposed optimization framework and solutions are detailed in Section 3. In Section 4, we describe the proposed parallel solution based on MapReduce model. Evaluations are shown in Section 5. We review the related work in Section 6 and conclude the paper in Section 7.

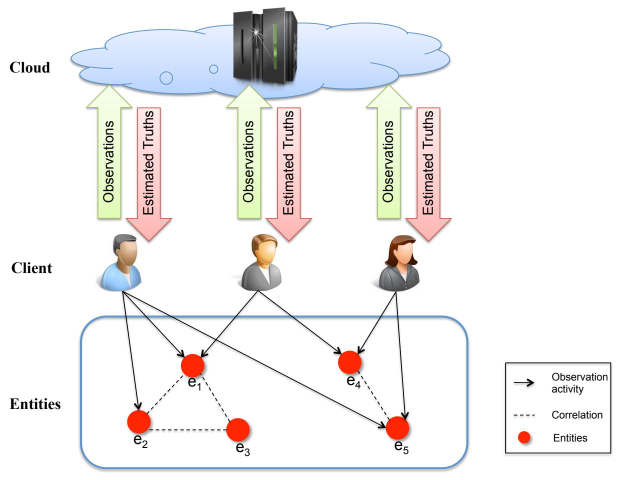


Figure 1: System Framework

2. SYSTEM OVERVIEW

In this section, we introduce the problem setting and the problem formulation. We start with some definitions.

Definition 1. *An* entity *is a thing or phenomenon which can be observed by crowd users; a* user *is a crowd sensing participant who contributes information about the entities; and an* observation *is the information perceived by a particular user on a particular entity.*

For example, the temperature at a particular location and time is an entity for a weather reporting application. Each participant is a user, and the temperature observed and reported by a user at a location and time is an observation.

Definition 2. *A* truth *is the true value of an entity. A user's* weight *is the reliability degree of the information provided by the user.*

Note here truths and user weights are unknown, and they are the expected output. Correlations between entities are considered: If two entities are correlated, their truths should be close to each other. Consider the aforementioned example. The truth is the real temperature on a particular location and time, and a weight is assigned to a user which indicates how likely the observations reported by him/her are reliable, i.e., how close they are to the truths. Then the problem we try to solve is to estimate both truths and user weights given the observations collected from users.

With these definitions, we now describe the crowd sensing system. Figure 1 shows the framework of this system. Users make observations on entities, and upload their observations to the cloud server via smartphones. On the server side, the proposed truth discovery method is conducted to derive both truths and user weights. In the following, we discuss details about the processes at the user and server sides respectively.

Client. On the client side, each user provides observations about entities, and uploads them to the server. The entities can be actively or passively observed, and the users may make the observations by themselves or report the observations with the help of sensors. Take WeatherSignal[1] as an example. It is a crowd sensing application which can

[1]http://weathersignal.com

not only automatically collect temperature, pressure data via sensors integrated in users' smartphones, but also allow users to manually report the weather conditions.

Server. On the server side, once observations are collected from users, the proposed truth discovery method will be conducted. The reliability degree of each user is calculated, together with the estimated truths for entities. As discussed, correlation information among entities is important, and thus it is taken into consideration in the proposed algorithm which will be detailed in Section 3.

3. TRUTH DISCOVERY ON CORRELATED ENTITIES

In this section, we describe the proposed truth discovery algorithm on correlated entities. The optimization framework and its solution are discussed. We then propose the parallel solution based on MapReduce model in Section 4.

3.1 Proposed Framework

We first define the mathematical notations for the problem. Each entity is indexed with i, and e_i denotes the i-th entity. Each user is indexed with k. An observation is represented as $x_i^{(k)}$ which indicates that it is observed by user k on entity e_i. All the observations made by user k is represented as $\mathcal{X}^{(k)}$. The truth for entity e_i is denoted as $x_i^{(*)}$, and the truths of all the entities are defined as $\mathcal{X}^{(*)}$. The weight for user k is w_k, and the collection of all the user weights is $\mathcal{W}$. Table 1 summarizes the important notations used in this paper. With these notations, we can formally define the problem as follows.

Problem Definition. Suppose there are N entities and K users. The input are observations $\{\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \cdots, \mathcal{X}^{(K)}\}$ collected from all the K users and the pairwise correlations among entities. Let $N(i)$ denote the set of entities which have correlations with entity e_i. The expected output are the truths $\mathcal{X}^{(*)}$ and the user weights $\mathcal{W}$.

The intuitions behind the proposed method are that truths should be close to the observations given by the reliable users, and correlated entities should have similar true values. Based on these intuitions, we formulate the following optimization problem:

$$\min_{\mathcal{X}^{(*)},\mathcal{W}} f(\mathcal{X}^{(*)}, \mathcal{W}) = \sum_{i=1}^{N} \left\{ \sum_{k=1}^{K} w_k ||x_i^{(*)} - x_i^{(k)}||^2 + \alpha \sum_{i' \in N(i)} S(i,i') ||x_{i'}^{(*)} - x_i^{(*)}||^2 \right\} \quad (1)$$
$$\text{s.t.} \quad \sum_{k=1}^{K} \exp(-w_k) = 1$$

Here α is a hyper parameter that balances between the two terms in the objective function. $S(i, i')$ is a similarity function which captures the correlation degree of two entities. Note here the design of this similarity function depends on the specific application since entity correlations have different properties in different applications. For example, one way of modeling the similarity under spatial correlation is Gaussian kernel, $S(i, i') = \exp(-d^2(i, i')/\sigma^2)$. Here $d(i, i')$ is the Euclidean distance of two entities and σ is a scaling parameter that controls how fast the similarity decreases as the distance increases. In general, the Gaussian kernel is a measure of similarity between entity i and i'. It evaluates to 1 if the two input values are identical, and approaches 0 as they move further apart.

The objective function $f(\mathcal{X}^{(*)}, \mathcal{W})$ in Eq(1) captures the intuitions we have mentioned. The first term $\sum_{i=1}^{N} \sum_{k=1}^{K} w_k ||x_i^{(*)} - x_i^{(k)}||^2$ aims at minimizing the disagreement between observations and truths, among which the disagreement on entities from the users with higher reliability degrees (i.e. w_k) are weighed higher. This means that higher penalties are given to more reliable users if their observations deviate from the corresponding truths. The second term in the objective function $\sum_{i=1}^{N} \sum_{i' \in N(i)} S(i, i') ||x_{i'}^{(*)} - x_i^{(*)}||^2$ models the other principle, i.e., the truth of an entity should be close to the truths of the entities it has correlation relationships with. In addition, the similarity function ensures that more attention should be paid to the entities with correlation degrees. By this objective function, we ensure that 1) the truths are picked from information contributed by reliable users and 2) correlated entities have similar truths.

The constraint in Eq(1) is used to restrict the range of weights, otherwise the weight may go to negative infinity.

In sum, with the proposed framework in Eq(1), we search for the values for two sets of variables, i.e. truths $\mathcal{X}^{(*)}$ and user weights $\mathcal{W}$, by minimizing the objective function $f(\mathcal{X}^{(*)}, \mathcal{W})$ under the constraint.

For an optimization problem with two sets of variables, it is natural to use block coordinate descent approach [3] to solve the problem. The idea is to iteratively update one set of variables while fixing the other one. However, due to the existence of the regularization term, it is non-trivial to solve Eq(1) as variables are correlated. In order to tackle this challenge, we first construct disjoint independent sets of entities based on their correlation relationships. Here, an independent set is a set of entities among which no correlations exist. For example, entities shown in Figure 1 can form three disjoint independent sets. There are five entities which are denoted as nodes in the graph, and the links indicate the correlations. As can be seen, e_1, e_2, e_3 are correlated with each other, and e_4, e_5 are correlated. The three independent sets are thus $\{e_1, e_4\}$, $\{e_2, e_5\}$ and $\{e_3\}$, in which entities within each set are not correlated. The methods to construct disjoint independent sets will be discussed in Section 3.3.

Here, we assume that independent sets are obtained. We denote j as the index for an independent set. We use $\mathcal{I} = \bigcup_{j=1}^{J} \mathcal{I}_j$ to denote the set containing all the entities, and $\mathcal{I}_j$ denotes the j-th subset. As truths are defined for entities, the collection of truths $\mathcal{X}^{(*)}$ can thus be divided into subsets based on independent sets, i.e., $\mathcal{X}^{(*)} = \{\mathcal{X}_1^{(*)}, \mathcal{X}_2^{(*)}, \cdots, \mathcal{X}_J^{(*)}\}$.

