Tackling the Redundancy and Sparsity in Crowd Sensing Applications

Chuishi Meng[1], Houping Xiao[1],Lu Su[1], Yun Cheng[2]
[1]SUNY Buffalo, Buffalo, NY USA [2]Air Scientific, Beijing, China
{chuishim,houpingx,lusu}@buffalo.edu,chengyun.hit@gmail.com

ABSTRACT

Driven by the proliferation of sensor-rich mobile devices, crowd sensing has emerged as a new paradigm of gathering information about the physical world. In crowd sensing applications, user observations are usually unevenly distributed across the monitored entities, and this gives rise to two major challenges – redundancy and sparsity. On one hand, multiple users may observe the same entity, and their observations are sometimes conflicting with each other due to the unreliable nature of human-carried sensors. On the other hand, crowd sensing data are usually very sparse, and there may exist considerable number of entities that never receive any observations from users. Some existing work studies these two challenges separately. However, we can gain great benefits by dealing with them jointly. In this paper, we develop an integrated framework to estimate the true values of entities from redundant and sparse data in crowd sensing applications. In this framework, we propose an effective algorithm to infer the "missing" observations for each entity, and aggregate both user-contributed and inferred observations to discover the true values of entities. We conduct extensive experiments on real-world crowd sensing systems to demonstrate the advantages of the proposed framework on correctly inferring entity truths from redundant and sparse data.

CCS Concepts

•Information systems → Information systems applications;

Keywords

Crowd Sensing; Data Sparsity; Matrix Factorization; Truth Discovery; Correlation

1. INTRODUCTION

Nowadays, we have witnessed the ubiquitous adoption of mobile sensing devices (e.g., smartphones, smartglasses, smartwatches) with a plethora of integrated or portable sensors (e.g., accelerometer, camera, GPS). These devices make it easier for the population to sense and share the information they perceived. Thanks to these innovations, *crowd sensing* has emerged as a new way of collecting information from the physical world. In crowd sensing applications, humans work as the sensor carriers or even the sensors, and report what they learn about the conditions of the surrounding environment, such as weather, traffic, air quality and etc. The observations are then gathered in a central server, and aggregated to obtain useful knowledge. Various crowd sensing systems have been developed in different domains [6, 9, 14, 24, 39, 42]. However, the crowdsourced data collected in this way usually have two characteristics, redundancy and sparsity, which significantly reduce the effectiveness of crowd sensing systems. In the following, we will shed more light on how these two characteristics affect the crowd sensing system.

Redundancy. In a crowd sensing task, it is likely that multiple users observe the same entities. For example, in an air quality sensing system, each user carries a portable device that can transmit air quality readings to the central sever. At some popular locations during peak hours, many redundant reports about the air quality at the same location and the same time may be submitted. Conflicts are inevitable in the redundant data. So the *redundancy challenge* is: *among the conflicting observations, what is the true value?*

A naive approach is to conduct voting or averaging, i.e., take the value that is claimed by the majority of users, or the average value of the observations. The drawback of this simple approach is obvious: It treats all the users equally and fails to capture the variety in their reliability (which usually refers to the probability of a user providing true information). Such variety is caused by not only the quality of hardware but also the ways in which people use the hardware. For example, the user who carries the air quality sensor in hand can obviously report more accurate measurements than the user who puts the sensor in pocket. Intuitively, if we can identify and put more weight on the reliable users, the aggregation accuracy can be significantly improved. To tackle this challenge, some truth discovery methods [32, 34, 51] have been proposed to simultaneously estimate true values and user reliability from crowdsourced data.

Sparsity. Although redundancy could be alleviated by existing truth discovery approaches, these approaches may fail when facing the sparsity issue. In fact, sparsity is also widely observed in crowdsourced data. In crowd sensing applica-

SenSys '16, November 14-16, 2016, Stanford, CA, USA
 ISBN 978-1-4503-4263-6/16/11…$15.00
DOI: http://dx.doi.org/10.1145/2994551.2994567

tions, there are usually a large number of entities that we wish to observe, and some of the entities may never receive an observation from any user. So the *sparsity challenge* is: *how to estimate the true values of these "missing" entities with no observation data?*

To address this challenge, we can exploit the information about the similarity between entities since similar entities usually have similar values. Take the air quality sensing example again. It would be helpful to infer the air quality of unpopular areas where no users report any sensing measurements based on the crowdsourced data collected at neighboring popular areas.

Redundancy and sparsity usually *co-exist* in many crowd sensing tasks. In the aforementioned air quality sensing application, it is common that some locations receive multiple users' sensing measurements while some others get none. One straightforward approach to tackle both challenges works as follows. We first run truth discovery approaches to aggregate multiple users' observations on the observed entities, and then conduct interpolation to infer the true values of the "missing" entities based on the values of similar entities. The limitation of this simple approach is that it regards redundancy and sparsity as separate challenges.

However, we can gain great benefits by dealing with two challenges jointly: If we can estimate users' missing observations, the estimated values can be used to better infer users' reliability degrees in the truth discovery process. Typically, data sparsity also implies that some users may only give a few observations, and thus truth discovery methods may not be able to correctly estimate the reliability degrees of such users. Therefore, we propose a novel method that estimates users' missing observations based on the observed values as well as entity similarity information. After this, truth discovery methods can be used to aggregate all the observed and estimated values to fully unleash the power of crowdsourced information.

To realize the above idea, we develop an integrated framework, called *Redundancy and Sparsity Tackling* (**RST**) framework, to infer the true values of entities from redundant and sparse data in crowd sensing applications. In this framework, we design an effective optimization method that extracts key information from not only user-contributed observations but also similarities between entities to estimate the missing observations and recover a complete user-entity observation matrix. After missing observations are estimated, we conduct truth discovery on the observation matrix to derive the true value of each entity. As the first step fills in missing observations of users, it directly impacts the second step of truth discovery to achieve a more accurate estimation of user reliability, which in turn results in a more accurate estimation of true values. This integrated framework thus tackles both redundancy and sparsity challenges.

In summary, this paper makes the following contributions:

- This paper recognizes the effect of redundancy and sparsity on crowd sensing applications. To our best knowledge, this is the first work that tries to tackle these two challenges jointly in an integrated framework.
- In order to estimate the missing user observations, we formulate an optimization problem which captures both the key patterns of user-contributed data and entity similarity.
- An effective solution is developed to solve the proposed optimization problem in an iterative way. The convergence property of the proposed solution is proved, and effective techniques are presented to further reduce the time complexity.
- Extensive experiments on the task of air quality sensing are conducted in various regions in Beijing, China. The results demonstrate that the proposed method is able to recover missing user observations and derive accurate estimates of air quality measurements in various scenarios.

The rest of the paper is organized as follows. We describe the system overview in Section 2. The proposed redundancy and sparsity tackling framework and solutions are detailed in Section 3. Evaluations are shown in Section 4. We review the related work in Section 5 and conclude the paper in Section 6.

2. SYSTEM OVERVIEW

In this section, we first describe several important concepts followed by the problem definition, and then discuss the system architecture.

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.*

Here we take the air quality sensing application as an example. Air quality is vital to human health and thus it needs to be monitored. The measurements include PM2.5 (particulate matter with diameter less than 2.5 microns), SO_2 (sulfur dioxide), AQI (air quality index) and etc. In these sensing applications, the air quality measurement (e.g., PM2.5) at a particular location and time is an *entity*. Each participant is a *user*, and the value of air quality measurement that is observed and reported by a user at a given location and time is an *observation*.

DEFINITION 2. *A* truth *is the true value of an entity.*

Note here the truths of entities are unknown, and they are the expected output of the proposed framework. Consider the aforementioned example, the truth is the real air quality value at a particular location and time.

DEFINITION 3. *The* similarity *between two entities is defined based on the closeness between their true values.*

Under this definition, if two entities are similar, their true values should be close to each other. The entity similarity information is provided as the input to the proposed framework, which can be derived based on the characteristics of the specific sensing application. As for air quality sensing, the entity similarity can be computed based on the closeness between locations and time.

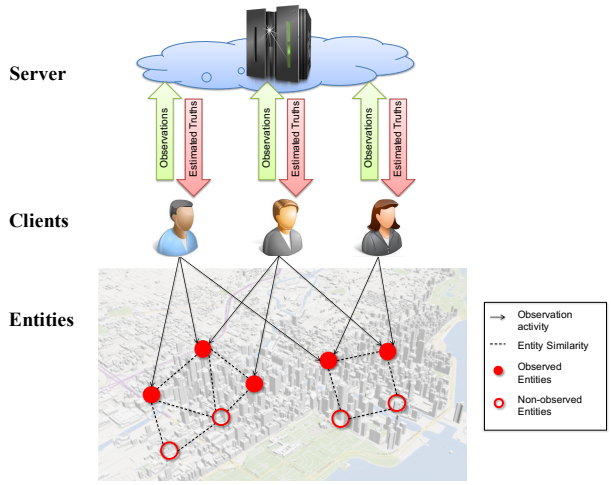


Figure 1: System Framework

Problem Definition. Given the redundant and sparse observation data collected from crowd sensing applications and the entity similarity information, the objective is to estimate the truth for each entity.

