Generalized Decision Aggregation in Distributed Sensing Systems

Lu Su*, Qi Li*, Shaohan Hu†, Shiguang Wang†, Jing Gao*, Hengchang Liu‡,
Tarek F. Abdelzaher†, Jiawei Han†, Xue Liu§, Yan Gao¶, Lance Kaplan‖
*State University of New York at Buffalo, †University of Illinois at Urbana-Champaign, §McGill University,
‡University of Science and Technology of China, ¶Accenture Technology Labs, ‖US Army Research Laboratory
Email: {lusu, qli22, jing}@buffalo.edu, {shu17, swang83, zaher, hanj}@illinois.edu,
hcliu@ustc.edu.cn, xueliu@cs.mcgill.ca, y.b.gao@accenture.com, lance.m.kaplan@us.army.mil

***Abstract*—In this paper, we present GDA, a generalized decision aggregation framework that integrates information from distributed sensor nodes for decision making in a resource efficient manner. Traditional approaches that target similar problems only take as input the discrete label information from individual sensors that observe the same events. Different from them, our proposed GDA framework is able to take advantage of the confidence information of each sensor about its decision, and thus achieves higher decision accuracy. Targeting generalized problem domains, our framework can naturally handle the scenarios where different sensor nodes observe different sets of events whose numbers of possible classes may also be different. GDA also makes no assumption about the availability level of ground truth label information, while being able to take advantage of any if present. For these reasons, our approach can be applied to a much broader spectrum of sensing scenarios. The advantages of our proposed framework are demonstrated through both theoretic analysis and extensive experiments.**

***Keywords*-Decision Aggregation; Distributed Sensing System; Participatory Sensing; Crowd Sensing; Social Sensing**

I. Introduction

The proliferation of embedded sensing devices in the recent years has given rise to the fast development and wide deployment of distributed sensing systems. Designed to connect people and the physical world in ways previously unimaginable, such systems have become an integral part of people's everyday lives, hosting a whole spectrum of civilian and military applications, providing useful information to help people with decision making about events in the physical world.

One of the most widely used techniques for decision making is classification [1], [2], which is the task of assigning objects (data) to one of several predefined categories (classes). Its basic idea is to "learn" a function (also called classifier) from a set of training data, in which each object has feature values and a class label, and use the learned function to determine the class labels for newly-arrived data.

The major challenge in applying classification techniques to solve decision making problems in distributed sensing systems lies in the *trade-off between decision accuracy and resource consumption*. On one hand, individual sensors are not reliable, due to various reasons such as incomplete observation, environment and circuit board noise, poor sensor quality, lack of sensor calibration, or even deceptive intent in the first place. To address this sensor reliability problem, one common approach is to integrate information from multiple sensors, as this will likely cancel out the errors of individual sensors and improve decision accuracy. On the other hand, distributed sensing systems usually have limited resources (energy, bandwidth, storage, time, money, or even human labor). Thus, it is often prohibitive to collect data from a large number of sensors due to the potential excessive resource consumption. Therefore, it is challenging to solve the *decision aggregation* problem, that is, to collect and integrate information from distributed sensors to reach a final decision in a resource efficient manner.

Recent efforts have been made to address this challenge under different sensing scenarios. Representative examples include decision aggregation approaches [3], [4] designed for remotely deployed sensing systems where unattended sensor nodes forward their findings through wireless ad hoc networks, and truth discovery schemes [5]–[7] that target on social and crowd sensing applications where people themselves act as "sensors" and share their observations via social networks. Both strategies dictate that each individual node report only its classification result (*decision*) as opposed to raw data, thus minimizing network transmission and leading to significant saving of system resources. Individual decisions, upon arrival at the server, are further combined to produce the final decision.

Though yielding reasonably good performance in certain cases, these approaches suffer from a major limitation, that is, they only take as input discrete decision information. Sometimes individual sensors may not be quite confident about their decisions due to various reasons, such as incomplete or noisy observations. In this case, if each sensor's confidence information (the probability that it "believes" the observed event belongs to each candidate class) can also be taken into consideration, we should be able to further improve the final decision accuracy. For example, suppose a vehicle is observed by three sensors that try to determine whether it is a tank or a truck (assuming it is actually a tank). Each sensor then provides a confidence probability vector corresponding to its belief in the vehicle being a tank or a truck. Suppose the three probability vectors are (0.99,

1052-8725/14 $31.00 © 2014 IEEE
DOI 10.1109/RTSS.2014.40

0.01), (0.49, 0.51), and (0.49, 0.51). In this case, only one sensor predicts the vehicle to be a tank, however, deciding so with high confidence, as opposed to the other two that both vote for a truck but with confidences not far from that of random guess, as reflected by their decision probability vectors. Therefore, we expect a reasonable decision aggregation scheme to output tank as the aggregated decision, as opposed to the traditional decision aggregation or truth discovery schemes that only takes discrete decision labels as input, which would likely get the wrong answer, favoring the majority though incorrect decision in our example.

In practice, the decision probability vectors can be obtained from both traditional device-centric sensing systems where sensor nodes conduct explicit classification computations, and newly-emerged people-centric sensing paradigms where people conduct implicit classifications through their logical reasoning. For example, when working with hardware sensor nodes, we can adopt classification algorithms that derive decision probabilities through heuristic metrics like the distance between the observed data and the decision boundary learned from training data; When having people carry out sensing tasks, we can ask each participant to explicitly provide the confidence level of each decision made.

The goal of this paper is to develop a *generalized decision aggregation* (GDA) framework for distributed sensing systems that can address the above challenge, by taking as input each individual sensor's decision probability vectors and computing the aggregated decision (class label) for all events under observation. In pursuing the generalizability so that it is applicable to a full range of sensing scenarios, our proposed GDA framework bears the following properties.

1) Each individual sensor's reliability level is explicitly accounted for when GDA integrates individual decisions. A sensor's reliability information is important as it reflects the general quality of information it can provide. The aggregated decision should favor more reliable sensors and weigh less unreliable ones instead of treating all individuals equally. In reality, however, the reliability information is usually unknown a priori. To address this, in our GDA framework, the sensors' reliability is estimated along with the decision aggregation process and provided as part of the final outputs to the user.