Based on the disjoint independent sets, Eq(1) can be rewritten as follows:

$$\min_{\mathcal{X}^{(*)},\mathcal{W}} f(\mathcal{X}^{(*)}, \mathcal{W}) = \sum_{\mathcal{I}_j \subset \mathcal{I}} \sum_{i \in \mathcal{I}_j} \left\{ \sum_{k=1}^{K} w_k ||x_i^{(*)} - x_i^{(k)}||^2 + \alpha \sum_{i' \in N(i)} S(i,i') ||x_i^{(*)} - x_{i'}^{(*)}||^2 \right\} \quad (2)$$
$$\text{s.t.} \quad \sum_{k=1}^{K} \exp(-w_k) = 1$$

Table 1: Frequently Used Notations

Symbol	Definition
K	number of users
k	index of users
w_k	weight of user k
$\mathcal{W}$	the set contains all user weights
N	number of entities
i	index of entities
e_i	the i_{th} entity
$\mathcal{I}$	the set contains all entities
$\mathcal{I}_j$	the j_{th} independent set of entities
J	number of independent sets
$x_i^{(*)}$	the true value of entity i
$x_i^{(k)}$	observation provided by user k on entity i
$\mathcal{X}^{(*)}$	the set of true values of all entities
$\mathcal{X}_j^{(*)}$	the set of true values of entities from independent set j
$N(i)$	entities which have correlation with entity i
$S(i,i')$	the similarity degree between two entities
$\mathcal{Y}$	the set of all variables

Comparing Eq(1) and Eq(2), the outer-most summation over entities in Eq(1) is decomposed into the summation over all independent sets which is further decomposed into summations over entities within one set. By this means, we are able to solve the problem with the solution proposed in the next section.

3.2 Proposed Solution

In Eq(2), user weights $\mathcal{W}$ and truths $\mathcal{X}_1^{(*)}, \mathcal{X}_2^{(*)}, \cdots, \mathcal{X}_J^{(*)}$ should be learned simultaneously to minimize the objective function. In order to achieve this goal, block coordinate descent approach [3] is utilized to iteratively update the values of one set while fixing the values of others until convergence. There are two cases at each iteration: One is to update user weights $\mathcal{W}$ while fixing all the sets of truths $\{\mathcal{X}_1^{(*)}, \mathcal{X}_2^{(*)}, \cdots, \mathcal{X}_J^{(*)}\}$, the other is to update one set of truths $\mathcal{X}_j^{(*)}$ while fixing user weights and all the other sets of truths $\{\mathcal{W}, \mathcal{X}_1^{(*)}, \cdots, \mathcal{X}_{j-1}^{(*)}, \mathcal{X}_{j+1}^{(*)}, \cdots, \mathcal{X}_J^{(*)}\}$. We discuss the details of the two cases as follows.

Update $\mathcal{W}$ while fixing $\{\mathcal{X}_1^{(*)}, \mathcal{X}_2^{(*)}, \cdots, \mathcal{X}_J^{(*)}\}$: In this case, all sets of truths are fixed, then user weights $\mathcal{W}$ can be computed based on the difference between the truths and the observations made by the user:

$$\mathcal{W} \leftarrow \arg\min_{\mathcal{W}} f(\mathcal{X}^{(*)}, \mathcal{W}) \quad \text{s.t.} \quad \sum_{k=1}^{K} \exp(-w_k) = 1. \tag{3}$$

We can solve this optimization problem based on Lagrange multipliers approach. The Lagrangian of Eq(2) is given as:

$$L(\{w_k\}_{k=1}^{K}, \lambda) = \sum_{\mathcal{I}_j \subset \mathcal{I}} \sum_{i \in \mathcal{I}_j} \left\{ \sum_{k=1}^{K} w_k ||x_i^{(*)} - x_i^{(k)}||^2 \right.$$
$$\left. + \alpha \sum_{i' \in N(i)} S(i,i') ||x_{i'}^{(*)} - x_i^{(*)}||^2 \right\} + \lambda (\sum_{k=1}^{K} \exp(-w_k) - 1),$$

where λ is a Lagrange multiplier. Let the partial derivative of Lagrangian with respect to w_k be 0, we get:

$$\sum_{\mathcal{I}_j \subset \mathcal{I}} \sum_{i \in \mathcal{I}_j} ||x_i^{(*)} - x_i^{(k)}||^2 = \lambda \exp(-w_k). \tag{4}$$

From the constraint that $\sum_{k=1}^{K} \exp(-w_k) = 1$, we can derive that

Algorithm 1 Truth Discovery on Correlated Entities

```
Input: Observations from $K$ users: $\{\mathcal{X}^{(1)}, \dots, \mathcal{X}^{(K)}\}$.
Output: Truths $\mathcal{X}^{(*)} = \{x_i^{(*)}\}_{i=1}^{N}$, user weights $\mathcal{W} = \{w_k\}_{k=1}^{K}$.
 1: Initialize the truths $\mathcal{X}^{(*)}$;
 2: for $i \leftarrow 1$ to $N$ do
 3:     Construct a correlation graph $G_{corr}$ in which an edge
        denotes the existence of correlation between the two nodes;
 4: end for
 5: Partition entities into disjoint independent sets $\{\mathcal{I}_1, \cdots, \mathcal{I}_J\}$
    in $G_{corr}$;
 6: repeat
 7:     Update user weights $\mathcal{W}$ according to Eq(6) to infer user
        reliability degrees based on the estimated truths;
 8:     for $j \leftarrow 1$ to $J$ do
 9:         for $i \in \mathcal{I}_J$ do
10:             Update the truth of the $i_{th}$ entity $x_i^{(*)}$ from the
                set $\mathcal{I}_J$ according to Eq(8) based on the current estimates of
                user weights;
11:         end for
12:     end for
13: until Convergence criterion is satisfied;
14: return $\mathcal{X}^{(*)}$ and $\mathcal{W}$.
```

$$\lambda = \sum_{k=1}^{K} \sum_{\mathcal{I}_j \subset \mathcal{I}} \sum_{i \in \mathcal{I}_j} ||x_i^{(*)} - x_i^{(k)}||^2. \tag{5}$$

We can then derive the update rule for each user's weight by plugging Eq(5) into Eq(4):

$$w_k = -\log \left(\frac{\sum_{\mathcal{I}_j \subset \mathcal{I}} \sum_{i \in \mathcal{I}_j} ||x_i^{(*)} - x_i^{(k)}||^2}{\sum_{\mathcal{I}_j \subset \mathcal{I}} \sum_{i \in \mathcal{I}_j} \sum_{k'=1}^{K} ||x_i^{(*)} - x_i^{(k')})||^2} \right), \tag{6}$$

where k' denotes the index of a user. This update rule shows that a user's weight is higher when his observations are more often close to the truths.

Update $\mathcal{X}_j^{(*)}$ while fixing $\{\mathcal{W}, \mathcal{X}_1^{(*)}, \cdots, \mathcal{X}_{j-1}^{(*)}, \mathcal{X}_{j+1}^{(*)}, \cdots,$ $\mathcal{X}_J^{(*)}\}$: In this case, user weights $\mathcal{W}$ are fixed, and the sets of truths are also fixed except $\mathcal{X}_j^{(*)}$. We update the truth for each entity in $\mathcal{X}_j^{(*)}$ by minimizing the objective function:

$$\mathcal{X}_j^{(*)} \leftarrow \arg\min_{\mathcal{X}_j^{(*)}} f(\mathcal{X}^{(*)}, \mathcal{W}). \tag{7}$$

Let the derivative of Eq(2) be 0 with respect to $x_i^{(*)}$, then for each $i \in \mathcal{I}_j$, or equivalently for each $x_i^{(*)} \in \mathcal{X}_j^{(*)}$, we have the following update rule:

$$x_i^{(*)} = \frac{\sum_{k=1}^{K} w_k x_i^{(k)} + \alpha \sum_{i' \in N(i)} S(i,i') x_{i'}^{(*)}}{\sum_{k=1}^{K} w_k + \alpha \sum_{i' \in N(i)} S(i,i')} \tag{8}$$

At this step, we can solve for $x_i^{(*)}$ with Eq(8) because the correlated variables are separated into disjoint independent sets. Specifically, we only update truths from independent set $\mathcal{X}_j^{(*)}$, which means that their correlated entities are not in this set, but in the other sets. Note that we have already fixed the truths of the other sets, and thus they are constants. As shown in Eq(8), $x_i^{(*)}$ is updated by averaging over the observed values weighted by user weights $\sum_{k=1}^{K} w_k x_i^{(k)}$ and the values of its correlated entities $\sum_{i' \in N(i)} x_{i'}^{(*)}$.

The proposed method is summarized in Algorithm 1. Disjoint independent sets $\{\mathcal{I}_1, \cdots, \mathcal{I}_J\}$ are first constructed according to the correlation information among entities (to be discussed in the following section). Then the algorithm iteratively updates user weights and truths based on Eq(6) and Eq(8) until convergence criterion is satisfied.

3.3 Construction of Disjoint Independent Sets

As discussed in Section 3.1, disjoint independent sets of entities should first be formed before the iterative solution can be used to solve Eq(1). Therefore, how to partition entities into disjoint independent sets is an important step. In this section, we first discuss how to form disjoint independent sets for any general cases in Section 3.3.1. Special cases of dealing with spatial and temporal correlations are discussed in Section 3.3.2 and Section 3.3.3 respectively.