With the definitions above, we then discuss the designed crowd sensing system. As shown in Figure 1, users (clients) make observations on some of the entities and report their values to the central server. Note that a large number of entities may not receive any observation from any user. At the server side, the proposed method is performed and all the estimated entity truths are returned to the end users. The detailed procedure is discussed in the following.

Client. With the sensors integrated or connected to their mobile devices, users can sense the surrounding environment and report their readings to the central server. This process can be completed actively or passively, i.e., the sensing activity can be started and reported manually by users or according to an automatic schedule. For example, in a crowd sourced weather report application, "WeatherSignal"[1], the weather condition can be reported manually by users (such as "sunny", "rainy" and etc.) or it can be performed automatically by uploading the sensoring data collected from the integrated sensors on temperature, pressure and etc.

Server. Once the observations are collected from the users, the server performs the proposed method to estimate the true value of each entity. The output is returned to users to assist their decision making. For example, in the air quality sensing application, we convert redundant and sparse crowd sensing data into air quality measurements at each location/time, which are then provided to users as an effective air quality monitoring tool.

3. REDUNDANCY AND SPARSITY TACKLING FRAMEWORK

In this section, we present the proposed Redundancy and Sparsity Tackling (**RST**) framework. We discuss the formulation of missing observation estimation in Section 3.1, and its solution in Section 3.2. The truth discovery method used for aggregation is introduced in Section 3.3. Then we prove the convergence of the solution in Section 3.4.

[1]http://weathersignal.com

Table 1: Frequently Used Notations

Symbol	Definition
M	number of users
i	index of users
N	number of entities
j	index of entities
X	observation matrix of size $M \times N$: $X = \{x_{ij}\}$
x_{ij}	the observation on the j_{th} entity by the i_{th} user
X^*	truth vector of size $N \times 1$: $X^* = \{x_j^*\}$
x_j^*	true value of the j_{th} entity
H	sparse indicator matrix of size $M \times N$: $H = \{h_{ij}\}$
h_{ij}	$h_{ij} = 0$ if x_{ij} is missing, $h_{ij} = 1$ otherwise
A	entity similarity matrix of size $N \times N$: $A = \{a_{jj'}\}$
$a_{jj'}$	similarity between entity j and j'
L	$L = D - A$ where D is a diagonal matrix with $D_{jj} = \sum_{j'} a_{jj'}$
W	diagonal weight matrix of size $K \times K$: $W = diag(w_1, \cdots, w_K)$
w_k	weight of virtual user k
K	number of virtual users
V^*	aggregated virtual observation vector of size $N \times 1$: $V^* = \{v_j^*\}$
v_j^*	aggregated observation of all virtual users on the j_{th} entity
Q	$1 \times K$ vector in which each entry equals to 1
U	coefficient matrix of size $M \times K$: $U = \{u_{ik}\}$
u_{ik}	mapping coefficient from user i to virtual user k
V	virtual user matrix of size $N \times K$: $V = \{v_{jk}\}$
v_{jk}	value of the j_{th} entity by the k_{th} virtual user
$V_{\cdot p}, U_{\cdot p}$	the p_{th} column of matrix V and U
$V_{p\cdot}, U_{p\cdot}$	the p_{th} row of matrix V and U
S_e	sparsity level on entities
S_u	sparsity level on users
S	overall data sparsity level of the observation matrix

3.1 Problem Formulation

Suppose there are N entities and M users. The observation matrix is denoted as $X \in \mathbb{R}^{M \times N}$, where x_{ij} represents an observation on entity j provided by user i. In crowd sensing applications, each user usually provides observations for a subset of entities. Because of this, the observation matrix X is not complete, i.e., there are many "missing" entries in X, which indicates that certain users did not provide observations on some of the entities. We use an indicator matrix H to distinguish between missing and non-missing entries. H has the same size as X, where $h_{ij} = 0$ if x_{ij} is missing and $h_{ij} = 1$ if x_{ij} is observed. The final goal of this problem is to infer a vector $X^* = \{x_1^*, \ldots, x_j^*, \ldots, x_N^*\}$ in which x_j^* represents the true value of entity j.

We achieve this objective via the following two steps:

- **Missing Observation Estimation**. We estimate the "missing" entries in the observation matrix X. By filling in the missing observations, we are able to approximate what users observe about entities so that we can better capture users' reliability degrees in the next step.

- **Aggregation**. After all the missing entries are filled, we aggregate along each column (i.e., $\{x_{1j}, \ldots, x_{Mj}\}$) to obtain the true value of each entity x_j^*. A naive approach is to take the average of the values, but this approach does not take into account the important factor of user reliability. At this step, we adopt an existing truth discovery method [32] that estimates users' reli-

ability degrees and weigh each observation value based on users' reliabilities in the aggregation.

In this section, we describe the proposed method of missing observation estimation in detail and the aggregation method will be discussed in the following sections.

To fill in the missing entries of X, we propose the following optimization framework which integrates three important aspects:

$$\min \quad \mathbf{MF} + \mathbf{R1} + \mathbf{R2}, \tag{1}$$

where **MF** is a matrix factorization term, **R1** and **R2** are two regularization terms that capture the constraints on entities and users. These three terms are introduced below.

MF: Matrix factorization. The first term **MF** adopts matrix factorization techniques to find two matrices $U \in \mathbb{R}^{M\times K}$ and $V \in \mathbb{R}^{N\times K}$, such that their product provides a good approximation to X (i.e., $X \approx UV^T$). The intuition is that: We identify K virtual users (typically K is much smaller than $\min(M, N)$) whose values towards N entities are stored in matrix V, and we represent the observations of the original M users as a linear combination of the virtual users' values. Each row in matrix U represents the coefficients of the linear combination that maps this original user to the combination of K virtual users. In some sense, the virtual users capture commonalities among users from K angles, which serve as the basis to recover missing observations by original users. In other words, virtual users' observations V can be regarded as a compression of the original data which summarizes important characteristics of the original data.

To find a good approximation of X, the values of X and UV^T should be as close as possible on the observed entries. Thus we should find U and V that minimize the distance between X and UV^T on the entries where $h_{ij} = 1$: $\|H\circ(X - UV^{\mathrm{T}})\|^2$. Here, $\|\cdot\|$ denotes the Frobenius norm and $\circ$ is the Hadamard product which satisfies $R = P \circ S \Leftrightarrow r_{ij} = p_{ij}s_{ij}$. To alleviate overfitting, we restrain the norms of the two matrices (i.e., $||U||^2$ and $||V||^2$) to be small. Putting them together, the first term **MF** in the minimization problem is:

$$\mathbf{MF}: \quad \|H \circ (X - UV^{\mathrm{T}})\|^2 + \alpha(\|U\|^2 + \|V\|^2), \tag{2}$$

where α is a hyperparameter balancing the approximation error and the overfitting constraint. By minimizing this term, we obtain U and V whose norms are small and their product approximates X.

R1: Regularization on entity similarity. If we simply minimize Eq(2), we may encounter issues on entities that receive no observations from any user. When there are no observations on an entity, there are no virtual observations on this entity from virtual users neither. To infer the observations of these entities, we propose to incorporate the similarity information among entities into the optimization function.

We use a $N \times N$ matrix A to encode known similarity between entities, in which $a_{jj'}$ denotes the similarity between two entities j and j'. The higher $a_{jj'}$, the more similar the two entities are. Take spatial data as an example, suppose each entity captures a particular location, and we can measure the distance between two locations. One way of modeling the similarity is to convert from the distance using Gaussian kernel, i.e., $a_{jj'} = \exp(-d^2(j, j')/\sigma^2)$. Here $d(j, j')$ is the 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 j and j'. It evaluates to 1 if the two input values are identical, and approaches 0 as they move further apart.