   Please note that high confidence does not necessarily imply high reliability. For example, a sensor should be labeled unreliable if it is always confident about its decisions that are actually wrong. The ability of accounting for sensor reliability thus differentiates our GDA framework from prior information integration schemes (e.g., data fusion [8]–[13]) that can also deal with continuous confidence probabilities.
2) Different from traditional decision aggregation schemes that assume all the events are observed by all the sensors, the proposed GDA framework is able to handle the scenarios where different sensor nodes observe different sets of events whose numbers of possible classes may also be different. Doing so enables us to seize more opportunities to estimate sensor reliability, thus leading to better final decision accuracy.
3) In order to be applied to newly emerged sensing scenarios where people are playing increasingly more critical roles, which implies more opportunities for ground truth label collection, we design our GDA framework from ground up to be able to cope with any availability level of label information, and do so in a dynamic and intelligent manner.

In summary, our GDA framework addresses challenges in carrying out classification tasks in distributed sensing systems by more naturally and organically modeling the sensing and decision making processes in dynamic and intelligent manners, it thus can be applied to a full spectrum of distributed sensing scenarios.

The rest of the paper is organized as follows. We first summarize the related work in Section II. Section III provides an overview of the system model and architecture. In Section IV, we formulate the generalized decision aggregation problem as an optimization program. An efficient algorithm is presented in Section V to solve this problem. The proposed algorithm is evaluated in Section VI. Finally, Section VII concludes the paper.

II. Related Work

Classification techniques are widely used in a full range of sensing scenarios, such as habitat monitoring [14]–[17], target surveillance and tracking [18], [19], environmental monitoring [20], activity recognition [21]–[23], road sensing and monitoring [24], and many others. Our generalized framework proposed in this paper can essentially be applied to all these sensing scenarios, addressing the decision aggregation problem by corroborating the scattered classification results and making the consolidated near-optimal final decision for the target events, and doing so in a resource efficient manner.

There are prior attempts on similar problems. For example, Su et al. [3], [4] study the decision aggregation problem for remotely deployed sensing systems where very limited label information can be accessed. Recently, the problem of truth discovery [5]–[7], [25], [26] is investigated in the data mining and social sensing communities. Their goal is to identify the truth from claims made by difference information sources (e.g., websites, social network users). These approaches suffer from a major limitation, that is, they only take as input discrete decision information. In contrast, our proposed decision aggregation framework is able to take advantage of the confidence information of each sensor about its decision, and thus achieves higher decision

accuracy. Recently, Li et al. [27] propose a truth discovery scheme that can deal with continuous values. Different from this work, we customize the proposed GDA framework to account for the sensor confidence probability vectors that are subject to the decision constraint.

Moreover, the proposed problem and solution in this paper are different from the traditional data aggregation or data fusion schemes in wireless sensor networks. First, data aggregation techniques [28], [29] do not consider sensor reliability, and usually only involve applying simple operations (e.g., mean, min, and max) directly on the raw sensory data. Second, data fusion (or classifier fusion) schemes [8]–[13] are designed to gather and combine information from multiple sensors in order to improve the accuracy of target detection and recognition. However, the aforementioned work does not take into consideration the reliability of individual sensor nodes. In contrast, the proposed GDA framework jointly optimizes aggregated decisions and sensor reliability, and can be applied in more general sensing scenarios where the sensor nodes, the observed event sets, and the possible candidate classes can all be different, which can thus be combined in arbitrarily complex manners.

III. System Overview

We now give an overview of our system model and architecture.

A. System Model

We consider a sensing system consisting of n sensor nodes that are denoted by $\mathcal{S} = \{s_i |\ i = 1, 2, \ldots, n\}$. The sensor nodes collect information about the events taking place within their sensing ranges, and classify these events into predefined classes. Formally, we let $\mathcal{E} = \{e_i | i = 1, 2, \ldots, t\}$ denote the sequence of events (sorted in chronological order) observed by the sensor nodes. Generally, each sensor observes a subset of events, and each event is observed by a subset of sensors. The relationship between sensor nodes and events can be represented as a bipartite graph, called *belief graph*, where vertices are partitioned into sensors and events, and edges represent the observation relationships of sensor-event pairs, as illustrated in Fig. 1.

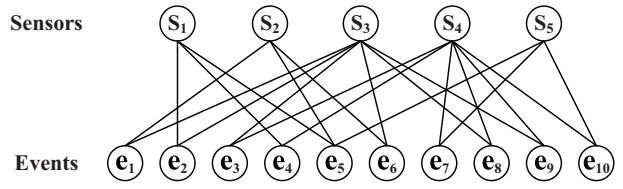


Figure 1. An example of belief graph

In this case, suppose the mission of the sensing system is to classify different vehicle types, specifically, to find out whether an observed vehicle is a tank, a jeep, or a truck. As shown, 10 events are observed by 5 sensor nodes. Each event corresponds to a vehicle. Each sensor can be either a sensing device deployed on the roadside, or a pedestrian in the vicinity.

B. System Architecture

In this section, we provide an overview of the system architecture. The system contains three modules: a data classification module, a decision aggregation module, and a feedback module. They are deployed on two different platforms: sensor nodes, and the base station. Figure 2 illustrates the system architecture. We next discuss each of these three modules in more detail.

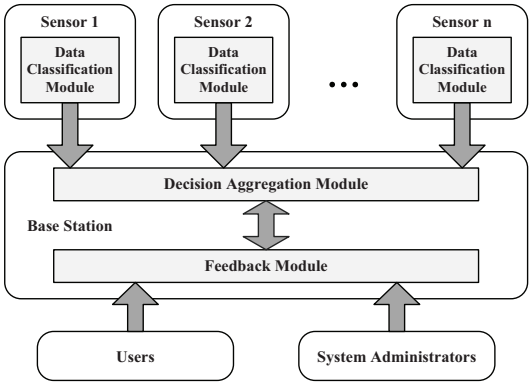


Figure 2. System Architecture

1) Data Classification Module: The data classification module runs on individual sensor nodes. It locally classify the events observed by each sensor node, and upload the classification result (i.e., decision vector) to the base station.

2) Decision Aggregation Module: As shown in Fig. 2, the decision aggregation module resides on the base station. It combines the decision vectors from multiple sensor nodes, and predict the class label of each unlabeled event.

3) Feedback Module: The feedback module is included to visually present data collection and/or decision results via a web service interface. It can also be used by system administrators and users to correct decision errors or provide ground truth information, which is then sent back to individual sensors for dynamic and adaptive performance improvement.