3.3.1 General Cases

We can form a correlation graph based on the pairwise correlations among entities. In the graph, each vertex represents an entity and two vertices have an edge if they are correlated (i.e., two vertices are associated). Then the task of partitioning entities into disjoint independent sets is equivalent to the vertex coloring problem on the correlation graph. A vertex coloring is an assignment of colors to each vertex of a graph such that no edge connects two identically colored vertices. The problem of finding minimum number of colors for a given graph, i.e. minimum vertex coloring, is an NP-complete problem. However, some heuristic algorithms have been proposed, which are efficient and can produce good solutions in practice. We adopt the well-known Dsatur algorithm [5] to find disjoint independent sets in our task.

The procedure is described as follows. We first define the *saturation degree* of a vertex as the number of different independent sets that are covered by the associated vertices. The main idea is to iteratively select the vertex (entity) with the maximal saturation degree and put the selected entity into the corresponding independent set. The algorithm starts by arranging the vertices by a decreasing order of degrees, and the vertex with the maximal degree is put into independent set 1. After the initialization, an iterative process is performed to put entities into independent sets. During each iteration, the entity with the maximal saturation degree is chosen. In the tie case, we can break the tie by choosing the entity with the maximal degree. Then the selected entity is put into the independent set with the smallest index. This process continues until all the entities are processed.

In sum, given any correlation graph, we are able to construct disjoint independent sets by applying the aforementioned method. However, this step may still be time consuming especially when there are a large amount of entities. Fortunately, efficient methods can be applied to certain types of correlations, such as temporal and spatial correlations. In the following, we present the proposed approaches of independent set construction for these two special cases, which have only $O(1)$ time complexity.

3.3.2 Temporal Correlations

In many real-world applications, temporal correlations exist among entities. Examples include weather conditions (e.g., temperature, humidity) and air quality measures (e.g., PM2.5, NO_x). Temporal correlation is a local correlation and it typically exists only within a certain time window, i.e., the values of entities within a small time window are similar. This property enables us to develop a more efficient way to construct disjoint independent sets.

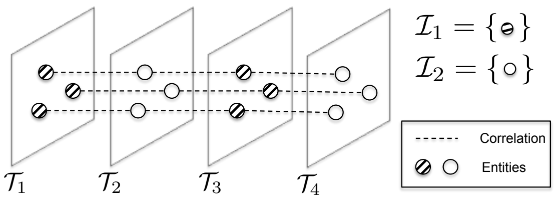


Figure 2: Separating entities based on temporal correlation when $p = 1$

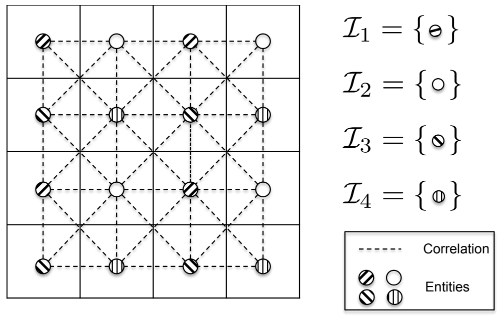


Figure 3: Separating entities based on spatial correlation

Let $\mathcal{T}_i$ denote the collection of entities in the i-th time slot, and $\mathcal{I}_i$ denote the i-th independent set. Without loss of generality, we assume that temporal correlations exist within $p+1$ consecutive time slots, i.e., entities correlate with all those from previous p time slots. We also assume that there are $(p+1)(Q+1)$ time slots in total. Then $p+1$ disjoint independent sets can be formed as follows. In each set, there are $Q+1$ time slots:

$$\left\{ \mathcal{I}_i = \bigcup_{q=0}^{Q} \mathcal{T}_{i+(p+1)q} \right\}_{i=1}^{p+1}.$$

Specifically, when $p = 1$, we only consider the temporal correlations among two consecutive time slots, i.e., entities only correlate with those from the previous time slot. In this case, all entities in odd-numbered time slots $\mathcal{I}_1$ and even-numbered time slots $\mathcal{I}_2$ can form two disjoint independent sets. An example is shown in Figure 2 in which we have four time slots $\{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{T}_4\}$. Then two disjoint independent sets can be formed as $\mathcal{I}_1 = \{\mathcal{T}_1, \mathcal{T}_3\}$ and $\mathcal{I}_2 = \{\mathcal{T}_2, \mathcal{T}_4\}$.

3.3.3 Spatial Correlations

Spatial correlations are another type of correlations that widely exist in real world. For example, gas prices among gas stations and weather conditions within certain geographical areas are usually very similar. Spatial correlation is also a local property which allows the design of an efficient method of disjoint independent set construction. Specifically, entities distributed on a gridded map can be separated into four disjoint independent sets. Let $e_{(i,j)}$ be the entity on the i-th row and j-th column of the gridded map, and let $p = \{1, 2, \cdots, P\}$ and $q = \{1, 2, \cdots, Q\}$ be the indices. Four independent sets can be constructed as follows:

$$\mathcal{I}_1 = \left\{ e_{(2p,2q-1)} \right\}_{p=1,q=1}^{P,Q}, \mathcal{I}_2 = \left\{ e_{(2p-1,2q-1)} \right\}_{p=1,q=1}^{P,Q},$$
$$\mathcal{I}_3 = \left\{ e_{(2p-1,2q)} \right\}_{p=1,q=1}^{P,Q}, \mathcal{I}_4 = \left\{ e_{(2p,2q)} \right\}_{p=1,q=1}^{P,Q}.$$

They are formed in a way that entities in even rows and odd columns form independent set $\mathcal{I}_1$, those in odd rows and odd columns form $\mathcal{I}_2$, those in odd rows and even columns form $\mathcal{I}_3$ and those in even rows and even columns form $\mathcal{I}_4$. Figure 3 illustrate an example of the four disjoint independent sets constructed on a gridded map. As can be seen, entities in each set are not correlated with the others in the same set.

3.4 Convergence Analysis

Here, we formally prove the convergence of the proposed iterative approach shown in Algorithm 1 by the following theorem.

THEOREM 1. *The iterations shown in Algorithm 1 converge and the solution is a stationary point of the optimization problem in Eq(2).*

PROOF. We can simplify the optimization problem as follows when variables are represented as $\mathcal{Y} = \{\mathcal{W}, \mathcal{X}_1^{(*)}, \mathcal{X}_2^{(*)}, \cdots, \mathcal{X}_J^{(*)}\}$

$$\begin{aligned} &\text{minimize} \quad f(y) \\ &\text{s.t.} \quad y \in \mathcal{Y} \end{aligned} \tag{9}$$

Suppose for each $y_i \in \mathcal{Y}_i$,

$$f(y_1, \cdots, y_{i-1}, \xi, y_{i+1}, \cdots, y_m)$$

viewed as a function of ξ, attains a unique minimum $\bar{\xi}$ over Y_i. Let $\{y^k\}$ be the sequence generated by the following updating rule

$$y_i^{k+1} \leftarrow \arg\min_{\xi \in Y_i} f(y_1^{k+1}, \cdots, y_{i-1}^{k+1}, \xi, y_{i+1}^k, \cdots, y_m^k)$$

According to the proof for block coordinate descent in [3], every limit point of $\{y^k\}$ is a stationary point if the above conditions hold.

In the following, we only need to prove that Algorithm 1 satisfies the conditions. Specifically, we need to prove that each iteration generates a unique minimum for the updated variables. This is proved based on the two cases of updates:

Case One: This is to update $\mathcal{W}$ while fixing $\{\mathcal{X}_1^{(*)}, \mathcal{X}_2^{(*)}, \cdots, \mathcal{X}_J^{(*)}\}$. In order to show the convexity of Eq(2), we introduce another variable t_k, so that $t_k = \exp(-w_k)$. Then Eq(2) can be expressed in terms of t_k as follows:

$$\begin{aligned} \min_{\{t_k\}_{k=1}^K} f(t_k) = &\sum_{\mathcal{I}_j \subset \mathcal{I}} \sum_{i \in \mathcal{I}_j} \left\{ \sum_{k=1}^K -\log(t_k) ||x_i^{(*)} - x_i^{(k)}||^2 \right. \\ &\left. + \alpha \sum_{i' \in N(i)} ||x_i^{(*)} - x_{i'}^{(*)}||^2 \right\} \\ \text{s.t.} \quad &\sum_{k=1}^K t_k = 1 \end{aligned} \tag{10}$$

The objective function of Eq(10) is a linear combination of negative logarithm functions and constants, and thus it is convex. In addition, the constraint is linear in t_k, which is affine. Therefore, Eq(2) is convex while fixing all the truths, such that a unique minimum for w_k can be achieved with the update rule in Eq(6).