In many applications, if two entities are similar then they are likely to receive similar observations, e.g., two close locations are likely to get similar air quality readings. By incorporating such similarity relationships, the observations of an unobserved entity can be approximated by some combinations of the observations on similar entities. To achieve this, we add a regularization term **R1** to the objective function. The basic idea is that the observations from virtual users should not differ too much on entities that are similar to each other. Therefore, when $a_{jj'}$ is large (similar entities), $\|V_{j\cdot} - V_{j'\cdot}\|^2$ should be small, while some difference between $V_{j\cdot}$ and $V_{j'\cdot}$ can be tolerated when $a_{jj'}$ is small.

$$\begin{aligned}
\mathbf{R1}: \quad & \sum_{j=1}^{N}\sum_{j'=1}^{N} a_{jj'}\|V_{j\cdot} - V_{j'\cdot}\|^2 \\
= \; & \sum_{j=1}^{N}\sum_{j'=1}^{N}\left[a_{jj'}\sum_{k=1}^{K}(V_{jk} - V_{j'k})^2\right] \\
= \; & \sum_{k=1}^{K} V_{\cdot k}^{\mathrm{T}} L V_{\cdot k} \\
= \; & tr(V^{\mathrm{T}} L V).
\end{aligned} \tag{3}$$

To simplify the representations, we can reduce this term to the trace of $V^{\mathrm{T}}LV$ as shown above. Here, $L = D - A$ and D is a diagonal matrix with $D_{jj} = \sum_{j'} a_{jj'}$.

R2: Regularization on virtual users. As we have mentioned, virtual users' observations serve as a compression of the original data. However, virtual users are not equally important. Some virtual users may play more important roles when recovering X. To recognize this difference and rely on those important virtual users more during this process, we propose to add the following regularization term **R2**.

We first introduce another two sets of variables. We use $V^* \in \mathbb{R}^{N\times 1}$ to represent the aggregated observations from all virtual users. Let $W = Diag(w_1, w_2, \ldots, w_K)$ be the importance degrees of K virtual users. If a virtual user is more important (i.e., w_k is high), higher penalty will be received when this virtual user's observation is quite different from the aggregated one (i.e., difference between $V_{\cdot k}$ and V^* is large). On the contrary, the observation made by a less important virtual user with a low weight w_k is allowed to be different from the aggregated one. This term helps select virtual users that are more important, and thus is added to the sum of Eq(2) and Eq(3) to further regularize the matrix V.

$$\begin{aligned}
\mathbf{R2}: \quad & \sum_{j=1}^{N}\sum_{k=1}^{K}(v_{jk} - v_j^*)^2 w_k \\
= \; & tr\left[(V - V^*Q)W(V - V^*Q)^{\mathrm{T}}\right],
\end{aligned} \tag{4}$$

where $\sum_{k=1}^{K}\exp(-w_k) = 1$, $Q \in \mathbb{R}^{1\times K}$ and each entry $q_k = 1$ for $k = 1, \cdots, K$.

Combining Eq(2) (**MF**), (3) (**R1**) and (4) (**R2**), we obtain the following optimization problem:

$$\begin{aligned}\mathbf{P}: \min_{U,V,V^*,W} & f(U,V,V^*,W) = \|H \circ (X - UV^{\mathrm{T}})\|^2 + \alpha(\|U\|^2 + \|V\|^2) \\ & + \beta tr(V^{\mathrm{T}}LV) + \gamma\ tr\left[(V - V^*Q)W(V - V^*Q)^{\mathrm{T}}\right] \\ \text{s.t.} & \sum_{k=1}^{K} \exp(-w_k) = 1,\end{aligned}$$

where α, β and γ are hyperparameters which give different emphases on the regularization terms.

The problem **P** aims to minimize the aggregated error of three parts: 1) the first part **MF** describes how well the matrix factorization UV^T can recover the observation matrix X; 2) the second part **R1**, i.e., the regularization on entity similarity, represents the divergence among correlated entities; 3) the third part **R2**, i.e., the regularization on virtual users, represents the differences between virtual users' observations and the aggregated ones, weighted by the virtual users' weights.

3.2 Solution

In this section, we present our solution to the proposed optimization problem. The solution is developed based on block coordinate descent [5], which iteratively solves for one set of variables when fixing the others. The convergence of the method is discussed in Section 3.4.

Update V **while fixing** U, V^*, W. We rewrite the objective function f in problem **P** with respect to V:

$$\begin{aligned} f &= \|X - UV^{\mathrm{T}}\|^2 + \alpha\|V\|^2 + \beta tr(V^{\mathrm{T}}LV) \\ &\quad + \gamma\ tr\left[(V - V^*Q)W(V - V^*Q)^{\mathrm{T}}\right] + C_1 \\ &= \sum_{i=1}^{M}\sum_{j=1}^{N}(x_{ij} - U_{i\cdot}V_{j\cdot}^{\mathrm{T}})^2 + tr(V^{\mathrm{T}}(\alpha I + \beta L)V) \\ &\quad + \gamma\sum_{j=1}^{N}\sum_{k=1}^{K}(v_{jk} - v_j^*)^2 w_k + C_1, \end{aligned} \tag{5}$$

where C_1 is a constant independent of V. The partial derivative of f with respect to V is

$$\frac{\partial f}{\partial V} = -2X^{\mathrm{T}}U + 2VU^{\mathrm{T}}U + 2(\alpha I + \beta L)V + 2\gamma\ (V - V^*Q)W. \tag{6}$$

Let the partial derivative of Eq(6) be 0, we can get

$$V(U^{\mathrm{T}}U + \gamma\ W) + (\alpha I + \beta L)V = X^{\mathrm{T}}U + \gamma\ V^*QW. \tag{7}$$

The above is a Sylvester equation and a classical algorithm for the numerical solution is Bartels–Stewart algorithm [4] which requires $O(N^3)$ time complexity. It becomes intractable when we have a large number of entities, i.e., when N is large. An alternative way can be adopted to iteratively learn each column of V, i.e., $V_{\cdot k}(k = 1, \cdots, K)$, with other columns fixed. As will be discussed in Section 3.4, f is convex with respect to V, and this strategy preserves the same convergence property.

Let $f_1 = tr(V^{\mathrm{T}}(\alpha I + \beta L)V)$, and $f_2 = \sum_i\sum_j(x_{ij} - U_{i\cdot}V_{j\cdot}^{\mathrm{T}})^2 + \gamma\sum_j\sum_k(v_{jk} - v_j^*)^2 w_k$. It is obvious that $f = f_1 + f_2 + C_1$ and the following partial derivatives can be obtained:

$$\frac{\partial f_1}{\partial V_{\cdot k}} = 2(\alpha I + \beta L)V_{\cdot k}, \tag{8}$$

$$\frac{\partial f_2}{\partial v_{jk}} = -2\sum_i(x_{ij} - \sum_k u_{ik}v_{jk})u_{ik} + 2\gamma\ (v_{jk} - v_j^*)w_k. \tag{9}$$

Combining Eq(8) and Eq(9), and let the partial derivative of f with respect to $V_{\cdot k}$ be 0, we can derive the following linear equation:

$$F^{(k)}V_{\cdot k} = e^{(k)}, \tag{10}$$

where $F^{(k)} = (\alpha + \sum_{i=1}^{M} u_{ik}^2 + \gamma w_k)I + \beta L$, and $e^{(k)} = (e_1^{(k)}, e_2^{(k)}, \ldots, e_N^{(k)})^{\mathrm{T}}$ with $e_j^{(k)} = \sum_{i=1}^{M} u_{ik}(x_{ij} - V_{j\cdot}U_{i\cdot}^{\mathrm{T}} + v_{jk}u_{ik}) + \gamma\ v_j^* w_k$.

One simple solution involves computing the inversion of an $N \times N$ matrix $F^{(k)}$, and it still requires $O(N^3)$ computation. The steepest descent method [33, 45] can reduce it to linear time complexity and the corresponding update rule is:

$$\begin{aligned} r(t) &= e^{(k)} - F^{(k)}V_{\cdot k}(t), \\ \delta(t) &= \frac{r(t)^{\mathrm{T}}r(t)}{r(t)^{\mathrm{T}}F^{(k)}r(t)}, \\ V_{\cdot k}(t+1) &= V_{\cdot k}(t) + \delta(t)r(t), \end{aligned} \tag{11}$$

where (t) denotes the t_{th} iteration.

Update U **while fixing** V, V^*, W. The objective function f in problem **P** can be rewritten with respect to U:

$$\begin{aligned} f &= \|X - UV^{\mathrm{T}}\|^2 + \alpha\|U\|^2 + C_2 \\ &= tr(-2XVU^{\mathrm{T}} + UV^{\mathrm{T}}VU^{\mathrm{T}} + \alpha UU^{\mathrm{T}}) + C_3, \end{aligned}$$

where C_2 and C_3 are constants independent of U.

Let the derivative of f with respect to U be 0, we can derive the following:

$$U(V^{\mathrm{T}}V + \alpha I) = XV. \tag{12}$$

Then the solution of U can be derived as: $U = (V^{\mathrm{T}}V + \alpha I)^{-1}XV$. Equivalently, we can update $U_{i\cdot}$, i.e., each row of U, iteratively. Let the derivative of f with respect to $U_{i\cdot}$ be 0, we derive the following updating rule:

$$U_{i\cdot} = (\sum_{j=1}^{N} x_{ij}V_{j\cdot})(\sum_{j=1}^{N} V_{j\cdot}^{\mathrm{T}}V_{j\cdot} + \alpha I)^{-1}. \tag{13}$$

Since the size of the matrix involved in inversion is $K \times K$ and K is typically a small number, the computation complexity is acceptable.