IV. Problem Formulation

With the previously defined notations and terminologies, we now formulate the generalized decision aggregation problem as an optimization program on the belief graph. In this section, we first introduce the variables and constants involved in the optimization program, then give detailed descriptions on the objective function as well as the constraints.

A. Variables and Constants

Constants: The constants of the optimization program are the inputs to the decision aggregation module, including the following:

- Belief Graph: We summarize the previously defined belief graph into an affinity matrix called observation matrix $\mathbf{A} = (a_{ij})_{t\times n}$, where a_{ij} indicates whether event e_i is observed by sensor s_j.
- Individual Decision: The decision of node s_j for e_i is a probability vector denoted by $\mathbf{d}_i^j = (d_{i1}^j, \ldots, d_{im_i}^j)$, where m_i is the number of possible classes of event e_i. In this vector, each element probability, say d_{ik}^j, represents the confidence level in which s_j "believes" the observed event belongs to the k-th class. For example, suppose sensor s_1 in Fig. 1 outputs a decision vector for event e_2 (i.e., $\mathbf{d}_2^1 = (0.8, 0.1, 0.1)$). This implies that s_1 believes that with 80% probability e_2 is a tank, and is a jeep or truck with 10% probability each.

Variables: The variables of the optimization program serve as the outputs of the decision aggregation module, including the following:

- Aggregated Decision: The aggregated decision for an event e_i is also a probability vector, denoted by $\mathbf{x}_i = (x_{i1}, \ldots, x_{im_i})$. It represents the consensus of the sensor nodes on the probability that e_i belongs to each candidate class.
- Sensor Reliability: As discussed in Section I, the reliability levels of individual sensor nodes should be taken into account when aggregating the decisions of multiple nodes. To capture sensor reliability, we associate each node, say s_j, with a non-negative weight w_j, where higher weights indicate higher reliability.

B. Optimization Program

Given the constants and variables defined above, we formulate the following optimization program:

$$\mathbf{P}: \min \quad \sum_{i=1}^{t}\sum_{j=1}^{n} a_{ij} w_j ||\mathbf{x}_i - \mathbf{d}_i^j||^2 \tag{1}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} \exp(-w_j) = 1 \tag{2}$$

$$\mathbf{x}_i \geq \mathbf{0}, \quad |\mathbf{x}_i| = 1 \quad \text{for} \;\; i = 1, 2, \ldots, t. \tag{3}$$

Objective Function: The objective function (1) aims at minimizing the disagreement over the belief graph, namely, the weighted summation of the distances between the decisions of individual sensor nodes and the aggregated decision. In this case, we use squared L2 norm as the distance function: $||\mathbf{x}_i - \mathbf{d}_i^j||^2 = (x_{i1} - d_{i1}^j)^2 + (x_{i2} - d_{i2}^j)^2 + \ldots + (x_{im_i} - d_{im_i}^j)^2$.

Intuitively, the optimal aggregated decision should be close to the majority of individual decisions[1]. Furthermore, the sensors with higher reliability score (i.e., w_j) should have more impact on the weighted summation. In other words, more reliable sensors would incur higher penalties if they deviate far away from the aggregated decision, as compared to less reliable ones. This way, the objective function tends to be minimized when the aggregated decision agrees with that of reliable sensors.

Constraints: Next, we elaborate on the constraints that our objective function is subject to.

- Reliability Constraint (2) is a regularization function. It is used to prevent the sensor weight w_j from going to infinity, otherwise the optimization problem would become unbounded. In fact, the most straightforward choice of regularization function could be $\sum_{j=1}^{n} w_j = 1$, which is unsuitable for our purpose as an optimal solution is achieved when the aggregated decision is set to that of any single sensor, whose weight is set to 1 and the rest sensors 0. Therefore we propose to formulate the regularization function using the sum over exponential value of weights. Exponential function is used to regularize weights so that they are rescaled by logarithm (the range of weights becomes smaller). One advantage of this regularization formulation is that a closed-form optimal solution can be derived.
- Decision Constraint (3) is used to guarantee that the elements of the decision probability vector $\mathbf{x}_i$ be non-negative, and sum to 1 (i.e., $\mathbf{x}_i$'s L1 norm $|\mathbf{x}_i| = \sum_{k=1}^{m_i} x_{ik} = 1$).

Unfortunately, $\mathbf{P}$ is not a convex program. This makes it difficult to find the global optimal solution. Next, we present an efficient approximate solution for the optimization program $\mathbf{P}$.

V. Generalized Decision Aggregation

We propose to solve $\mathbf{P}$ using the block coordinate descent method [30]. The basic idea is as follows: In each iteration, we update the values of sensor reliability and aggregated decision alternatively and separately. In particular, in the first step, we fix the weight (w_j) of each sensor node, and solve $\mathbf{P}$ with respect to the aggregated decision ($\mathbf{x}_i$) only. In the second step, $\mathbf{x}_i$ is fixed and $\mathbf{P}$ is solved with respect to w_j. The two-step process is repeated until convergence, which is guaranteed by the property of the block coordinate descent method. That is, if we can find the optimal aggregated decision (sensor reliability) when sensor reliability (aggregated decision) is fixed, convergence can be achieved [30]. Next, we give detailed explanation on these two steps, and show that each step itself is convex, and thus has a globally optimal solution.

A. Updating Aggregated Decision

When the reliability w_j of each sensor is fixed, we update the aggregated decision $\mathbf{x}_i$ for each event in order to minimize the weighted distances between $\mathbf{x}_i$ and the

[1]Here we assume the majority of the sensor nodes are functioning appropriately and thus can make reasonable decisions.

decisions $\mathbf{d}_i^j$ made by individual sensor nodes:

$$\mathbf{P_x}: \min \quad \sum_{i=1}^{t}\sum_{j=1}^{n} a_{ij}w_j||\mathbf{x}_i - \mathbf{d}_i^j||^2$$
$$\text{s.t.} \quad \mathbf{x}_i \geq \mathbf{0}, \quad |\mathbf{x}_i| = 1 \quad \text{for} \quad i = 1, 2, \ldots, t.$$

Different from $\mathbf{P}$, $\mathbf{P_x}$ has only one set of variables (i.e., $\mathbf{x}_i$'s), and thus is a convex program. This ensures that we can find globally optimal aggregated decisions. The detailed steps are as follows. First, we denote the objective function by

$$f(\mathbf{x}) = \sum_{i=1}^{t}\sum_{j=1}^{n} a_{ij}w_j \sum_{k=1}^{m_i}(x_{ik} - d_{ik}^j)^2,$$

then the optimal solution can be obtained through setting the partial derivative with respect to $\mathbf{x}$ to zero,

$$\frac{\partial f(\mathbf{x})}{\partial x_{ik}} = \sum_{j=1}^{n} 2a_{ij}w_j(x_{ik} - d_{ik}^j) = 0,$$

for $i = 1, 2, \ldots, t$ and $k = 1, 2, \ldots, m_i$.