Case Two: This is to update $\mathcal{X}_i^{(*)}$ while fixing $\{\mathcal{W}, \mathcal{X}_1^{(*)}, \cdots, \mathcal{X}_{i-1}^{(*)}, \mathcal{X}_{i+1}^{(*)}, \cdots, \mathcal{X}_J^{(*)}\}$. When fixing user weights $\mathcal{W}$, Eq(2) is a summation of quadratic functions with respect to $x_i^{(*)}$. It is convex since these quadratic functions are convex and summation operation preserves convexity. As a consequence, a unique minimum for $x_i^{(*)}$ can be achieved with the update rule in Eq(8).

Therefore, this proves the convergence of the proposed method. □

3.5 Time Complexity

We analyze the time complexity of the proposed method by analyzing each iteration's running time. Assume there are K users and N entities. The user weight update step costs $O(KN)$ time because for each user we calculate the square error between its observations and truths. In the truth update step, for each entity we calculate the weighted sum of observations from K users, the sum of correlated values and the sum of K users' weights. Totally it also needs $O(KN)$ time. Therefore, the time complexity is linear with respect to the number of observations.

4. MAPREDUCE IMPLEMENTATION

Crowd sensing applications usually have a large amount of participants who may generate a large amount of observations. To conduct truth discovery on this "big data", we hope to take advantage of cloud computing techniques to process large-scale data in parallel. Among parallel programming models, MapReduce [11] is widely adopted for many data mining tasks on large-scale data. In this section, we describe our design of a MapReduce model based parallel algorithm on Hadoop platform.

A typical MapReduce model contains two phases: 1) the map phase reads the input data, and converts it into key-value pairs; 2) the reduce phase takes the key-value pairs generated from the map phase as input, and performs the needed operations on them. For the proposed truth discovery task, the objective is to adapt Algorithm 1 to a parallel version. However, this is a non-trivial task. The challenge is that truths are updated according to Eq(8) with respect to each independent set. This requires us to fix all the other variables while updating those within one independent set. Obviously, this prevents us from deriving an efficient parallel process to compute truths all at once.

In order to solve this problem, we design a MapReduce algorithm based on asynchronous parallel coordinate descent. The proposed method iteratively calculates user weights and truths. During each iteration, the input data include observations from all K users $\{\mathcal{X}^{(k)}\}_{k=1}^K$, truths and user weights estimated from the last iteration (at iteration t)—$\mathcal{X}^{(t)(*)} = \{x_i^{(t)(*)}\}_{i=1}^N$ and $\mathcal{W}^{(t)} = \{w_k^{(t)}\}_{k=1}^K$. The outputs are truths and user weights calculated in the current iteration (at iteration $t+1$). The proposed method is shown in Algorithm 2. In the following, we will describe the details of the functions used in the proposed MapReduce truth discovery algorithm.

Input Data Format. The input data is formatted as tuples of three elements: the ID of an entity (denote as i), the ID of a user (denote as k) and the observation by the k-th user

Algorithm 2 MapReduce Implementation

```
Input: Data from $K$ users: $\{\mathcal{X}^{(1)},\ldots,\mathcal{X}^{(K)}\}$.
    Truths at iteration $t$: $\mathcal{X}^{(t)(*)} = \{x_i^{(t)(*)}\}_{i=1}^N$
    User weights at iteration $t$: $\mathcal{W}^{(t)} = \{w_k^{(t)}\}_{k=1}^K$
Output: Truths at iteration $t+1$: $\mathcal{X}^{(t+1)(*)}$
    User weights at iteration $t+1$: $\mathcal{W}^{(t+1)}$

 1: function MAP(Key $id$, Value $v$)
 2:     $(i, x_i^{(k)}, k) \leftarrow v$;
 3:     $observ\_error \leftarrow (x_i^{(k)} - x_i^{(t)(*)})^2$;
 4:     EMIT($-k$, $observ\_error$);
 5:     EMIT($i$, $(x_i^{(k)}, k)$);
 6: end function

 7: function COMBINE(Key $id$, Value[] $v$)
 8:     if $id > 0$ then
 9:         $combined\_value = null$;
10:         for $v_i \in v$ do
11:             Append $v_i$ to $combined\_value$;
12:         end for
13:     end if
14:     if $id < 0$ then
15:         $combined\_value = 0$;
16:         for $v_i \in v$ do
17:             $combined\_value = combined\_value + v_i$;
18:         end for
19:     end if
20:     EMIT($id$, $combined\_value$);
21: end function

22: function REDUCE(Key $id$, Value[] $v$)
23:     if $id > 0$ then
24:         $[x_i^{(k)}, k] \leftarrow v$;
25:         Calculate $x_{id}^{(t+1)(*)}$ according to Eq(8);
26:         EMIT($id$, $x_{id}^{(*)(t+1)}$);
27:     end if
28:     if $id < 0$ then
29:         $id \leftarrow -id$;
30:         Calculate nominator of $w_{id}^{(t+1)}$ according to Eq(6);
31:         EMIT($-id$, $w_{id}^{(t+1)}$);
32:     end if
33: end function

34: Calculate $w_{id}^{(t+1)}$ according to Eq(6);
35: return $\mathcal{X}^{(t+1)(*)} = \{x_i^{(t+1)(*)}\}_{i=1}^N$ and
    $\mathcal{W}^{(t+1)} = \{w_k^{(t+1)}\}_{k=1}^K$.
```

for the i-th entity (denote as $x_i^{(k)}$). In this way, the input is represented in the form of $(i, x_i^{(k)}, k)$.

Map Function. The input to the Map function is a collection of tuples. The Map function pre-processes input tuples and outputs key-value pairs to be used in the Reduce function. In order for Reduce function to calculate truths, one output key-value pair includes entity id i (i.e., the key) and $(x_i^{(k)}, k)$ (i.e., the observed value together with its user id). Also, in order for Reduce function to calculate user weights, the Map function pre-calculates the squared observation error based on the observation $x_i^{(k)}$ and the truth estimated from the last iteration $x_i^{(t)(*)}$. Then another key-value pair is emitted, which is user id k (i.e., the key) and the observation error (i.e., the value). Here we use a positive index as the emitted entity id, and a negative one as the emitted user id. This helps the Reduce function to distinguish between these two types.

Reduce Function. The Reduce function takes the key-value pairs emitted from the Map function as input. Truths and user weights are calculated and emitted as outputs. Upon receiving key-value pairs with positive keys, the Reduce function calculates the truths according to Eq(8). Each output tuple includes entity id and the calculated truth for this entity. If the Reduce function receives key-value pairs with negative keys, the nominator in the user weight calculation (Eq(6)) is calculated and emitted. Final user weights will be derived in the wrapper function.

Combine Function. Since the number of entities and users can be quite large, the overhead from the communication and sorting operations may dominate the running time. In order to reduce this kind of overhead, a Combine function is designed. It works in a similar way with the Reduce function. For key-value pairs with positive keys, the combiner will return local summation over the observation errors with the key. As for key-value pairs with negative keys, the combiner will just append the value and output the key-value pair.

Wrapper Function. A wrapper function is designed to control the iterative procedure. This function first initializes the truths as the mean of the observations provided by different users, and initializes user weights as $\frac{1}{K}$. Then the initial values and input data are fed to the Map function. Truths can be directly collected from the output of the Reduce function and user weights can be derived according to Eq(6) with the returned values from the Reduce function. We repeat the whole process until the estimated truths converge or after a certain number of iterations.

5. EVALUATIONS

In this section, we report the experimental results on datasets from three real-world crowd sensing applications, i.e., air quality sensing, gas price inference and weather condition estimation datasets. The results demonstrate that the proposed method outperforms state-of-the-art truth discovery methods because it incorporates correlations among entities into the model. The experiment setup is discussed in Section 5.1. Results are discussed in Section 5.2, Sections 5.3 and 5.4. Running time of the proposed method on Hadoop cluster is presented in Section 5.5.

5.1 Experiment setup

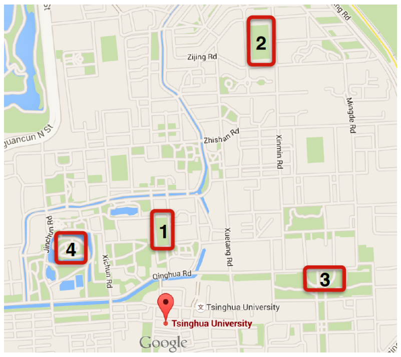
In this section, we present the performance measures and discuss the baseline methods which are compared in the experiments.

5.1.1 *Performance Measures*

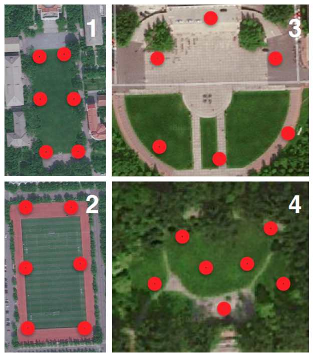
In the experiments, the inputs for all methods are observations about entities given by different users. The outputs are the estimated truths and user weights. For each dataset, we have the ground truths, i.e., the actual true values of entities. However, they are not used by the proposed approach or the baselines, but are only used for evaluation. Note that all datasets contain continuous data, and thus we adopt the following measures to evaluate the performance.