Update W **while fixing** U, V, V^*. We derive the Lagrangian of the problem **P** with respect to W as follows:

$$\begin{aligned} L(\{w_k\}_{k=1}^K, \lambda) &= \gamma\sum_{j=1}^{N}\sum_{k=1}^{K}(v_{jk} - v_j^*)^2 w_k \\ &\quad + \lambda(\sum_{k=1}^{K}\exp(-w_k) - 1) + C_4, \end{aligned} \tag{14}$$

where C_4 is a constant independent of W. Let the partial derivative with respect to w_k be 0, we get:

$$\gamma\sum_{j=1}^{N}(v_{jk} - v_j^*)^2 = \lambda\exp(-w_k). \tag{15}$$

Algorithm 1 RST Framework

Input: Sparse observation matrix X, entity similarity matrix A, and the number of virtual users K.
Output: Entity truths X^*.

1: Initialize missing data of X as 0;
2: Initialize $w_k = -\log(1/K), k = 1, \cdots, K$;
3: Initialize $v_j^* = mean(X_{\cdot j})$, if entries in $X_{\cdot j}$ are not all missing; or $v_j^* = mean(H \circ X)$ for all non-missing values in X;
4: Initialize $u_{ik} = uniform(0,1), i = 1, \cdots, M$, and $k = 1, \cdots, K$;
5: $\hat{X} \leftarrow X$;
6: **repeat**
7: Update V according to Eq(11);
8: Update U according to Eq(13);
9: Update W according to Eq(17);
10: Update V^* according to Eq(18);
11: $\hat{X} \leftarrow UV^{\mathrm{T}}$;
12: $\hat{X} \leftarrow (1-H) \circ \hat{X} + H \circ X$;
13: **until** Convergence criterion is satisfied;
14: $X \leftarrow (1-H) \circ (U * V^{\mathrm{T}}) + H \circ X$;
15: Aggregate X to derive the entity truths X^* via a truth discovery method [32];
16: **return** X^*.

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

$$\lambda = \gamma \sum_{k=1}^{K} \sum_{j=1}^{N} (v_{jk} - v_j^*)^2. \tag{16}$$

We can then derive the update rule for each weight by plugging Eq(16) into Eq(15):

$$w_k = -\log\left(\frac{\sum_{j=1}^{N}(v_{jk} - v_j^*)^2}{\sum_{k=1}^{K}\sum_{j=1}^{N}(v_{jk} - v_j^*)^2}\right). \tag{17}$$

Update V^* while fixing U, V, W. The objective function of problem **P** with respect to V^* can be rewritten as

$$f = \gamma \sum_{j=1}^{N} \sum_{k=1}^{K} (v_{jk} - v_j^*)^2 w_k + C_5,$$

where C_5 is a constant independent of V^*. Let the partial derivative with respect to v_j^* be 0, we can derive the solution

$$v_j^* = \frac{\sum_{k=1}^{K} v_{jk} w_k}{\sum_{k=1}^{K} w_k}. \tag{18}$$

The procedure to solve the problem **P** is summarized in Lines 1 to 14 in Algorithm 1. The input includes observed entries in X (i.e., entries whose $h_{ij} = 1$), similarity matrix A and the number of virtual users K. After initializing all the variables (W, V^* and U), we iteratively update each set of variables using Eq(11), Eq(13), Eq(17) and Eq(18) respectively until convergence criterion is satisfied.

3.3 Aggregation on X

After filling in all the missing observations in X, we need to aggregate along each column to obtain the true value for each entity x_j^*. In order to achieve this, we adopt a state-of-the-art truth discovery method [32] which aims to find out the true information from noisy user observations. It formulates the problem as an optimization problem to minimize the overall weighted deviation between the identified truths and the input observations. Please refer to [32] for more details of this method. This step is included in Algorithm 1 (Line 15).

3.4 Convergence

In this section, we first present the convexity property of the objective function with respect to each set of variables, then prove the convergence of the problem **P**.

LEMMA 1. *f is convex with respect to V.*

PROOF. As we have discussed in Section 3.2, let $f_1 = tr(V^{\mathrm{T}}(\alpha I + \beta L)V)$, and $f_2 = \sum_i \sum_j (x_{ij} - U_{i\cdot}V_{j\cdot}^{\mathrm{T}})^2 + \gamma \sum_j \sum_k (v_{jk} - v_j^*)^2 w_k$. It is obvious that $f = f_1 + f_2 + C_1$ where C_1 is a constant independent of V.

We can rewrite $f_1 = \sum_{k=1}^{K} V_{\cdot k}(\alpha I + \beta L)V_{\cdot k}^T$. The Hessian of f_1 with respect to V is a block-diagonal matrix, i.e., $\frac{\partial^2 f_1}{\partial V \partial V^T} = diag(B_1, B_2, \ldots, B_K)$ where $B_k = \frac{\partial^2 f_1}{\partial V_{\cdot k} \partial V_{\cdot k}^T} = 2(\alpha I + \beta L)$. In addition, for any $K \times 1$ vector $r \neq 0$, we have $r^T L r = \sum_j \sum_{j'} a_{jj'}(r_j - r_{j'})^2 \geq 0$, i.e., the matrix L is positive semidefinite. Then we can conclude that $det(\frac{\partial^2 f_1}{\partial V \partial V^T}) > 0$, and thus f_1 is convex with respect to V.

As for f_2, it can be rewritten as $f_2 = \sum_i \sum_j (x_{ij}^2 - 2x_{ij}U_{i\cdot}V_{j\cdot}^T + V_{j\cdot}U_{i\cdot}^T U_{i\cdot}V_{j\cdot}^T) + \gamma \sum_j \sum_k (v_{jk} - v_j^*)^2 w_k$. The Hessian of f_2 with respect to V is also a block-diagonal matrix, i.e., $\frac{\partial^2 f_2}{\partial V \partial V^T} = diag(G_1, G_2, \ldots, G_N)$ where $G_n = \sum_i U_{i\cdot}^T U_{i\cdot} + \gamma W$. Since for any $N \times 1$ vector $b \neq 0$, $b^T(\sum_i U_{i\cdot}^T U_{i\cdot})b = \sum_i (U_{i\cdot}b)^2 \geq 0$ and for $k = 1, \ldots K$, $w_k \geq 0$ (because of Eq(17)), we can derive that $det(\frac{\partial^2 f_2}{\partial V \partial V^T}) \geq 0$, and thus f_2 is convex with respect to V.

In conclusion, f is convex with respect to V. □

LEMMA 2. *f is convex with respect to U.*

PROOF. As shown in Eq(12), the Hessian of f with respect to U is a block-diagonal matrix, i.e., $\frac{\partial^2 f}{\partial U \partial U^T} = diag(B_U, B_U, \ldots, B_U)$, and $B_U = \sum_j V_{j\cdot}^T V_{j\cdot} + \alpha I$. Similar to the proof of Lemma 1, we can derive that $det(\frac{\partial^2 f}{\partial U \partial U^T}) > 0$, and it is convex with respect to U. □

LEMMA 3. *f is convex with respect to W.*

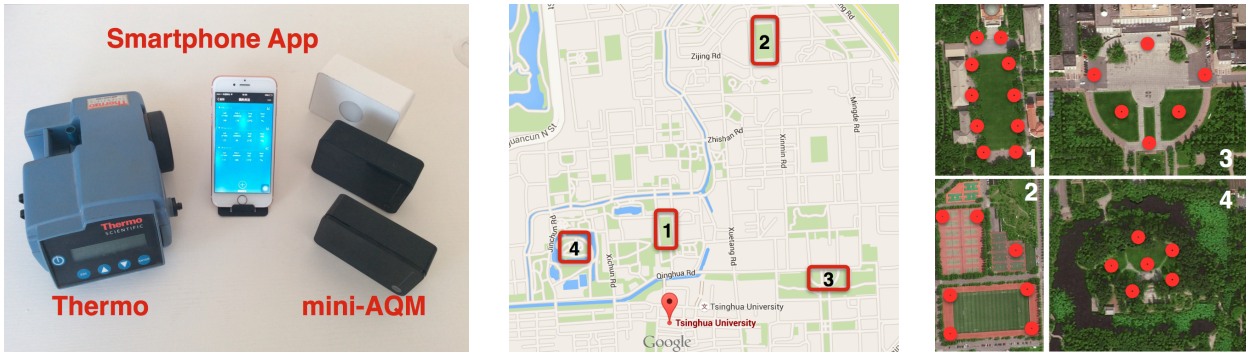
PROOF. In order to show the convexity in this case, we introduce another variable t_k, so that $t_k = exp(-w_k)$. Then the problem can be rewritten as follows,

$$\begin{aligned} \min_{\{t_k\}_{k=1}^K} \quad & f(t_k) = \sum_{j=1}^{N} \sum_{k=1}^{K} -\log(t_k)(v_{jk} - v_j^*)^2 + C \\ \text{s.t.} \quad & \sum_{k=1}^{K} t_k = 1, \end{aligned} \tag{19}$$

where C is a constant independent of W. The objective function in Eq(19) is a linear combination of negative logarithm functions and constants, and thus is convex. In addition, the constraint is linear in t_k, which is affine. Therefore, f is convex with respect to W. □

LEMMA 4. *f is convex with respect to V^*.*

PROOF. In this case, the objective function in the problem **P** is a summation of convex functions with respect to V^*. It is convex since these functions are convex and summation operation preserves convexity. □



(a) Air Quality Sensing Devices (b) Areas at Tsinghua University (c) Locations at Tsinghua University

Figure 2: Air quality sensing devices and locations at Tsinghua University. (a) shows the different versions of mini-AQM, its smartphone App and the Thermo. The mini-AQM is portable and carried by participants. The three versions are different in design and technology. 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.