Solving this equation, we are able to get the optimal value of x_{ik}:

$$x_{ik} = \frac{\sum_{j=1}^{n} a_{ij}w_j d_{ik}^j}{\sum_{j=1}^{n} a_{ij}w_j}.$$

Therefore, the optimal aggregated decision vector is actually the weighted average of individual decision vectors:

$$\mathbf{x}_i = \frac{\sum_{j=1}^{n} a_{ij}w_j \mathbf{d}_i^j}{\sum_{j=1}^{n} a_{ij}w_j}. \tag{4}$$

One should note that when we solve for $\mathbf{x}_i$, we do not take into account the decision constraint (Eqn. (3)). This is because the aggregated decisions obtained from Eqn. (4) can automatically satisfy the constraint. In particular, each individual decision $\mathbf{d}_i^j$ is a probability vector, obviously we have $\mathbf{d}_i^j \geq 0$ and $|\mathbf{d}_i^j| = 1$. Thus, it can be derived that

$$|\mathbf{x}_i| = \sum_{k=1}^{m_i} x_{ik} = \frac{\sum_{j=1}^{n} a_{ij}w_j|\mathbf{d}_i^j|}{\sum_{j=1}^{n} a_{ij}w_j} = 1$$

Moreover, since both a_{ij} and w_j are nonnegative, it is clear that $x_{ik} \geq 0$. Therefore, $\mathbf{x}_i$'s automatically satisfy the decision constraint.

B. Updating Sensor Reliability

Next, we fix the values of the aggregated decision $\mathbf{x}_i$, and update the reliability of each sensor w_j through solving the following optimization program:

$$\mathbf{P_w}: \min \quad \sum_{i=1}^{t}\sum_{j=1}^{n} a_{ij}w_j||\mathbf{x}_i - \mathbf{d}_i^j||^2$$
$$\text{s.t.} \quad \sum_{j=1}^{n} \exp(-w_j) = 1.$$

Similar to the previous step, $\mathbf{P_w}$ has only one set of variables, the w_j's. And the decision constraint (Eqn. (3)) in $\mathbf{P}$ is just constant here. $\mathbf{P_w}$ is clearly convex since the objective function is linear with respect to w_j, while the constraint is a convex function.

We use the method of Lagrange multipliers to solve $\mathbf{P_w}$. We first take a look at the Lagrangian of $\mathbf{P_w}$:

$$L(\mathbf{w}, \lambda) = \sum_{i=1}^{t}\sum_{j=1}^{n} a_{ij}w_j||\mathbf{x}_i - \mathbf{d}_i^j||^2 + \lambda(\sum_{j=1}^{n} \exp(-w_j) - 1).$$

In $L(\mathbf{w}, \lambda)$, λ is a Lagrange multiplier, corresponding to the reliability constraint. It can be interpreted as the "shadow price" charged for the violation of the constraint.

Let the partial derivative of Lagrangian with respect to w_j be 0:

$$\frac{\partial L(\mathbf{w}, \lambda)}{\partial w_j} = \sum_{i=1}^{t} a_{ij}||\mathbf{x}_i - \mathbf{d}_i^j||^2 - \lambda \exp(-w_j) = 0,$$

we can get

$$\frac{\sum_{i=1}^{t} a_{ij}||\mathbf{x}_i - \mathbf{d}_i^j||^2}{\lambda} = \exp(-w_j). \tag{5}$$

Summing both sides over j's, we have,

$$\frac{\sum_{j=1}^{n}\sum_{i=1}^{t} a_{ij}||\mathbf{x}_i - \mathbf{d}_i^j||^2}{\lambda} = \sum_{j=1}^{n} \exp(-w_j) = 1,$$

from which we can derive that

$$\lambda = \sum_{j=1}^{n}\sum_{i=1}^{t} a_{ij}||\mathbf{x}_i - \mathbf{d}_i^j||^2. \tag{6}$$

Plugging Eqn. (6) into Eqn. (5), we obtain a closed-form solution of reliability:

$$w_j = \log\Big(\frac{\sum_{j=1}^{n}\sum_{i=1}^{t} a_{ij}||\mathbf{x}_i - \mathbf{d}_i^j||^2}{\sum_{i=1}^{t} a_{ij}||\mathbf{x}_i - \mathbf{d}_i^j||^2}\Big). \tag{7}$$

As can be seen, the reliability of a sensor is the log ratio between the summed decision deviation (i.e., the difference between a sensor's decision and the aggregated decision) of all the sensors and the decision deviation of this sensor. Sometimes, the summed decision deviation may dominate individual decision deviations, and thus diminish the variance among the sensor radiabilities. In this case, we can replace the the summed decision deviation by the maximum decision deviation among all the sensors.

C. Algorithm

The detailed steps of the generalized decision aggregation (GDA) algorithm are shown in Algorithm 1. The algorithm takes as input the observation matrix $\mathbf{A}$ as well as the individual decision of each sensor s_j for each event e_i

(i.e., $\mathbf{d}_i^j$). It starts by initializing the aggregated decisions randomly[2] (line 1). The iterative process then begins in line 3. First, we collect the aggregated decision of each event observed by a sensor s_i, and update its reliability via Eqn. (7) (line 5). Then, the sensors' reliability information are used to consolidate the aggregated decision of each event e_i through Eqn. (4) (line 8). Finally, each event is assigned to the class corresponding to the highest probability in the aggregated decision (line 13).