- *Mean Absolute Distance* (*MAD*) measures the overall absolute distance between each method's outputs and

(a) Air Quality Sensing Devices



(b) Areas for Sensing



(c) Locations for Sensing

Figure 4: Air quality sensing devices and locations. (a) shows the mini-AQM, its smartphone App and the thermo. The mini-AQM is portable and carried by participants. Thermo is the device for collecting ground truths. (b) shows the four areas designated for air quality sensing at Tsinghua University. (c) shows the locations designated for sensing in each area.

the ground truths, which is computed by averaging the absolute difference over all the entities.

- *Root Mean Square Error* (*RMSE*) is computed by taking the root of the mean squared differences between each method's outputs and the ground truths.

MAD and *RMSE* both measure the differences between outputs compared with ground truths. The lower the measure, the closer the method's outputs to the ground truths, and the better it performs. *RMSE* emphasizes on larger errors compared with *MAD*.

5.1.2 *Baseline Methods*

In the experiment, the proposed Truth Discovery method on correlated entities is denoted as TD-corr. We compare it with the following state-of-the-art truth discovery methods which are designed to work on continuous data:

Gaussian Truth Model (GTM) is a Bayesian model that solves the problem of truth discovery on continuous data. Conflict Resolution on Heterogeneous data (CRH) is a framework that infers the truths from multiple sources with different data types, such as continuous and categorical data. Confidence-Aware Truth Discovery (CATD) is proposed to detect truths from multi-source data with long-tail phenomenon. Besides these truth discovery approaches, simple averaging methods, such as Mean and Median, are also compared.

We implement all the baselines and set the parameters as suggested in the corresponding papers. All the experiments are conducted on a Windows machine with 8G RAM, Intel Core i7 processor.

5.2 Experiments on Air Quality Sensing System

Air quality monitoring has drawn great attention these days, since people are suffering from deteriorated air quality over the past years, especially for cities in developing countries such as Beijing and New Delhi. In this experiment, we target the sensing of particulate matter with diameter less than 2.5 micron (PM2.5) with mini-AQM – a portable air quality sensing device designed for personal use [1, 7], as shown in Figure 4(a). The readings of mini-AQM are uploaded automatically into the server and can be checked on users' smartphone App as well. We have 14 participants equipped with mini-AQMs and let them conduct sensing tasks in 4 areas at Tsinghua University (Figure 4(b)). These areas are selected to represent different types of environments. Within each area, there are 6 or 7 locations (Figure 4(c)). The participants perform air quality sensing on all locations of each area within one hour. The PM2.5 values of the locations within one area are correlated, since the areas are limited in scale and the time window is short. The collected data have 25 entities and each entity represents the PM2.5 value of one location. The ground truths are collected with Thermo [2] (Figure 4(a)), an accurate but expensive sensing device.

5.2.1 *Experimental Results*

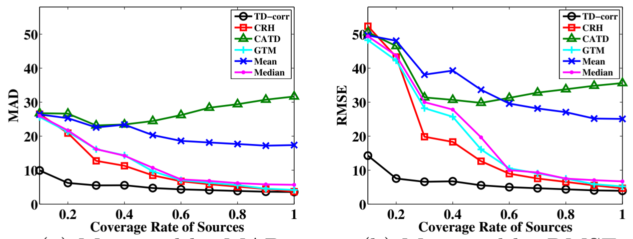
Performance w.r.t users' coverage rate. The results on this air quality sensing dataset are summarized in Table 2 measured by MAD and RMSE. To inspect the results visually, we also show the results in Figure 5. As users may not have observations for all the entities in many real-world applications, we conduct experiments on different coverage rates of users. The rate is defined as the percentage of entities that are observed by the user. For example, the coverage rate of 1.0 means that the user provides observations for all entities, and the coverage rate of 0.2 means that the user only provides observations for 20% of entities. We compare the performance of all the approaches when the coverage rate varies from 0.1 to 1.0.

As shown in Table 2 and Figure 5, TD-corr outperforms all the baselines under any coverage rate, as the MAD and RMSE of TD-corr are the lowest. These results demonstrate the advantage of the proposed approach which incorporates important correlation information in truth discovery. Note here CATD performs worse compared with other truth discovery methods. The reason is that CATD is designed for data with long tail effect and is not suitable for this data.

Another observation from Table 2 and Figure 5 is that the improvement over the baselines increases when the coverage rate decreases. For example, when the coverage rate is 0.8, TD-corr performs 25% better than the best baseline, and when the coverage rates are 0.4 and 0.2, TD-corr per-

Table 2: Performance Comparison on Air Quality Sensing System w.r.t Varying Coverage Rate

	Coverage Rate									
	0.2		0.4		0.6		0.8		1.0	
Method	MAD	RMSE	MAD	RMSE	MAD	RMSE	MAD	RMSE	MAD	RMSE
TD-corr	**6.2442**	**7.5426**	**5.5579**	**6.7327**	**4.3506**	**4.9962**	**3.8850**	**4.3650**	**3.5830**	**3.9234**
CRH	20.8943	43.2959	11.2952	18.3067	6.7254	8.9643	5.1653	6.5588	3.6809	4.6713
CATD	26.6218	46.4943	23.4346	30.6744	26.1897	31.2542	29.3994	33.7851	31.6699	35.6129
GTM	21.2988	42.2436	14.3768	25.7079	7.0405	10.5310	5.6053	7.5208	4.1867	5.2125
Mean	25.2916	48.0523	23.3865	39.2801	18.6068	29.5817	17.6935	27.0485	17.3739	25.0735
Median	21.6812	43.7129	14.2530	27.8183	7.2927	10.0167	6.1827	7.5137	5.6931	6.7028



(a) Measured by MAD (b) Measured by RMSE

Figure 5: Performance Comparison on Air Quality Sensing System w.r.t Varying Coverage Rate

forms 51% and 70% better than the best baseline respectively. The reason is that when the coverage rate is lower, the number of entities observed by reliable users also decreases. Thus, there will be more entities receiving observations mostly from unreliable users, or even worse, receiving all the observations from unreliable users. The baseline methods purely infer truths by integrating over those unreliable observations, and thus their estimates are biased. In contrast, the proposed method is able to discover the truths because it takes correlated entities into consideration. Then information observed by reliable users can be propagated to entities that are not observed by reliable users and helps infer the truths of those entities.

5.3 Experiments on Gas Price Inference

An interesting crowd sensing application is to solicit gas price reports from crowd users and aggregate their answers to get correct gas prices. In this application, each user reports the observed gas prices for some gas stations to the server—some may report true values but some of the reported information may be wrong. The server then conducts truth discovery to identify the true price for each station.

To conduct this experiment, we first collect gas prices of 2175 gas stations from the GasBuddy website[2], which are treated as the ground truths. The data are collected from 48 cities across the whole US continent for one day. Correlations exist among gas stations within the same city. The observations of users with different reliability degrees are generated by adding different levels of Gaussian noise upon the ground truths. There are 2175 entities in the dataset. We conduct each experiment for 10 times and report the average results.

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

5.3.1 Experimental Results

Performance w.r.t users' coverage rate. The results with 30 users are summarized in Table 3 and Figure 6 measured by MAD and RMSE. Compared with the experiments on the air quality dataset, similar results can be observed. The proposed method outperforms the baselines especially when the coverage rate is low.

Performance w.r.t number of users. Figure 7 demonstrates the performance when varying the number of users from 10 to 100. The results show the following: 1) TD-corr performs better than the baselines on the data with any number of users. The incorporation of correlation information leads to a better estimation of truths. 2) The improvement is more significant when there are fewer users. The reason is when the number of users is smaller, there are more entities receiving observations mainly from less reliable users, and the estimated truths would be biased towards unreliable reports. The proposed method's advantage in propagating reliable information over correlated entities is thus clearly shown.

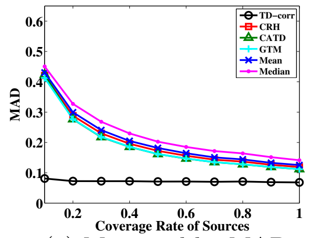
Correlation accuracy. Here we show that the proposed approach indeed takes correlation among entities in the estimation of truths. We define the measure of correlation accuracy as follows. We measure the similarity between entities based on the estimated truths, and regard two entities as correlated ones if their distance passes a threshold. We then compare the correlations derived from the estimated truths and the ground-truth correlations. Then the correlation accuracy is defined as the percentage of correlations that are the same as the ground truths. Figure 8 shows the correlation accuracy under different thresholds (0.01, 0.03, 0.05, 0.07, 0.09, 0.1, 0.2, 0.3). Figure 8(a) and Figure 8(b) show the results when coverage rates are 20% and 80% respectively. It can be seen that TD-corr captures the correlation information better than the baselines under any setting, and its correlation accuracy can reach a high value even when the threshold is low, such as 0.05 or 0.07.