THEOREM 5. *The iterative process shown in Algorithm 1 converges to a stationary point of the optimization problem* **P**.

PROOF. According to the property of the block coordinate descent [5], every limit point of the objective function in problem **P** is a stationary point under the condition that the objective function can attain a unique minimum during each iteration. According to the above lemmas, the objective function is convex with respect to each set of variables in each iteration, and thus a unique minimum is attained at each iteration. As the condition holds, this theorem holds. □

3.5 Computational Complexity

Assume there are M users, N entities, and K virtual users. It is also safe to assume that $K \ll M \ll N$ for most of the crowd sensing applications. Then, updating matrix V costs $O(KN)$ time for solving Eq(11) with the steepest descent method [45], plus $O(MNK^2)$ time for the matrix multiplication; updating matrix U costs $O(MNK)$ time for the matrix multiplication and $O(K^3)$ time for the inversion; updating W costs $O(NK)$ time according to Eq(17); and updating V^* also requires $O(NK)$ time according to Eq(18). In total, the time complexity is $O(MNK^2)$. Since K is usually a small number and thus can be regarded as a constant, the computational complexity of the proposed method is well within the feasible realm.

4. EVALUATIONS

In this section, we present the evaluation of the RST framework compared with several baseline methods. The experiment is conducted on real-world air quality sensing applications as well as simulation. The experiments on air quality sensing are conducted at Tsinghua University and the Haidian District in Beijing, China. The baselines and evaluation metrics are discussed in Section 4.1 and Section 4.2. Results are shown and discussed in Section 4.3, Section 4.4 and Section 4.5.

4.1 Baselines

As discussed in Section 3, the proposed Redundancy and Sparsity Tackling (RST) method first performs Matrix Factorization (MF) with regularization terms on entity SIMilarity (sim) and virtual user importance to fill in users' "missing" observations, then infer entity truths by aggregating observations with Truth Discovery (TD). As the proposed method captures both entity similarity and virtual users' importance, it gives good estimates on the "missing" observations, and thus the aggregation on the observation matrix can derive better estimates of entity truths. In this section, we justify the advantage of the proposed method empirically by comparing it with the following three baselines:

- MF_sim+TD (Matrix Factorization only with entity similarity regularization + Truth Discovery). Compared with RST, the difference is that this method does not consider the importance of different virtual users. In other words, this method does not incorporate the regularization term **R2** discussed in Section 3. The comparison with this alternative of the proposed method shows the benefits of adding **R2** into the optimization problem.

- TD+ITP (Truth Discovery + InTerPolation). As discussed in the introduction, one possible baseline is to tackle redundancy and sparsity challenges separately as follows: First, truth discovery is used to aggregate observations of non-missing entities to estimate their true values. Second, interpolation is conducted on missing entities to infer their values. Specifically, via interpolation, a missing entity's value is computed as the weighted average of its correlated observed entities. As truth discovery and interpolation are not integrated, they may not perform well compared with RST. Especially when the data is extremely sparse, the interpolation is likely to fail.

- Mean+ITP (Mean + InTerPolation). With this method, mean (or average) is performed first on observations of non-missing entities to estimate their true values, then an interpolation is conducted on missing entities to infer their values. This is the simplest approach to tackle redundancy and sparsity issues, but it ignores the characteristics of crowd sensing applications and thus is not as good as RST.

Table 2: Performance over All Entities on Tsinghua Data with Varying Sparsity

			All Entities							
			MAE				RMSE			
S_e	S_u	S	RST	MF_sim+TD	TD+ITP	Mean+ITP	RST	MF_sim+TD	TD+ITP	Mean+ITP
	0.3	0.72	**16.764**	30.059	20.749	21.430	**28.432**	55.969	36.525	36.603
0.6	0.6	0.84	**16.981**	36.842	21.298	22.628	**28.581**	70.692	36.370	37.780
	0.9	0.96	**19.271**	59.421	24.632	23.851	**32.540**	107.274	40.992	39.523
	0.3	0.79	**17.719**	31.053	22.683	23.289	**29.827**	57.632	39.331	39.091
0.7	0.6	0.88	**17.921**	39.848	22.736	22.232	**30.008**	74.779	39.020	36.233
	0.9	0.97	**21.429**	62.156	23.940	22.668	36.958	112.728	38.672	**35.576**
	0.3	0.86	**16.587**	30.991	21.000	21.648	**28.164**	57.167	36.923	36.836
0.8	0.6	0.88	**17.547**	38.636	22.173	22.409	**29.347**	71.168	37.572	37.108
	0.9	0.98	**21.142**	67.214	26.060	25.841	42.281	121.999	42.948	**41.847**

Table 3: Performance over Non-missing Entities on Tsinghua Data with Varying Sparsity

			Non-missing Entities							
			MAE				RMSE			
S_e	S_u	S	RST	MF_sim+TD	TD	Mean	RST	MF_sim+TD	TD	Mean
	0.3	0.72	**15.780**	27.097	20.232	21.120	**27.204**	51.174	35.246	35.409
0.6	0.6	0.84	**15.507**	30.117	20.724	22.193	**26.622**	60.859	35.493	37.187
	0.9	0.96	**17.635**	51.044	26.015	24.730	**30.251**	98.369	42.447	40.562
	0.3	0.79	**15.652**	29.987	21.854	22.751	**27.969**	56.119	38.431	38.336
0.7	0.6	0.88	**18.131**	43.363	23.290	23.151	**29.858**	78.140	39.427	37.074
	0.9	0.97	**18.751**	56.584	22.814	21.473	**33.400**	105.294	37.933	34.961
	0.3	0.86	**16.831**	32.925	22.951	23.800	**28.573**	59.214	38.573	38.439
0.8	0.6	0.88	**19.594**	44.072	25.484	25.782	**31.543**	76.154	41.343	40.939
	0.9	0.98	**22.364**	63.201	26.350	25.558	**39.894**	115.851	43.300	42.255

4.2 Evaluation Metrics

For the proposed method and the baseline methods, the input includes observations about entities given by different users as well as the entity similarity. The output is the estimated entity truths. In each sensing scenario, 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. In the experimented sensing applications, the sensor data are continuous, and thus we adopt the following measures to evaluate the performance.

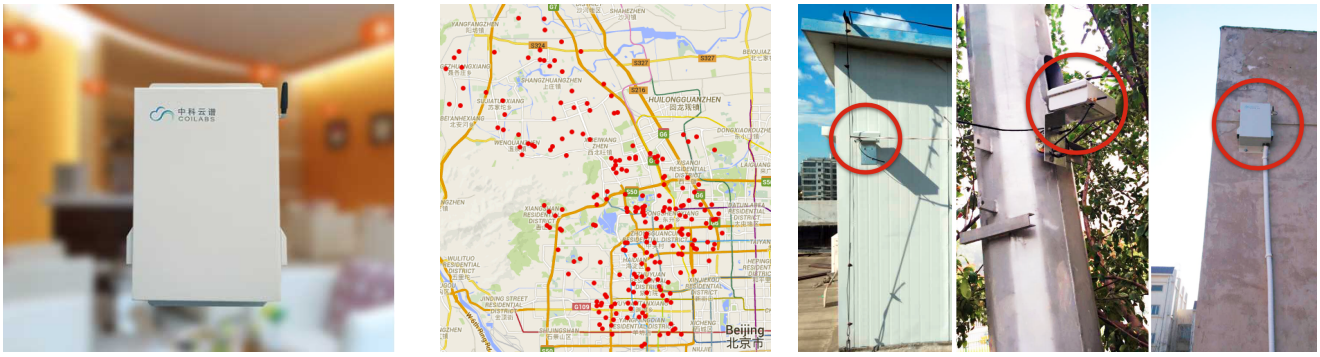
- *Mean Absolute Error* (*MAE*) measures the overall absolute error between each method's outputs and the ground truths, which is computed by averaging the absolute difference over all the entities.
- *Root Mean Square Error* (*RMSE*) is computed by taking the root of the mean squared differences between each method's outputs and the ground truths.