Algorithm 1 Generalized Decision Aggregation

Input: Observation matrix $\mathbf{A}$, individual decisions $\mathbf{d}_i^j$, and error threshold ϵ;
Output: The class label for each event L_i;

1: Initialize $\mathbf{x}_i^{(0)}$, $\mathbf{x}_i^{(1)}$ randomly.
2: $\tau \leftarrow 1$
3: **while** $\sqrt{\sum_{i=1}^{t} ||\mathbf{x}_i^{(\tau)} - \mathbf{x}_i^{(\tau-1)}||^2} > \epsilon$ **do**
4: **for** $j \leftarrow 1$ to n **do**
5: $w_j^{(\tau+1)} \leftarrow \log\left(\frac{\sum_{j=1}^{n}\sum_{i=1}^{t} a_{ij}||\mathbf{x}_i^{(\tau)} - \mathbf{d}_i^j||^2}{\sum_{i=1}^{t} a_{ij}||\mathbf{x}_i^{(\tau)} - \mathbf{d}_i^j||^2}\right)$
6: **for** $i \leftarrow 1$ to t **do**
7: $\mathbf{x}_i^{(\tau+1)} \leftarrow \frac{\sum_{j=1}^{n} a_{ij} w_j^{(\tau+1)} \mathbf{d}_i^j}{\sum_{j=1}^{n} a_{ij} w_j^{(\tau+1)}}$
8: $\tau \leftarrow \tau + 1$
9: **for** $i \leftarrow 1$ to t **do**
10: **return** $L_i \leftarrow \arg\max_k x_{ik}^{(\tau)}$

D. Performance Analysis

In each iteration, the GDA algorithm takes $O(mnt)$ time, where n and t represent the number of sensors and events, while $m = \max_{e_i \in \mathcal{E}} m_i$ is the maximum number of classes among all the events. Also, the convergence rate of coordinate descent method is usually linear [30] (we actually fix the number of iterations in the experiments). In practice, the number of candidate classes of the observed events and the number of sensor nodes that observe the same events are usually small. Thus, the computational complexity of the algorithm can be considered linear with respect to the number of events. Consequently, the proposed algorithm is not more expensive than the classification algorithms, and thus can be applied to any platform running classification tasks. Furthermore, since wireless/wired communication is the dominating factor of the energy consumption in distributed sensing systems, our algorithm actually saves much more energy than it consumes since it significant reduces the amount of information delivered by each sensor by transforming its raw data into decisions.

[2]Since $\mathbf{P}$ is not a convex program, the block coordinate descent based algorithm would probably converge to local optimum. A common way to address this problem is to run the algorithm multiple times with a different set of randomly chosen initial aggregated decisions and select the best solution.

E. Example

We now walkthrough a simple example to illustrate the iterative process of the GDA algorithm. Table I provides the information of the first 5 events shown in Fig. 1. We omit other events' information due to space limitation. In this table, we list the sensor nodes that observe each of the 5 events, and the corresponding decision probability vectors generated by the sensors. In addition, the ground truth label of each event is given in the last column.

Table I
EVENTS

Event	Sensor Node	Decision Vector	Ground Truth
e_1	s_2	(0.1, 0.8, 0.1)	1
	s_3	(0.6, 0.2, 0.2)	
e_2	s_1	(0.3, 0.5, 0.2)	3
	s_3	(0.1, 0.3, 0.6)	
e_3	s_3	(0.8, 0.1, 0.1)	1
	s_4	(0.8, 0.1, 0.1)	
e_4	s_1	(0.4, 0.3, 0.3)	2
	s_4	(0.2, 0.6, 0.2)	
e_5	s_1	(0.5, 0.5, 0)	2
	s_2	(0.7, 0.3, 0)	
	s_5	(0.3, 0.7, 0)	

We apply the GDA algorithm to the sensing system in Fig. 1. Initially, the aggregated decision of each event is set as the average of individual decisions made by the sensors that observe this event. The predicted label corresponds to the class with the highest probability. In rare cases where ties occur, we break them randomly.

Table II
FINAL RESULTS

Event	Aggregated Decision	Label	Sensor	Reliability
e_1	(0.5701,0.2358,0.1941)	1	s_1	1.6093
e_2	(0.1517,0.3517,0.4966)	3	s_2	0.2731
e_3	(0.8,0.1,0.1)	1	s_3	4.5801
e_4	(0.2505,0.5243,0.2252)	2	s_4	4.7438
e_5	(0.3712,0.6288,0)	2	s_5	3.9157

Then, the algorithm starts to iterate, and update the values of sensor reliability and aggregated decisions repeatedly. After the algorithm converges, as shown in Table II, the predicted label for each event exactly matches the ground truth. From the results, we have several observations. For example, sensor s_2 and s_3 have conflicting decisions about event e_1, and s_2 is more confident with its decision. Thus, the simple averaged decision gives a predicted label of 2, which contradicts against the ground truth. In this case, the GDA algorithm outputs the correct label, because it takes into account the reliability of individual sensors. As can be seen in Table II, the reliability score of s_3 is much higher than that of s_2, so the aggregated result should favor s_3's decision. A decision is considered to be correct (or incorrect)

if the class with the highest probability in the decision vector matches (or differs from) the ground truth label. As shown in Table II, the sensors that can make more correct decisions are assigned higher reliability scores.

VI. Performance Evaluation

In this section, we evaluate the proposed generalized decision aggregation (GDA) framework. Experiment results on both synthetic data and a set of realistic audio recordings are presented and discussed. We compare our GDA framework against the naïve majority voting scheme as well as state-of-the-art truth discovery and data fusion approaches. The experiment results show that GDA excels under various settings.

A. Synthetic Data

In this experiment, we simulate a sensing system where a set of events are observed and monitored by multiple sensor nodes. In particular, we randomly generate events belonging to different classes. For simplicity, we assume that each class contains the same number of events. Then, we generate sensor nodes with uniformly distributed reliability. The sensors with higher reliability are more likely to generate decision vectors whose highest probabilities correspond to the ground truth event label.

For comparison purposes, we include four baseline methods in the experiment.

The first two baselines take the discrete decision information as input:

- **Majority Voting**: counts the votes for each class, and picks the one with the highest vote count.
- **EM TruthFinder**: is a state-of-the-art truth discovery approach [5], [6], which uses Expectation Maximization (EM) algorithm to jointly optimize the correctness of the claims made by a group of sources and the reliability of these sources.

For each individual sensor decision, we feed these two baselines with the class label that has the highest confidence in the decision vector.

