Scalable Social Sensing of Interdependent Phenomena

Shiguang Wang[1], Lu Su[2], Shen Li[1], Shaohan Hu[1], Tanvir Amin[1], Hongwei Wang[1]
Shuochao Yao[1], Lance Kaplan[3], Tarek Abdelzaher[1]
[1]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801
[2]Department of Computer Science and Engineering, SUNY at Buffalo, Buffalo, NY 14260
[3]Networked Sensing and Fusion Branch, US Army Research Labs, Adelphi, MD 207843

ABSTRACT

The proliferation of mobile sensing and communication devices in the possession of the average individual generated much recent interest in social sensing applications. Significant advances were made on the problem of uncovering ground truth from observations made by participants of unknown reliability. The problem, also called *fact-finding* commonly arises in applications where unvetted individuals may opt in to report phenomena of interest. For example, reliability of individuals might be unknown when they can join a participatory sensing campaign simply by downloading a smartphone app. This paper extends past social sensing literature by offering a scalable approach for exploiting dependencies between observed variables to increase fact-finding accuracy. Prior work assumed that reported facts are independent, or incurred exponential complexity when dependencies were present. In contrast, this paper presents the first scalable approach for accommodating dependency graphs between observed states. The approach is tested using real-life data collected in the aftermath of hurricane Sandy on availability of gas, food, and medical supplies, as well as extensive simulations. Evaluation shows that combining expected correlation graphs (of outages) with reported observations of unknown reliability, results in a much more reliable reconstruction of ground truth from the noisy social sensing data. We also show that correlation graphs can help test hypotheses regarding underlying causes, when different hypotheses are associated with different correlation patterns. For example, an observed outage profile can be attributed to a supplier outage or to excessive local demand. The two differ in expected correlations in observed outages, enabling joint identification of both the actual outages and their underlying causes.

Categories and Subject Descriptors

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

IPSN '15, April 14 - 16, 2015, Seattle, WA, USA.
Copyright 2015 ACM 978-1-4503-3475-4/15/04 ...$15.00.
http://dx.doi.org/10.1145/2737095.2737114.

Keywords

Social Sensing, Data Reliability, Maximum Likelihood Estimators, Expectation Maximization

1. INTRODUCTION

Recent advent of sensing applications, where unvetted participants can perform measurements, generated interest in techniques for uncovering ground truth from observations made by sources of *unknown* reliability. This paper extends prior work by offering scalable algorithms for exploiting known dependency graphs between observed variables to improve the quality of ground truth estimation.

Consider, for example, post-disaster scenarios, where significant portions of a city's infrastructure are disrupted. Communication resources are scarce, rumors abound, and means to verify reported observations are not readily available. Survivors report to a central unit the locations of damage and outages, so that help may be sent. Some reports are accurate, but much misinformation exists as well. Not knowing the individual sources in advance, it may be hard to tell which reports are more reliable. Simply counting the number of reports that agree on the facts (called *voting* in prior literature) is not always a good measure of fact correctness, as different sources may have different reliability. Hence, a different weight should be associated with each report (or vote), but that weight is not known in advance.

Prior work of the authors addressed the above problem when the reported observations are independent [22] and considered the case where second-hand observations were reported by other than the original sources [21]. In work that comes closest to the present paper, an algorithm was presented for the case, where the reported variables are correlated [20]. Unfortunately, the computational and representational complexity of the correlation was exponential in the number of correlated variables. Hence, in practice, it was not feasible to consider more than a small number of correlated variables at a time.

In sharp contrast to the above results, in this paper, we consider the case where reported variables have non-trivial dependency graphs. For example, upon the occurrence of a natural or man-made disaster, flooding, traffic conditions, outages, or structural damage in different parts of a city may be correlated at large scale. Furthermore, the structure of the correlations might be partially known. Areas of the same low elevation may get flooded together. Nearby parts of the same main road may see correlated traffic conditions. Buildings on the same power line may suffer correlated power outages. Gas stations that have the same supplier might have correlated availability of gas. Correlations (e.g., among failures) can also shed light on the root cause. For example, in a situation where a supply line simultaneously feeds several consumers, a failure in the line will result in correlated failures at the downstream consumers. If the topology of the supply lines is known, so is the correlation structure among expected consumer failures. If consumers build products that need multiple suppliers, knowing the pattern of the corelated consumer failures can give strong evidence as to which one of the suppliers may have failed.

Clearly, if the aforementioned correlation structure is not known, we cannot use this approach. Scenarios where correlations structures are not known can be addressed using prior work that simply views the underlying variables as uncorrelated [22]. This paper offers performance and accuracy improvements in the special (but important) case, where hypotheses regarding possible correlations are indeed available. Exploiting such correlations reduces problem dimensionality, allowing us to infer the state of points of interest more accurately and in a more computationally efficient manner.