MAE and *RMSE* both measure the differences between outputs and 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 *MAE*. We perform all the experiments 40 times and report the average results.

4.3 Air Quality Sensing at Tsinghua University

Air quality has become a great concern around the world, especially for developing countries such as China and India where people are suffering from the deteriorated air quality. Although official monitoring stations with high-quality sensors are deployed throughout the country, the number is very limited since they are very expensive. Take Beijing (the capital of China) as an example, it only has 22 stations covering a 50 $km \times 50$ km land, and this means each station covers an area as large as 113 km^2 [54]. With such a limited number of monitoring stations, we may obtain the overall air quality condition of the whole city, but we are unable to obtain fine-grained air quality measurements. Fortunately, crowd sensing becomes a promising solution for the fine-grained air quality monitoring task.

In this section, we discuss the experiment on sensing the particulate matter with diameter less than 2.5 micron (PM2.5) with mini-AQM [9]. Mini-AQM is a portable air quality device for personal use. It is designed and manufactured by the Coilabs Co. Ltd [1]. The device is shown in Figure 2(a). Mini-AQM can automatically sense its surrounding environment's PM2.5 value, and upload the data to the server. The value can also be viewed via a smartphone App. We recruited 18 participants and let them conduct sensing tasks with mini-AQM in four designated areas at Tsinghua University, Beijing, China. There are around seven locations within each area, and the PM2.5 value at each location is regarded as an entity. The entities are considered to be similar to each other if the geographical distances among them are close. In particular, the similarities among entities are set to be 1 if they reside within the same area, and 0 if they are from different areas. The ground truths are collected with Thermo [2] which is an accurate but expensive sensing device. Several experiments are performed, and the results are discussed in the following. Please note that, the proposed method aims to tackle the redundancy and sparsity



(a) Air Atlas DT Pro (b) Deployment in Haidian District (c) Real Deployments

Figure 3: Air Atlas DT Pro and the deployment locations in Haidian district. (a) shows the Air Atlas DT Pro device. (b) shows the locations for deployment in Haidian District. (c) shows several examples in real deployment.

Table 4: Performance on Haidian Data with Varying Number of Entities

Sparsity	#Entities	MAE			
		RST	MF_sim+TD	TD+ITP	Mean+ITP
$S_e = 0.8$ $S_u = 0.5$ $S = 0.9$	20	**10.664**	11.783	12.703	13.211
	40	**8.098**	8.423	11.978	12.580
	60	**7.690**	7.759	11.436	12.572
	80	**7.836**	7.909	10.462	11.295
	98	**7.706**	7.762	10.320	11.151
$S_e = 0.8$ $S_u = 0.8$ $S = 0.96$	20	**10.876**	13.940	19.719	18.690
	40	**8.997**	9.674	19.080	18.497
	60	**8.678**	8.936	19.108	18.899
	80	**8.624**	8.889	17.876	17.724
	98	**8.445**	8.677	16.845	17.266

problems for general crowd sensing applications including not only the air quality monitoring but also many others such as road congestion detection, gas price estimation, etc. Thus, we do not take any application-specific factors into consideration, such as wind speed and weather condition.

Performance comparison when varying the sparsity levels. In different crowd sensing applications, we may encounter various data sparsity levels, and thus we test how the performance varies with respect to sparsity levels. We denote the overall sparsity of the data as S, and it consists of two parts – the sparsity with respect to entities (S_e) and the sparsity with respect to users (S_u). Specifically, S_e describes the percentage of "missing" entities, i.e., the entities that receive no observations. For those "non-missing" entities, it is probable that they are only observed by some of the users. Then the sparsity with respect to users S_u indicates the percentage of users who do not report information for these entities. It is easy to derive that $S = S_e + (1 - S_e) \times S_u$. In this experiment, we vary S_e among $(0.6, 0.7, 0.8)$ and vary S_u among $(0.3, 0.6, 0.9)$. The corresponding overall sparsity S and the evaluation results are shown in Tables 2 and 3. Table 2 shows the performance over all the entities, and Table 3 shows the performance over non-missing entities (i.e., the entities that receive user observations).

There are 29 entities and 18 users in this experiment. As shown in Table 2, we can see that RST performs better than the other baselines under most of the sparsity settings because the corresponding MAE and RMSE are the lowest. This result demonstrates the advantages of RST. The proposed method solves redundancy and sparsity challenges by tightly integrating the missing observation estimation and the truth discovery, and the matrix factorization and regularization terms in the missing observation estimation enable a better estimation of the observation matrix, which is used to derive the entity truths. Therefore, the proposed RST method is able to outperform the baselines. We can also see that all the methods' performance deteriorates as the sparsity level increases. This is because more information would be missing when the sparsity level is higher, and it makes the estimation of true values harder. However, the proposed method still outperforms others in most situations. From Table 3, it can be seen that RST performs better on non-missing entities. The reason is that the proposed method can capture users' behavior better when the users provide some observations.

4.4 Air Quality Sensing in Haidian District

As we have discussed, air quality monitoring stations are expensive to build and thus are very limited in number. In order to provide a fine-grained air quality monitoring service, Coilabs also designed another air quality sensing device, "Air Atlas DT Pro" (Figure 3), that can be deployed in outdoor environment. Over one hundred such devices are deployed across the Haidian District in Beijing, and the PM2.5 values are monitored continuously to provide a timely and fine-grained report. We acquired all the sensed data at 03:00 on March 20, 2015, and regard them as the ground truth values at those locations. The PM2.5 value at each location is treated as an entity, and there are totally 98 entities after removing outliers. The similarity between any two entities is calculated with Gaussian kernel based on their geographical distance. The observations of users with different reliability levels are generated by adding Gaussian noise upon the ground truths. In this experiment, we have six groups of users and the users in the same group have the same noise level selected from (-50%, -30%, -10%, 10%, 30% and 50%). The noise level measures to what extent their observations deviate from the ground truth. In total we have 30 users, and thus there are 5 users in each group. Experimental results are summarized in Table 4, and discussions can be found in the following.

Performance comparison when varying the sparsity levels. In Table 4, we show the performance on two sparsity

Table 5: Performance over All Entities on Simulation with Varying Sparsity

			All Entities							
			MAE				RMSE			
S_e	S_u	S	RST	MF_sim+TD	TD+ITP	Mean+ITP	RST	MF_sim+TD	TD+ITP	Mean+ITP
	0.3	0.72	**5.158**	5.394	5.789	5.958	**8.105**	8.166	8.325	8.415
0.6	0.6	0.84	**5.157**	6.111	6.274	6.576	**8.135**	8.655	8.618	8.869
	0.9	0.96	**6.037**	13.311	9.361	9.753	**8.285**	17.240	12.157	12.665
	0.3	0.79	**6.180**	6.409	6.641	6.776	**9.120**	9.252	9.225	9.301
0.7	0.6	0.88	**6.340**	7.166	7.392	7.665	**9.292**	9.831	9.952	10.217
	0.9	0.97	**6.789**	17.046	10.043	10.266	**9.245**	22.268	13.035	13.259
	0.3	0.86	**7.197**	7.586	7.502	7.613	**10.129**	10.525	10.202	10.272
0.8	0.6	0.92	**7.210**	8.013	8.162	8.390	**10.023**	10.408	10.717	10.931
	0.9	0.98	**9.504**	24.892	10.530	10.604	**12.707**	31.561	13.543	13.695

Table 6: Performance over Non-missing Entities on Simulation with Varying Sparsity

			Non-missing Entities							
			MAE				RMSE			
S_e	S_u	S	RST	MF_sim+TD	TD	Mean	RST	MF_sim+TD	TD	Mean
	0.3	0.72	**0.187**	0.670	1.651	2.034	**0.239**	0.853	2.167	2.675
0.6	0.6	0.84	**0.317**	2.275	2.955	3.570	**0.409**	2.983	3.998	4.795
	0.9	0.96	**2.626**	11.156	8.869	9.489	**3.446**	14.492	11.969	12.738
	0.3	0.79	**0.198**	0.681	1.620	1.988	**0.257**	0.873	2.141	2.627
0.7	0.6	0.88	**0.351**	2.361	3.092	3.688	**0.447**	3.058	4.077	4.851
	0.9	0.97	**2.763**	13.858	9.281	9.708	**3.770**	18.216	12.509	12.958
	0.3	0.86	**0.218**	0.715	1.575	1.928	**0.285**	0.908	2.079	2.527
0.8	0.6	0.92	**0.358**	3.416	3.149	3.690	**0.466**	4.470	4.117	4.777
	0.9	0.98	**4.306**	19.832	9.058	9.292	**5.969**	25.615	12.210	12.473

levels, i.e., 90% and 96%. Compared with the experiment performed on the Tsinghua data, similar patterns can be observed when varying the sparsity levels. RST outperforms the baselines, especially when the sparsity level is higher. This demonstrates the effectiveness of the proposed integrated framework that combines missing observation estimation and truth discovery to derive entity truths.