The other two baselines are representative data fusion schemes [8]–[11] that also take advantage of confidence levels of individual sensors when integrating their decisions:

- **Product-Rule Fusion**: under our problem setting, is equivalent to multiplying the decision probabilities over the sensors and labeling the event with the class corresponding to the largest probability product:

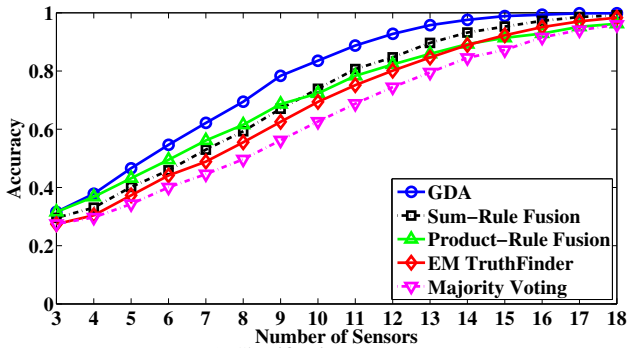
$$L_i \leftarrow \arg\max_k \prod_{j=1}^{n} d_{ik}^j. \tag{8}$$

- **Sum-Rule Fusion**: behaves exactly the same as the product rule fusion, except that summation is used instead of multiplication:

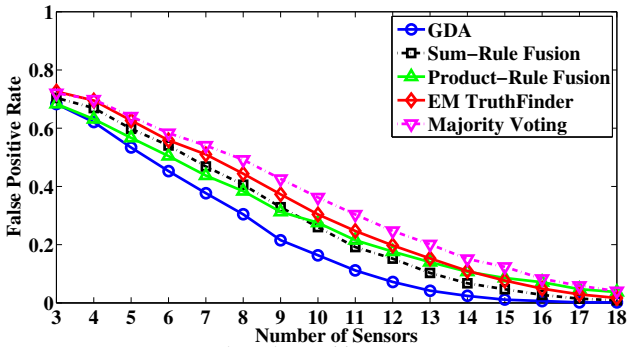
$$L_i \leftarrow \arg\max_k \sum_{j=1}^{n} d_{ik}^j. \tag{9}$$

  As can be seen, if normalized by dividing each sum by n, the sum rule fusion is equivalent to simply averaging the decision probability vectors on each event and labeling this event corresponding to the class with the highest probability in the averaged decision vector.

Both of the above fusion schemes, despite of being able to deal with continuous confidence probabilities, fail to take into account the varying sensor reliability.



(a) Classification Accuracy



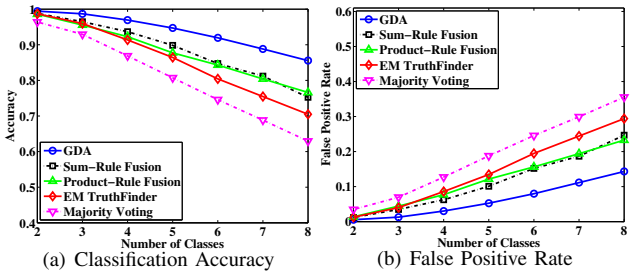
(b) False Positive Rate

Figure 3. Classification performance under different sensor numbers on synthetic data.

1) Classification Performance under Varying Number of Sensors: We first demonstrate the classification performance under varying number of sensor nodes that observe the same events. We generate 6 classes with 100 events each, where the number of observing sensors varies from 3 to 18. The experiment is repeated 10 times. We report the average results.

Figure 3(a) and 3(b) show, for all approaches, their classification accuracies (the percentage of correctly classified events, equivalent to true positive rate in this case), and false positive rates (the percentage of misclassifications of all the events that are classified to be of a particular class, then averaged among all classes), respectively. As clearly seen, our GDA framework outperforms the other

approaches under any number of observing sensors in terms of both classification accuracy and false positive rate, as the classification benefits from accounting for both sensor reliability and decision confidences. On the other end of the spectrum, the majority voting yields the worst performance as it disregards useful information (sensor reliability and decision confidences) that otherwise would be useful for reaching more accurate final decisions. The EM TruthFinder and data fusion approaches take only one factor (sensor reliability for EM TruthFinder, or decision confidences for data fusion) into consideration when aggregating individual decisions, therefore they, though outperforming majority voting, still fall short compared to our GDA approach, which utilizes *all* useful information. One other interesting observation is that all methods show similar performance when the number of sensors is either very small or quite large (e.g., 3 and 18, respectively in our experiments). This makes sense because, on one hand, when the number of sensors that observe the same events is small, it is hard to improve upon their individual poor decisions; On the other hand, as the number of sensors increases, each event is being observed by more and more diversified sensor nodes, which are more and more likely to cancel out each other's biases and errors, thus reaching better classification results. When there are a sufficiently large number of observing sensors, even the most naïve approach (e.g., majority voting) can achieve near perfect classification accuracy.



(a) Classification Accuracy (b) False Positive Rate

Figure 4. Classification performance under different number of classes on synthetic data.

2) Classification Performance under Varying Number of Classes: Next, we look at how GDA's classification performance compares to the other approaches with varying number of classes. The results are shown in Fig. 4. In this experiment, we assume that each event is observed by 10 different sensors, and each class contains 100 events. The number of classes ranges from 2 to 8.

Figure 4(a) and 4(b) show the classification accuracies and false positive rates of all approaches. As seen, our GDA approach consistently outperforms the other methods regardless of the number of classes, where the relative effectiveness of all studied approaches remains the same as that of the previous experiment. This is not surprising, as, still, the scheme that can take advantage of more information performs better. Also seen from the figures, it is clear that all approaches' classification performance degrades as the number of classes increases. This is generally expected for any classification task as the more candidate classes there are, the more confusion the classification algorithms need to comb through. We do, however, notice that as the number of classes increases, our GDA's performance degradation is slightly slower than the other approaches in general.

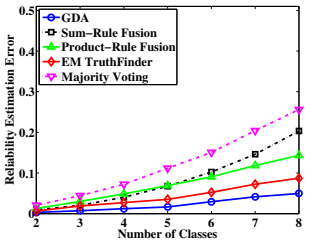


Figure 5. Estimation errors of sensor reliability under different number of classes.