5.4 Experiments on Weather Condition Estimation

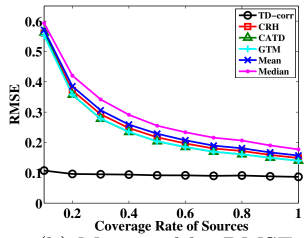
Another interesting crowd sensing application is to collect and aggregate weather conditions reported by users. User can report weather information, such as temperature, pressure, humidity. Data can be acquired with integrated sensors in smartphones or manually input by users. The truth discovery approach can then be conducted on the collected noisy data to identify the true weather conditions. In order to emulate this application, we collect weather forecast data from three weather forecast platforms (Wunder-

Table 3: Performance Comparison on Gas Price Inference w.r.t Varying Coverage Rate

	Coverage Rate									
	0.2		0.4		0.6		0.8		1.0	
Method	MAD	RMSE	MAD	RMSE	MAD	RMSE	MAD	RMSE	MAD	RMSE
TD-corr	**0.0727**	**0.0963**	**0.0725**	**0.0941**	**0.0713**	**0.0917**	**0.0713**	**0.0912**	**0.0685**	**0.0867**
CRH	0.2881	0.3714	0.1959	0.2475	0.1563	0.1968	0.1370	0.1720	0.1191	0.1488
CATD	0.2766	0.3574	0.1854	0.2344	0.1469	0.1850	0.1280	0.1610	0.1120	0.1400
GTM	0.2770	0.3582	0.1856	0.2348	0.1471	0.1852	0.1281	0.1610	0.1120	0.1400
Mean	0.2994	0.3851	0.2051	0.2590	0.1645	0.2071	0.1442	0.1810	0.1256	0.1570
Median	0.3272	0.4204	0.2299	0.2912	0.1853	0.2337	0.1640	0.2064	0.1415	0.1772

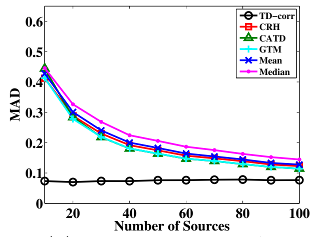


(a) Measured by MAD

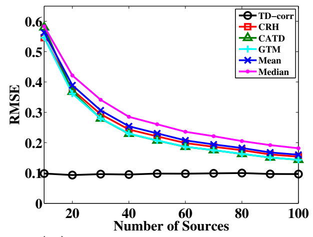


(b) Measured by RMSE

Figure 6: Performance Comparison on Gas Price Inference w.r.t Varying Coverage Rate



(a) Measured by MAD



(b) Measured by RMSE

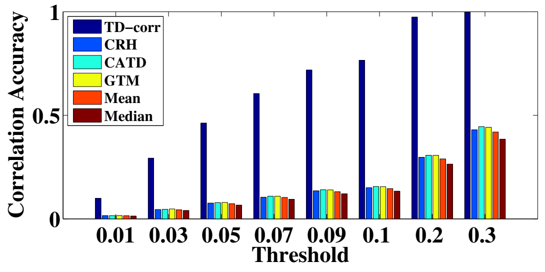
Figure 7: Performance w.r.t Varying Number of Users

ground[3], HAM weather[4], and World Weather Online[5]) for 147 locations within New York City Area. From each platform, we collect temperature forecasts of two different days as two different users, thus 6 users are formed in total. We also crawl the ground truths of hourly temperature for each location. An entity here is defined as the temperature of one location at one time. The spatial correlation is modeled using Euclidean distance with Gaussian kernel, and the temporal correlation exists among entities from the same location in consecutive time slots. The whole process of collection lasts over a month and there are 145459 entities in the dataset.

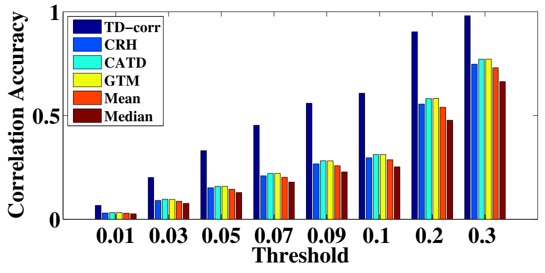
5.4.1 Experimental Results

Table 4 and Figure 9 summarize the performance of TD-corr and baselines on the weather condition estimation dataset. Similar results can be observed, compared with the experiments on the previous two datasets.

[3]http://www.wunderground.com
[4]http://www.hamweather.com
[5]http://www.worldweatheronline.com



(a) Coverage Rate is 20%



(b) Coverage Rate is 80%

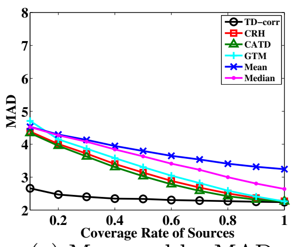
Figure 8: Correlation Accuracy w.r.t Varying Threshold

To demonstrate the proposed approach's ability of distinguishing reliable users from unreliable ones, we compare the reliability degrees estimated by each truth discovery method. The reliability degree of a user is defined as the overall similarity between its observations and truths. The ground-truth reliability degree is calculated and compared as well. In the compared methods, different functions are adopted to calculate the reliability degree of users. For the sake of comparison, we normalize the reliability output into the range of [0,1]. Figure 10 and Figure 11 show the comparison results when coverage rates are 0.2 and 0.8 respectively. To make it clear, we demonstrate the user reliability degrees in two plots, and each of them shows the comparison between the ground truths and some of the approaches. TD-corr and ground truths are compared in Figures 10(a) and 11(a). Other baseline methods are compared in Figures 10(b) and 11(b).

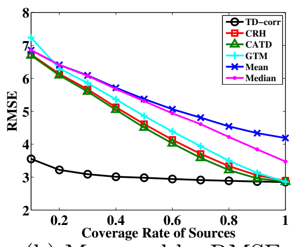
We can observe that the user reliability estimated by TD-corr is very close to the user reliability derived from the ground truth. With the consideration of correlation information, the proposed TD-corr approach can better estimate the truths. Due to the intertwined process of truth and user reliability estimation, such an improved truth estima-

Table 4: Performance Comparison on Weather Condition Estimation w.r.t Varying Coverage Rate

	Coverage Rate									
	0.2		0.4		0.6		0.8		1.0	
Method	MAD	RMSE	MAD	RMSE	MAD	RMSE	MAD	RMSE	MAD	RMSE
TD-corr	**2.5081**	**3.2734**	**2.3700**	**3.0434**	**2.2984**	**2.9327**	**2.2691**	**2.8919**	**2.2420**	**2.8520**
CRH	4.0757	6.2663	3.4305	5.1666	2.8943	4.1617	2.5022	3.3352	2.2708	2.8955
CATD	4.0093	6.2208	3.3308	5.0819	2.7883	4.0444	2.4211	3.2127	2.2638	2.8598
GTM	4.7015	7.3111	3.6016	5.3953	3.0506	4.3992	2.5939	3.4938	2.2641	2.8600
Mean	4.3614	6.5426	3.9766	5.7515	3.6624	5.0880	3.4089	4.5359	3.2419	4.1882
Median	4.3418	6.5432	3.8614	5.7051	3.4252	4.9632	3.0016	4.2171	2.6409	3.4795

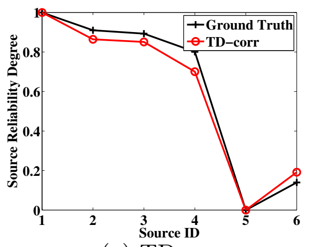


(a) Measured by MAD

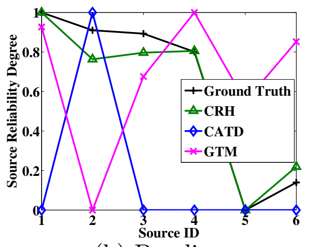


(b) Measured by RMSE

Figure 9: Performance Comparison on Weather Condition Estimation w.r.t Varying Coverage Rate

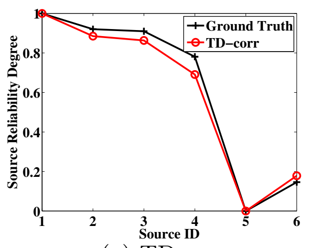


(a) TD-corr

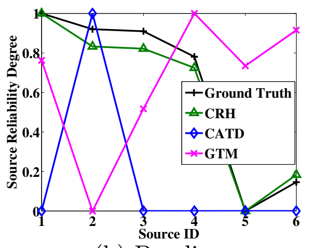


(b) Baselines

Figure 10: Comparison of User Reliability Degrees with Ground Truths (Coverage = 0.2)



(a) TD-corr



(b) Baselines

Figure 11: Comparison of User Reliability Degrees with Ground Truths (Coverage = 0.8)

tion leads to an improved estimation of user reliability. In contrast, due to the fact that the baselines do not consider the important correlation information, their truth and user reliability estimation may not be accurate, so we can observe the deviation of their estimated user reliability from that of the ground truth.