Performance comparison when varying the number of entities. To check the performance on the scenarios with different number of entities, we uniformly sample the entities and conduct the experiments. As can be observed in Table 4, the performance of RST is consistently better than other baselines under any setting. In addition, as shown in Table 4, the performance improvement of RST is more significant when the information density is low which is caused by less entities. When we have more entities whose values are distributed over an interval, the simple interpolation method can give a fair estimation of entity truths. On the contrary, when there are only a few entities having similar values, the interpolation method fails. But the proposed method which makes better use of entity similarity and selects important virtual users, outperforms the other methods. Moreover, when we fix the number of users, less entities means less observations collected from users. This demonstrates that RST can better estimate users' observations, even when we have very limited information.

4.5 Simulation Results

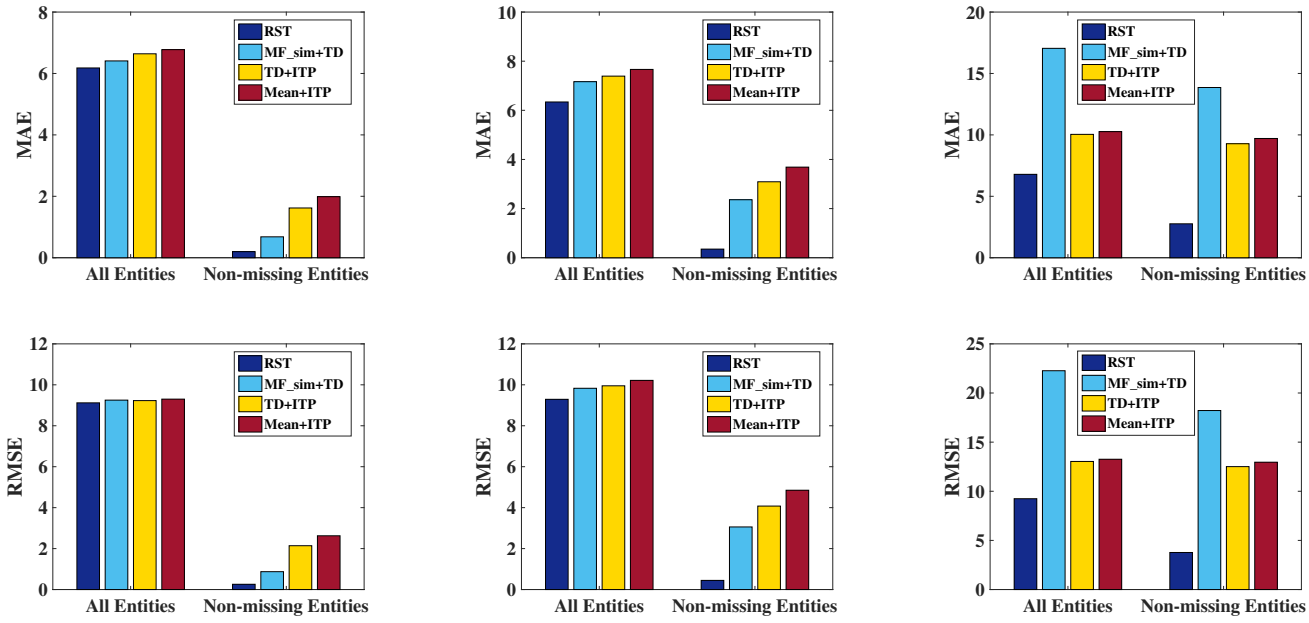
In order to further examine the performance of RST on dealing with redundant and sparse data, we conduct simulation studies. The benefit of simulation is that we have the full control over the data, i.e., we can alter the sparsity, the number of entities and the number of users, such that the performance under all conditions can be exposed.

Table 7: Performance on Simulation with Varying Number of Entities

Sparsity	#Entities	MAE			
		RST	MF_sim+TD	TD+ITP	Mean+ITP
$S_e = 0.8$ $S_u = 0.5$ $S = 0.9$	40	**6.247**	6.728	7.319	7.499
	60	**6.847**	7.386	7.689	7.891
	80	**7.066**	7.590	7.853	8.048
	100	**7.206**	7.685	7.832	7.989
$S_e = 0.8$ $S_u = 0.8$ $S = 0.96$	40	**6.113**	14.114	9.143	9.308
	60	**6.593**	12.963	8.946	9.154
	80	**7.132**	12.888	9.271	9.529
	100	**7.449**	12.087	9.079	9.275

In this experiment, we simulate a crowd sensing scenario with 10 areas, each of which contains several entities. We assume that the entities within the same area are similar with each other, i.e., the similarity degree between any two is 1. First, we generate the ground truths for each area and each entity. The true values for these areas are $40, 50, \cdots, 130$, and the true values for the entities within each area are generated by adding a relatively small Gaussian noise into the area truths (the noise is set as 10 in our experiment). Then we generate 6 groups of users, and assign certain amount of users to each group. The users in one group have the same noise level selected from (-50%, -30%, -10%, 10%, 30% and 50%). Finally, the users' observations are generated by adding their corresponding Gaussian noise to the ground truths of entities.

Performance comparison when varying sparsity levels and number of entities. Firstly, we vary the sparsity



(a) Sparsity=0.79 (b) Sparsity=0.88 (c) Sparsity=0.97

Figure 4: Performance Comparison on Simulation

Table 8: Performance on Simulation with Varying Number of Users

Sparsity	#Users	MAE				RMSE			
		RST	MF_sim+TD	TD+ITP	Mean+ITP	RST	MF_sim+TD	TD+ITP	Mean+ITP
$S_e = 0.8$ $S_u = 0.5$ $S = 0.9$	30	**6.790**	7.612	8.288	8.610	**9.794**	10.328	10.883	11.196
	60	**6.788**	7.297	7.706	7.897	**9.924**	10.290	10.402	10.564
	120	**6.876**	7.202	7.500	7.630	**9.890**	10.029	10.117	10.206
	180	**6.909**	7.221	7.413	7.491	**9.899**	10.031	10.158	10.198
$S_e = 0.8$ $S_u = 0.8$ $S = 0.96$	30	**6.835**	17.758	11.372	11.714	**9.404**	22.347	14.718	15.205
	60	**6.859**	14.050	9.468	9.658	**9.809**	18.394	12.265	12.500
	120	**6.238**	11.418	7.707	7.803	**9.082**	15.135	10.145	10.215
	180	**6.196**	12.129	7.230	7.315	**9.130**	15.956	9.629	9.688

levels with 60 users (10 users per group) and 100 entities (10 entities per area). The results are summarized in Tables 5 and 6 measured by MAE and RMSE. To inspect the results visually, we also show the results in Figure 4 on three sparsity levels. We find that RST outperforms other baselines on all the sparsity levels, and the performance improvement is better when the sparsity level is higher. In addition, the estimation error is low on the non-missing entities as shown in Table 6 and Figure 4. In another experiment, we vary the number of entities per group as 4, 6, 8, 10 while fixing the number of users to be 60 (10 users per group) and the sparsity levels as 90% and 96%. The results are shown in Table 7. Similar trends can be found as in the previous experiment in Haidian District. We also observe the clear advantage of the proposed method when the entity density becomes low.

Performance comparison when varying number of users. In this experiment, we compare the performance by changing the number of users from 30 to 180, while the number of entities is fixed as 50 and the sparsity is set to be 90% and 96%. The results are summarized in Table 8. As can be seen, the performance of RST is the best among all methods. Especially when there are less number of users, the proposed method performs even better compared with other baselines. When we fix the number of entities, less number of users indicates less observations that we can collect from users. The proposed method performs better in such cases because it can better capture user behavior compared with baselines. Since RST can distinguish the importance of virtual users and make a better use of the entity similarity information, it has good performance even when the number of users is small.

5. RELATED WORK

There are three research fields that are related to this work, and we summarize them in this section.