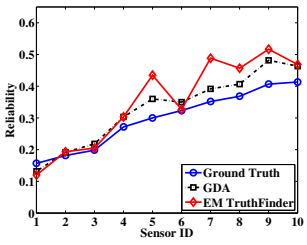


Figure 6. Reliability measures of 10 sensors observing the same set of events.

3) Sensor Reliability: Using the same setting as the previous experiment, we now examine how the different schemes perform in terms of estimating sensor reliability by comparing the reliability estimation errors of our proposed GDA framework to that of the other four approaches, under varying number of classes. In particular, the reliability estimation error is computed as follows. For each individual sensor node, its ground truth reliability is defined as its standalone classification accuracy derived from comparing its individual decisions to the ground truth event labels, and its estimated reliability under a particular scheme is the classification accuracy derived from comparing its individual decisions to the aggregated decision reached by that scheme. A sensor's reliability estimation error is thus computed as the normalized distance between its estimated and ground truth reliability.

The results are shown in Figure 5. Similar to previous experiments' results, our proposed GDA still consistently outperforms the others. In particular, we see that the approaches that take sensor reliability into account when performing decision aggregations (i.e., GDA and EM TruthFinder) achieve better performance than those who do not (i.e., Data fusion and Majority voting). Also, as the number of classes becomes larger, the estimation performance of all approaches gets poorer. Similar to the previous experiment, a higher number of classes would lengthen the distance between the aggregated decisions and ground truth event labels, thus causing more inaccurate sensor reliability estimations. That said, we do, however, still observe that our GDA scheme shows higher robustness than the other four methods as the number of classes increases.

Figure 6 shows the reliability of 10 sensor nodes that observe the same set of events. For ease of illustration, we sort the sensor nodes in the increasing order of ground truth

reliability. As can be seen, the ground truth reliability of these 10 sensor nodes roughly follow a uniform distribution. In Fig. 6, we also show the reliability as estimated by our GDA as well as the truth discovery schemes. It is clearly seen that the estimations from our GDA framework follow more closely to the ground truth.

4) Convergence: Next up, we demonstrate the convergence of GDA. Specifically in the experiment we have 600 events equally distributed under 6 classes, where each event is observed by 10 different sensor nodes.

Figure 7 shows the evolution of the objective value (Eqn. (1)) of $\mathbf{P}$, which denotes the weighted summation of the distances between individual decisions and the aggregated decision. According to the GDA algorithm, the objective value is initialized based on randomly selected aggregated decisions and the resultant sensor reliability. In the subsequent iterations of the algorithm, the objective value is progressively reduced by optimizing the aggregated decisions and the sensor reliability alternatively. As shown in Fig. 7, the objective value converges quickly within just a few iterations.

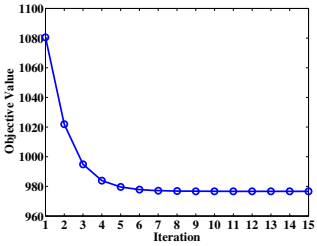


Figure 7. Convergence

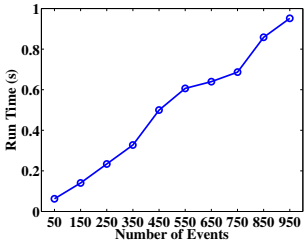


Figure 8. Run time

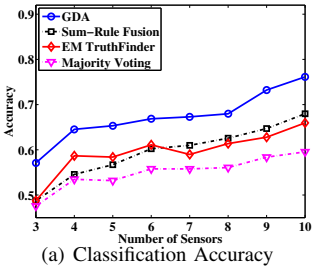
5) Complexity: Lastly, we look at GDA's computational complexity. We demonstrate that GDA's running time is linear with respect to the number of events under practical settings where, in particular, events are equally distributed under 6 classes with each event being observed by 10 different sensor nodes.

Figure 8 shows the running times of GDA under different input sizes (i.e., the number of events in each class). As seen, GDA displays linear complexity with respect to the number of events. To further demonstrate this, we compute Pearson's correlation coefficient, a commonly used metric for testing linear relationship between variables. The coefficient ranges between -1 and 1, and the closer it is to 1 (or -1), the stronger the variables are positively (or negatively) linearly correlated. In our experiment, the Pearson's correlation coefficient for running time and the number of events is 0.985, indicating strong positive linear correlation.

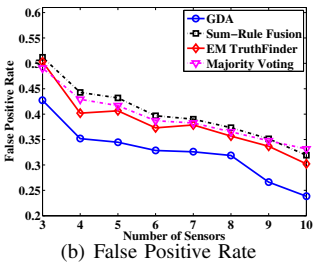
B. Audio Data

We next shift our attention from synthetic data to realistic audio data, using which we examine the classification performance of our GDA framework as well as the aforementioned baseline approaches except the product rule fusion scheme. The product rule fusion scheme suffers when sources give confident but conflicting decision vectors. For example, if two sensors come up with the decision vectors like $(0, 1, 0)$ and $(0.8, 0, 0.2)$ for a particular event, then the resulted vector product would be $(0, 0, 0)$ no matter what decision vectors other sensors may provide, resulting in an undecidable L_i. We found that such cases occur frequently with the real audio data, and thus exclude this baseline from this experiment.

The audio clips we use in this experiment include the sounds of a tank moving, a helicopter flying, and a machine gun firing, corresponding to 3 different classes. We cut the audio clips into pieces with equal time duration, and make a copy for each sensor node. We then add random noise to the sounds received by sensor nodes with various SNRs (Signal-to-Noise Ratios). Next, we extract the MFCC (Mel-Frequency Cepstral Coefficients) features from each audio clip, and feed them as the input to the classification algorithms. In this experiment, we choose random forest as individual sensor node's local classifier. Random forest is a decision tree based classification algorithm that trains multiple decision trees simultaneously and has them vote for the final classification decision. Random forest can output both decision probability vectors and discrete labels (derived from decision probability vectors) that are fed to different approaches under evaluation.