Practical convergence property of the proposed method. The proposed method converges as discussed in Section 3.4. In order to show the convergence in practice, we conduct experiments on the weather dataset and record objective function value at each iteration. As can be observed from Figure 12, the objective value decreases quickly in the first 10 iterations, then gradually reaches to a steady value. This demonstrates the convergence of the proposed method.

5.5 Efficiency Evaluation on Hadoop Cluster

Running time on hadoop cluster. As discussed in Section 4, the proposed method can be implemented based on the MapReduce model. We conduct experiments to demonstrate the running time of this parallel algorithm. Datasets are generated with different numbers of observations, varying from 10^4 to 10^8. The experiments are conducted on a 4-node Dell Hadoop cluster with Intel Xeon E5-2403 processor (4x 1.80 GHz, 48 GB RAM). Results are summarized in Table 5, which show that the running time is linear with respect to the number of observations. We calculate the Pearson product-moment correlation coefficient to further justify this. The Pearson product-moment correlation coefficient is a measure of the linear correlation between two variables. It gives a value between $+1$ and -1, where $+1$ indicates positive correlation, 0 indicates no correlation and -1 indicates negative correlation. The Pearson's correlation coefficient in Table 5 shows a strong linear correlation between the number of observations and the running time. This means that the proposed method is scalable and can be applied to large-scale data.

6. RELATED WORK

We review related work based on the following two categories.

Crowd Sensing. The research of crowd sensing [4, 6, 8, 9, 10, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 32, 35, 36, 37, 38, 40, 42] has received more and more attention with the rapid development of pervasive sensors, smartphones, and network technologies. Thanks to the rich set of sensors (e.g., accelerometer, gyroscope, GPS, microphone, and camera) integrated in the mobile devices, and the pervasive WiFi and cellular networks, now people can record and share the information they perceive wherever they are and whenever they want. Recently, a variety of sensing systems have been designed for a wide spectrum of applications. Huang et al. [21] implemented a system for the search and rescue of people in emergency situations in wilderness areas. In their system, RF-based sensors, storage and processing devices are used. Eisenman et al. [14] designed a system for the cyclists to collect and share data with each other about their performance and the surrounding environment. Reddy et al. [38] designed a smartphone based platform for the cyclists to document and share routes, ride statistics, experience and etc. Mun et al. [32] proposed an application which

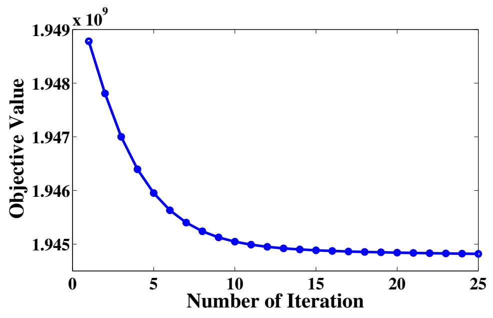


Figure 12: Convergence

Table 5: Running Time on Hadoop Cluster

# Observations	Time(s)
1×10^4	77
1×10^5	83
1×10^6	136
1×10^7	254
1×10^8	413
Pearson Correlation	0.9088

utilizes users' trajectory data to infer environmental impact and exposure. EasyTracker [4] is designed with functions that determine routes and local stops, and infer schedules with collected trajectory data. These papers focused on the design of the crowd sensing system, which is different from the task of truth discovery studied in this paper.

Truth Discovery. Truth discovery was first proposed by Yin et al. [47] as a Bayesian based approach in which the confidence of each observation is calculated as the product of the reliability of the sources that provide the observation. Later, the topic attracts much attention and methods are developed to infer both source reliabilities and true facts. Pasternack et al. [33, 34] proposed methods to incorporate prior knowledge into truth discovery. In these methods, a source "invests" its reliability on the observations it provides, and credits are collected by the confidence on its invested observations. Galland et al. [15] modeled the difficulty of getting the truths while computing source weights. Yin et al. [48] proposed a semi-supervised method which can propagate the information trustworthiness obtained from the observed ground truths. Probabilistic graphical models were adopted in [49, 50] and source selection problem was studied in [12, 39]. A maximum likelihood estimation approach was proposed by Wang et al. in [44] which adopted expectation maximization method, and it was designed for categorical data. Recently, approaches have been proposed to handle heterogeneous data types [27, 26, 41], such as categorical data and continuous data. Ma et al. [30] proposed a fine grained truth discovery model to handle users with various expertise levels on different topics. In [31], Miao et al. proposed a cloud-enabled privacy-preserving truth discovery framework for crowd sensing systems. Note that the approaches discussed in this paragraph tackle some other challenges in truth discovery but do not consider correlations among entities.

In [43], the authors considered physical correlations among entities which are modeled with the joint probability of correlated variables, and derived an expectation maximization based algorithm to infer both entities' states and sources' reliabilities. In [46], they proposed a method to capture the temporal correlations among entities, and a joint probability of any given sequence of observed values are calculated from historical data to model the temporal correlation. In [45], the authors proposed a scalable algorithm which modeled the structure of correlations as a Bayesian network. The proposed algorithm is efficient because the conditional independence property in the network is exploited. However, these approaches work on binary data and correlations, and cannot be generalized to other data types, such as continuous data. In contrast, the proposed method in this paper works on continuous data and correlations, and thus the problem setting is quite different. Li et al. [29] proposed an incremental truth discovery framework that can dynamically update object truths and source weights when the information comes sequentially. However, they only consider the temporal relations for dynamically generated data, and the proposed framework in this paper can handle any types of correlations.

7. CONCLUSIONS

In this paper, we investigate the truth discovery problem on correlated entities in crowd sensing applications. In a crowd sensing system, crowd users contribute their information about entities to the server. Since users' reliability degrees are unknown *a priori*, the observations they provide should not be trusted equally. Then how to discover the true information among the conflicting observations is a crucial task. Although some approaches have been developed to detect truths from conflicting sources, these approaches do not take the valuable information about entity correlations into account and thus cannot detect truths accurately when the coverage is low. In fact, correlations among entities widely exist and can greatly benefit the truth discovery process. In this paper, we formulate the task of truth discovery on correlated entities as an optimization problem, which models both the distance between truths and observations and the distance among truths of correlated entities. User weights are plugged into the optimization function to capture the unknown user reliability. We tackle the difficulty caused by the regularization terms added upon correlated variables. In order to solve this problem, we propose methods to partition variables into disjoint independent sets, and conduct block coordinate descent to update truths and weights iteratively. The convergence of the approach is proved. To further speed up the process, we propose a MapReduce version of the algorithm that is implemented on Hadoop cluster. We conduct experiments on three crowd sensing datasets, i.e., air quality sensing, gas price inference and weather condition estimation datasets. Results demonstrate the advantages of the proposed method in discovering true information for correlated entities over existing truth discovery approaches.

8. ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers and our shepherd, Polly Huang, for their valuable comments and suggestions, which help us tremendously in improving the quality of the paper. This work was sponsored in part by US National Science Foundation under grant IIS-1319973. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agency.

9. REFERENCES

[1] Air scienctific. http://www.coilabs.com/.

[2] Thermo. http://www.thermoscientific.com.

[3] D. P. Bertsekas. *Nonlinear programming.* Athena scientific Belmont, 1999.

[4] J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson. Easytracker: Automatic transit tracking, mapping, and arrival time prediction using smartphones. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys'11).* ACM, 2011.

[5] D. Brélaz. New methods to color the vertices of a graph. *Commun. ACM*, 22(4):251–256, Apr. 1979.

[6] S. Chen, M. Li, K. Ren, X. Fu, and C. Qiao. Rise of the indoor crowd: Reconstruction of building interior view via mobile crowdsourcing. In *Proceedings of the 13th ACM Conference on Embedded Network Sensor Systems (SenSys'15)*, SenSys '15, 2015.

[7] Y. Cheng, X. Li, Z. Li, S. Jiang, Y. Li, J. Jia, and X. Jiang. Aircloud: A cloud-based air-quality monitoring system for everyone. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems (SenSys'14).* ACM, 2014.

[8] Y. Chon, N. D. Lane, Y. Kim, F. Zhao, and H. Cha. Understanding the coverage and scalability of place-centric crowdsensing. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing (UbiComp'13)*, pages 3–12. ACM, 2013.

[9] Y. Chon, N. D. Lane, F. Li, H. Cha, and F. Zhao. Automatically characterizing places with opportunistic crowdsensing using smartphones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp'12)*, pages 481–490. ACM, 2012.

[10] V. Coric and M. Gruteser. Crowdsensing maps of on-street parking spaces. In *Proceedings of the 9th International Conference on Distributed Computing in Sensor Systems (DCOSS'13)*, pages 115–122. IEEE, 2013.