Crowd Sensing. The research of crowd sensing [17, 27, 30] has attracted significant attention thanks to the proliferation of smart devices. With the aid of integrated or portable sensors, such as accelerometer, gyroscope, GPS, and microphone, now individuals can sense, record and share the information of their surrounding environment whenever they want. Thus far, a large variety of crowd sensing systems have been developed, and their crowd-contributed information has brought significant benefits to the society. For ex-

ample, some systems are built to improve our travel experience by providing various services such as travel time estimation [6, 49], fuel-efficient navigation [16, 43] and traffic regulators detection [23]. Systems for better cycling experience are also developed [14] and make it easier to document and share the routes, ride statistics, weather conditions and etc. Huang et al. [24] implemented a system for the search and rescue of people in emergent situations in wilderness areas. Chen et al. [8] proposed a method for the reconstruction of building interior. Chon et al. [10, 11] examined the characteristics of place-centric crowd sensing systems. In [13,25], the data trustworthiness of crowd sensing systems was investigated. In addition, the energy efficiency of the crowd sensing systems is also a hot topic and some work [21,22,29] were conducted towards this end. These aforementioned papers focus on the design and implementation of the sensing systems, and none of them handles the redundant and sparse data in crowd sensing systems.

There are some existing crowdsourced air quality monitoring systems. In [19, 26], portable devices are designed and implemented to detect gasses like CO_2 and O_3. Unfortunately, these works either do not consider data redundancy problem or use simple averaging method to solve it, and more importantly, none of them can address the challenge of data sparsity. Some other air quality monitoring systems [9, 20] are developed to address the data sparsity problem. However, they all require significant context information, such as land-use, humidity, temperature, POI and etc. In contrast, our goal is to design a general redundancy and sparsity tackling framework that can benefit a wide spectrum of crowd sensing applications. For this reason, our proposed framework does not require any application-specific information.

Matrix Factorization. Matrix factorization method [3, 7, 18, 28, 33, 37] factorizes a target matrix into the product of multiple matrices. Such a method can be used in matrix completion to estimate the missing values. The first step of the proposed RST method is based on matrix factorization method, but we achieve more: By incorporating the entity similarity regularization, we can handle the situations when a whole column of the matrix is missing; With the virtual user regularization, we select virtual users with high importance, such that it can further improve the estimation accuracy on the observation matrix. In addition, context-aware tensor decomposition is adopted in [44,49,55] to estimate entity values such as the noise levels and vehicle travel time in big cities. However, these methods only tackle the data sparsity problem without taking into consideration the redundancy problem.

Truth Discovery. In many real-world applications, different sources may provide conflicting information on the same entity, and thus how to discover the true information (i.e., the truth) among these conflicting observations becomes a key question. Truth discovery methods [34] try to solve this problem by inferring both the source reliability and entity truths. These methods can be roughly divided into three categories [34]: iterative methods [12, 15, 40, 51], probabilistic model based methods [41,47,48,52,53], and optimization-based methods [31, 32, 35, 46]. Although they differ in the specific formulations, these methods share a common principle: A source which provides many pieces of true information is likely to be reliable, and a piece of information stated by many reliable sources is likely to be true. The truth discovery problem is formulated based on this principle, and it typically leads to an iterative solution which iteratively updates source reliability and entity truths. Recently, Miao et al. [38] proposed a cloud-enabled truth discovery framework which provides privacy guarantees in crowdsourced data aggregation. Meng et al. [36] incorporated the entity correlation information to further improve the performance of truth discovery. These truth discovery methods can aggregate crowd-contributed data to resolve the redundancy challenge. However, they cannot tackle the sparsity challenge. Wen et al. [50] propose a framework to estimate the accuracy of sensor measurements. In this framework, the Gaussian process is applied first to interpolate the sensor data over time and space, then state and accuracy estimations are performed on the sensor data. However, this framework is not suitable for the problem studied in this paper, because the Gaussian process cannot make good use of the prior knowledge of entity correlations, and may not be able to deal with highly sparse and large scale data due to high computational complexity.

6. CONCLUSIONS

Redundancy and Sparsity are two major challenges daunting the crowd sensing applications. The data are redundant because multiple users may provide conflicting observations on the same entity, and they are sparse since the users are oftentimes outnumbered by the entities to be sensed. In this paper, we propose an integrated framework to infer the true values of entities from redundant and sparse data in crowd sensing applications. We design an effective optimization method that estimates users' missing observations, which consists of matrix factorization and regularization terms. Via matrix factorization, we can represent the observations of the original users as a linear combination of the virtual users' values. Then a regularization term on entity similarity is added to the objective function to enable the estimation on entities with no observations. Another regularization term on virtual users is also incorporated, such that we can distinguish those virtual users that play more important roles when recovering the observation matrix. After missing observations are estimated, a truth discovery process is conducted to finally infer the true value for each entity. Estimating user observations helps the estimation of user reliability degrees in truth discovery, and thus the redundancy and sparsity challenges are tackled jointly in the proposed framework. We conduct experiments on the task of air quality sensing at Tsinghua University and Haidian District, together with a thorough simulation. Results demonstrate the ability of the proposed method in estimating true entity values in various sensing scenarios, and show its advantage compared with baseline methods.

7. ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers and our shepherd, Nicholas Lane, 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 CNS-1566374. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agency.

8. REFERENCES

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

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

[3] J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert. Low-rank matrix factorization with attributes. *Computing Research Repository*, 2006.

[4] R. H. Bartels and G. Stewart. Solution of the matrix equation ax+ xb= c [f4]. *Communications of the ACM*, 15(9):820–826, 1972.

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

[6] 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 International Conference on Embedded Networked Sensor Systems (SenSys'11)*, 2011.

[7] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[8] 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)*, 2015.

[9] 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)*, 2014.

[10] 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 ACM international Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'13)*, pages 3–12, 2013.

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

[12] X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: the role of source dependence. *Proceedings of the VLDB Endowment*, 2(1):550–561, 2009.

[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)*, 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 *Proceedings 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, 2010.

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

[18] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[19] D. Hasenfratz, O. Saukh, S. Sturzenegger, and L. Thiele. Participatory air pollution monitoring using smartphones. *Mobile Sensing*, pages 1–5, 2012.

[20] D. Hasenfratz, O. Saukh, C. Walser, C. Hueglin, M. Fierz, and L. Thiele. Pushing the spatio-temporal resolution limit of urban air pollution maps. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom'14)*, pages 69–77, 2014.

[21] 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 *Proceedings of the 33rd IEEE International Conference on Computer Communications (INFOCOM'14)*, 2014.

[22] 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 *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'15)*, 2015.

[23] 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.

[24] 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, 2005.

[25] 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, 2010.

[26] Y. Jiang, K. Li, L. Tian, R. Piedrahita, X. Yun, O. Mansata, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang. Maqs: a personalized mobile sensing system for indoor air quality monitoring. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'11)*, pages 271–280, 2011.

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

[28] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[29] 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)*, pages 7:1–7:14, 2013.

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

[31] 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. *Proceedings of the VLDB Endowment*, 8(4):425–436, 2014.

[32] 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 *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'14)*, pages 1187–1198, 2014.

[33] W.-J. Li and D.-Y. Yeung. Relation regularized matrix factorization. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'09)*, 2009.

[34] Y. Li, J. Gao, C. Meng, Q. Li, L. Su, B. Zhao, W. Fan, and J. Han. A survey on truth discovery. *SIGKDD Explorations*, 17(2):1–16, 2015.

[35] 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, 2015.

[36] C. Meng, W. Jiang, Y. Li, J. Gao, L. Su, H. Ding, and Y. Cheng. Truth discovery on crowd sensing of correlated entities. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys'15)*, pages 169–182, 2015.

[37] C. D. Meyer. *Matrix analysis and applied linear algebra*, volume 2. Siam, 2000.

[38] 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)*, 2015.

[39] 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, 2009.

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

[41] J. Pasternack and D. Roth. Latent credibility analysis. In *Proceedings of the 22nd international conference on World Wide Web (WWW'13)*, pages 1009–1020, 2013.

[42] 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, 2010.

[43] 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 (TMC)*, 2015.

[44] J. Shang, Y. Zheng, W. Tong, E. Chang, and Y. Yu. Inferring gas consumption and pollution emission of vehicles throughout a city. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14)*, pages 1027–1036, 2014.

[45] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain, 1994.

[46] 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, 2014.

[47] 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*, pages 202–213. ACM, 2015.

[48] 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, 2014.

[49] Y. Wang, Y. Zheng, and Y. Xue. Travel time estimation of a path using sparse trajectories. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14)*, pages 25–34, 2014.

[50] H. Wen, Z. Xiao, A. Markham, and N. Trigoni. Accuracy estimation for sensor systems. *IEEE Transactions on Mobile Computing*, 14(7):1330–1343, 2015.

[51] 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.

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

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

[54] Y. Zheng, F. Liu, and H.-P. Hsieh. U-air: When urban air quality inference meets big data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13)*, pages 1436–1444, 2013.

[55] Y. Zheng, T. Liu, Y. Wang, Y. Zhu, Y. Liu, and E. Chang. Diagnosing new york city's noises with ubiquitous data. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'14)*, pages 715–725, 2014.