(a) Classification Accuracy

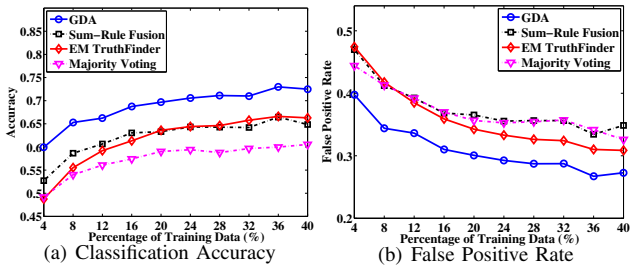


(b) False Positive Rate

Figure 9. Classification performance under different number of sensors on realistic audio data

The classification result with varying number of sensor nodes is shown in Figure 9, which, as seen, is quite similar to that of the experiment on synthetic data shown in Figure 3. The curves, however, are not as smooth, due to the randomness in the audio sounds themselves. Nevertheless, we can still observe the same general performance trends as displayed in the previous experiment.

Figure 10 shows the classification performance of studied approaches under varying training data availability levels. We see that the general relative classification effectiveness of all approaches remains the same as all previous experiments, with our GDA framework consistently yielding the best performance. Also, the figure shows that, for all approaches, higher training data availability lead to better classification performance, as expected.



(a) Classification Accuracy (b) False Positive Rate

Figure 10. Classification performance under varying training data availability levels on realistic audio data

VII. Conclusions

In this paper we take a closer look at the decision aggregation problem in distributed sensing systems. Though some efforts have been made towards this problem, the resulting approaches suffer from the limitation of only examining discrete decisions from individual sensor nodes as a way to avoid high energy cost potentially caused by excessive network transmission if raw data from sensor nodes were to be transmitted. Our proposed generalized decision aggregation (GDA) framework overcame this limitation by thoroughly accounting for and intelligently taking advantage of the decision confidence and reliability of each sensor, thus consistently achieving higher final decision accuracy over the state of the art approaches, as we extensively demonstrated through various experiments using both synthetic and realistic data. We believe our GDA framework's superior generalizability and flexibility make it suitable for a broad spectrum of sensing scenarios.

References

[1] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann, 2011.

[2] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[3] L. Su, Y. Yang, B. Ding, J. Gao, T. Abdelzaher, and J. Han, "Hierarchical aggregate classification with limited supervision for data reduction in wireless sensor networks," in *SenSys*, 2011.

[4] L. Su, S. Hu, S. Li, F. Liang, J. Gao, T. F. Abdelzaher, and J. Han, "Quality of information based data selection and transmission in wireless sensor networks," in *RTSS*, 2012.

[5] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, "On truth discovery in social sensing: a maximum likelihood estimation approach," in *IPSN*, 2012.

[6] D. Wang, T. Abdelzaher, L. Kaplan, R. Ganti, S. Hu, and H. Liu, "Exploitation of physical constraints for reliable social sensing," in *RTSS*, 2013.

[7] S. Wang, D. Wang, L. Su, L. Kaplan, and T. Abdelzaher, "Towards cyber-physical systems in social spaces: The data reliability challenge," in *RTSS*, 2014.

[8] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 3, pp. 226–239, 1998.

[9] R. R. Brooks, P. Ramanathan, and A. M. Sayeed, "Distributed target classification and tracking in sensor networks," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1163–1171, 2003.

[10] L. M. Kaplan, S. Chakraborty, and C. Bisdikian, "Fusion of classifiers: A subjective logic perspective," in *Aerospace Conference*, 2012.

[11] H. Wen, Z. Xiao, A. C. Symington, A. Markham, and N. Trigoni, "Comparison of accuracy estimation approaches for sensor networks," in *DCOSS*, 2013.

[12] G. Xing, R. Tan, B. Liu, J. Wang, X. Jia, and C.-W. Yi, "Data fusion improves the coverage of wireless sensor networks," in *MobiCom*, 2009.

[13] R. Tan, G. Xing, Z. Yuan, X. Liu, and J. Yao, "System-level calibration for fusion-based wireless sensor networks," in *RTSS*, 2010.

[14] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *WSNA*, 2002.

[15] Y. Guo, P. Corke, G. Poulton, T. Wark, G. Bishop-Hurley, and D. Swain, "Animal behaviour understanding using wireless sensor networks," in *LCN*, 2006.

[16] J. Cai, D. Ee, B. Pham, P. Roe, and J. Zhang, "Sensor network for the monitoring of ecosystem: Bird species recognition," in *ISSNIP*, 2007.

[17] W. Hu, V. N. Tran, N. Bulusu, C. T. Chou, S. Jha, and A. Taylor, "The design and evaluation of a hybrid sensor network for cane-toad monitoring," in *IPSN*, 2005.

[18] L. Gu, D. Jia, P. Vicaire, T. Yan, L. Luo, A. Tirumala, Q. Cao, T. He, J. A. Stankovic, T. Abdelzaher, and B. H. Krogh, "Lightweight detection and classification for wireless sensor networks in realistic environments," in *SenSys*, 2005.

[19] R. R. Brooks, P. Ramanathan, and A. M. Sayeed, "Distributed target classification and tracking in sensor networks," in *Proceedings of the IEEE*, 2003, pp. 1163–1171.

[20] X. Cheng, J. Xu, J. Pei, and J. Liu, "Hierarchical distributed data classification in wireless sensor networks," in *MASS*, 2009.

[21] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," in *Pervasive*, 2004.

[22] K. Lorincz, B.-r. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh, "Mercury: a wearable sensor network platform for high-fidelity motion analysis," in *Sensys*, 2009.

[23] Z. Zeng, S. Yu, W. Shin, and J. C. Hou, "PAS: A Wireless-Enabled, Cell-Phone-Incorporated Personal Assistant System for Independent and Assisted Living," in *ICDCS*, 2008.

[24] S. Hu, L. Su, H. Liu, H. Wang, and T. Abdelzaher, "S-martroad: a crowd-sourced traffic regulator detection and identification system," in *IPSN*, 2013.

[25] X. Yin, J. Han, and P. S. Yu, "Truth discovery with multiple conflicting information providers on the web," in *KDD*, 2007.

[26] ——, "Truth discovery with multiple conflicting information providers on the web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 6, pp. 796 – 808, 2008.

[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 *SIGMOD*, 2014.

[28] S. M. Michael, M. J. Franklin, J. Hellerstein, and W. Hong, "Tag: a tiny aggregation service for ad-hoc sensor networks," in *OSDI*, 2002.

[29] B. Krishnamachari, D. Estrin, and S. B. Wicker, "The impact of data aggregation in wireless sensor networks," in *ICDCSW*, 2002.

[30] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1995.