[11] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[12] X. L. Dong, B. Saha, and D. Srivastava. Less is more: Selecting sources wisely for integration. *PVLDB*, 6(2):37–48, 2012.

[13] A. Dua, N. Bulusu, W.-C. Feng, and W. Hu. Towards trustworthy participatory sensing. In *Proceedings of the 4th USENIX conference on Hot topics in security (HotSec'09)*, pages 8–8. USENIX Association, 2009.

[14] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell. Bikenet: A mobile sensing system for cyclist experience mapping. *ACM Transactions on Sensor Networks (TOSN)*, 6(1):6, 2009.

[15] A. Galland, S. Abiteboul, A. Marian, and P. Senellart. Corroborating information from disagreeing views. In *Proc. of the ACM International Conference on Web Search and Data Mining (WSDM'10)*, pages 131–140, 2010.

[16] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher. Greengps: a participatory sensing fuel-efficient maps application. In *Proceedings of the 8th international conference on Mobile systems, applications, and services (MobiSys'10)*, pages 151–164. ACM, 2010.

[17] R. K. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: current state and future challenges. *Communications Magazine, IEEE*, 49(11):32–39, 2011.

[18] S. Hu, H. Liu, L. Su, H. Wang, T. F. Abdelzaher, P. Hui, W. Zheng, Z. Xie, and J. Stankovic. Towards automatic phone-to-phone communication for vehicular networking applications. In *The 33rd IEEE International Conference on Computer Communications (INFOCOM'14).* IEEE, 2014.

[19] S. Hu, L. Su, S. Li, S. Wang, C. Pan, S. Gu, T. Amin, H. Liu, S. Nath, R. R. Choudhury, et al. Experiences with enav: A low-power vehicular navigation system. In *The ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'15)*, 2015.

[20] S. Hu, L. Su, H. Liu, H. Wang, and T. F. Abdelzaher. Smartroad: Smartphone-based crowd sensing for traffic regulator detection and identification. *ACM Transactions on Sensor Networks (TOSN)*, 11(4):55, 2015.

[21] J.-H. Huang, S. Amjad, and S. Mishra. Cenwits: a sensor-based loosely coupled search and rescue system using witnesses. In *Proceedings of the 3rd international conference on Embedded networked sensor systems (SenSys'05)*, pages 180–191. ACM, 2005.

[22] K. L. Huang, S. S. Kanhere, and W. Hu. Are you contributing trustworthy data?: the case for a reputation system in participatory sensing. In *Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems (MSWIM'10)*, pages 14–22. ACM, 2010.

[23] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, and Q. Arshad. Mobile phone sensing systems: A survey. *Communications Surveys & Tutorials, IEEE*, 15(1):402–427, 2013.

[24] N. D. Lane, Y. Chon, L. Zhou, Y. Zhang, F. Li, D. Kim, G. Ding, F. Zhao, and H. Cha. Piggyback crowdsensing (pcs): energy efficient crowdsourcing of mobile sensor data by exploiting smartphone app opportunities. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys'13)*, page 7. ACM, 2013.

[25] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell. A survey of mobile phone sensing. *Communications Magazine, IEEE*, 48(9):140–150, 2010.

[26] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han. A confidence-aware approach for truth discovery on long-tail data. *Proc. VLDB Endow.*, 8(4):425–436, 2014.

[27] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *Proc. of the 2014 ACM SIGMOD International Conference on Management of Data (SIGMOD'14)*, pages 1187–1198, 2014.

[28] Y. Li, J. Gao, C. Meng, Q. Li, L. Su, B. Zhao, W. Fan, and J. Han. A survey on truth discovery. *arXiv preprint arXiv:1505.02463*, 2015.

[29] Y. Li, Q. Li, J. Gao, L. Su, B. Zhao, W. Fan, and J. Han. On the discovery of evolving truth. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15)*, pages 675–684. ACM, 2015.

[30] 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 *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15)*, pages 745–754. ACM, 2015.

[31] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, and K. Ren. Cloud-enabled privacy-preserving truth discovery in crowd sensing systems. In *Proceedings of the 13th ACM Conference on Embedded Network Sensor Systems (SenSys'15)*, SenSys '15, 2015.

[32] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda. Peir, the personal environmental impact report, as a platform for participatory sensing systems research. In *Proceedings of the 7th international conference on Mobile systems, applications, and services (MobiSys'09)*, pages 55–68. ACM, 2009.

[33] J. Pasternack and D. Roth. Knowing what to believe (when you already know something). In *Proc. of the International Conference on Computational Linguistics (COLING'10)*, pages 877–885, 2010.

[34] J. Pasternack and D. Roth. Making better informed trust decisions with generalized fact-finding. In *Proc. of the International Jont Conference on Artifical Intelligence (IJCAI'11)*, pages 2324–2329, 2011.

[35] K. K. Rachuri, C. Efstratiou, I. Leontiadis, C. Mascolo, and P. J. Rentfrow. Metis: Exploring mobile phone sensing offloading for efficiently supporting social sensing applications. In *Proceedings of the 11th International Conference on Pervasive Computing and Communications (PerCom'13)*, pages 85–93. IEEE, 2013.

[36] K. K. Rachuri, C. Mascolo, M. Musolesi, and P. J. Rentfrow. Sociablesense: exploring the trade-offs of adaptive sampling and computation offloading for social sensing. In *Proceedings of the 17th annual international conference on Mobile computing and networking (MobiCom'11)*, pages 73–84. ACM, 2011.

[37] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu. Ear-phone: an end-to-end participatory urban noise mapping system. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'10)*, pages 105–116. ACM, 2010.

[38] S. Reddy, K. Shilton, G. Denisov, C. Cenizal, D. Estrin, and M. Srivastava. Biketastic: sensing and mapping for better biking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'10)*, pages 1817–1820. ACM, 2010.

[39] T. Rekatsinas, X. L. Dong, and D. Srivastava. Characterizing and selecting fresh data sources. In *Proc. of the 2014 ACM SIGMOD International Conference on Management of Data (SIGMOD'14)*, pages 919–930, 2014.

[40] F. Saremi, O. Fatemieh, H. Ahmadi, H. Wang, T. Abdelzaher, R. Ganti, H. Liu, S. Hu, S. Li, and L. Su. Experiences with greengps–fuel-efficient navigation using participatory sensing. *IEEE Transactions on Mobile Computing (IEEE TMC)*.

[41] L. Su, Q. Li, S. Hu, S. Wang, J. Gao, H. Liu, T. F. Abdelzaher, J. Han, X. Liu, Y. Gao, et al. Generalized decision aggregation in distributed sensing systems. In *Proceedings of the 35th IEEE Real-Time Systems Symposium (RTSS'14)*, pages 1–10. IEEE, 2014.

[42] V. Subbaraju, A. Kumar, V. Nandakumar, S. Batra, S. Kanhere, P. De, V. Naik, D. Chakraborty, and A. MISRA. Conferencesense: A case study of sensing public gatherings using participatory smartphones. International Workshop on Pervasive Urban Crowdsensing Architecture and Applications (PUCAA'13), in conjunction with ACM Ubicomp'13, 2013.

[43] D. Wang, T. Abdelzaher, L. Kaplan, R. Ganti, S. Hu, and H. Liu. Exploitation of physical constraints for reliable social sensing. In *Proceedings of the 34th IEEE Real-Time Systems Symposium (RTSS'13)*, pages 212–223. IEEE, 2013.

[44] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher. On truth discovery in social sensing: A maximum likelihood estimation approach. In *Proceedings of the 11th international conference on Information Processing in Sensor Networks (IPSN'12)*, pages 233–244. ACM, 2012.

[45] S. Wang, L. Su, S. Li, S. Hu, T. Amin, H. Wang, S. Yao, L. Kaplan, and T. Abdelzaher. Scalable social sensing of interdependent phenomena. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks (IPSN'15)*, pages 202–213. ACM, 2015.

[46] S. Wang, D. Wang, L. Su, L. Kaplan, and T. F. Abdelzaher. Towards cyber-physical systems in social spaces: The data reliability challenge. In *Proceedings of the 35th IEEE Real-Time Systems Symposium (RTSS'14)*, pages 74–85. IEEE, 2014.

[47] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. *IEEE Transactions on Knowledge and Data Engineering*, 20(6):796–808, 2008.

[48] X. Yin and W. Tan. Semi-supervised truth discovery. In *Proc. of the International Conference on World Wide Web (WWW'11)*, pages 217–226, 2011.

[49] B. Zhao and J. Han. A probabilistic model for estimating real-valued truth from conflicting sources. In *Proc. of the VLDB workshop on Quality in Databases (QDB'12)*, 2012.

[50] B. Zhao, B. I. P. Rubinstein, J. Gemmell, and J. Han. A bayesian approach to discovering truth from conflicting sources for data integration. *PVLDB*, 5(6):550–561, 2012.