What complicates the problem (in social sensing scenarios) is that actual state of the underlying physical system is not accurately known. All we have are reports from sources of unknown reliability. This paper develops the first scalable algorithm that takes advantage of the structure of correlations between (large numbers of) observed variables to better reconstruct ground truth from reports of such unreliable sources. We show that our algorithm has better accuracy than prior schemes such as voting and maximum likelihood schemes based on independent observations [22]. It also significantly outperforms, in terms of scalability, previous work that is exponential in dependency structures [21].

The general idea behind the scalability of the new scheme lies in exploiting conditional independence, when one catalyst independently causes each of multiple consequences to occur. Identification of such conditional independence relations significantly simplifies reasoning about the joint correlations between observed values, thus simplifying the exploitation of such correlations in state estimation algorithms. Although previous work [20] considers correlated variables in social sensing applications, it does not exploit conditional independence. The computational complexity of the previous solution increases exponentially in the number of correlated variables, which makes it applicable only to applications with a small number of such variables. By modeling the structural correlations of variables as a Bayesian network and exploiting conditional independence, our algorithm is more computationally efficient. Its computational complexity depends on the size of the largest clique (i.e., complete sub-graph) in the Bayesian network, while the complexity of the previous solution [20] depends on the total number of nodes in the Bayesian network making the latter intractable for applications with a large number of correlated variables.

The contributions of this paper are thus summarized as follows:

- We extend the previous solution to a new application domain in which a large number of variables are structurally interdependent. Our algorithm is shown to be more accurate and computationally efficient, compared with previous solutions.

- The interdependent structure is formulated as a Bayesian network, which enables us to exploit well-established techniques of Bayesian network analysis. We also show that the Bayesian network generalizes the models we proposed in previous work.

- Our algorithm is evaluated by both extensive simulations and a real-world data set. The evaluation results show that our solution outperforms the state of the art.

The rest of the paper is organized as follows. We formulate our problem in Section 2. In Section 3, we argue that our solution is general and can be applied to solve previous social sensing challenges by showing that all the previous models are special cases of our Bayesian model. We propose our solution in Section 4 and evaluate our algorithm in Section 5. A literature review is presented in Section 6. The paper concludes in Section 7.

2. PROBLEM FORMULATION

Social sensing differs from sensing paradigms that use in-field physical sensors (e.g. Wireless Networked Sensing [15]) in that it exploits sensors in social spaces. Examples include sensor-rich mobile devices like smartphones, tablets, and other wearables, as well as using *humans as sensors*. The involvement of humans in the sensing process enables an application to directly sense variables with higher-level semantics than what traditional sensors may measure. However, unlike physical devices, which are usually reliable or have the same error distribution, the reliability of human sources is more *heterogeneous* and may be *unknown a priori*. This source reliability challenge in social-sensing systems was recently articulated by Wang et al. [22]. Solutions that estimate source reliability were improved in follow-up publications [20, 21, 23].

In recent work, the authors modeled human sources as *binary* sensors, reporting events of interest. The rationale behind the binary model is that humans are much better at categorizing classes of observations than at estimating precise values. For example, it is much easier to tell whether a room is warm or not than to tell its exact temperature. Binary variables can be easily extended to multivalued ones [23], which makes the binary model versatile. In this paper, we adopt the binary model and assume that a group of human sources, denoted by $\mathcal{S}$, participate in a sensing application to report values of binary variables, we call the *event variables*. For example, they may report the existence or absence of gas at a set of gas stations after a hurricane. These variables are collectively denoted by $\mathcal{C}$. The goal of this paper is to jointly estimate both the source reliability values and ground-truth measured variable values, given only the string of noisy reports. In contrast to prior work, we assume that the underlying variables are structurally correlated at scale. The question addressed in this paper is how to incorporate knowledge of these correlations into the analysis.

2.1 Modeling Interdependent Event Variables

In previous work, event variables were either assumed to be independent [22, 21] or were partitioned into groups of small size [20, 23] with no dependencies among groups. Solution complexity grew exponentially with the maximum group size. In practice, it is not uncommon that all or a large portion of event variables are interdependent. For example, in an application that monitors traffic conditions in a city, pertinent variables might denote weather conditions (e.g., snowy or rainy weather), local entertainment events that impact traffic (e.g., football games or concerts), road surface conditions (e.g., potholes on road surfaces), and traffic speed, among others. These variables are correlated. Bad weather results in slow traffic. So do the local entertainment events and bad road surfaces. Traffic congestion on one road segment might cause congestion on another road segment. The pervasive dependencies among variables make previous work (e.g., Wang et al. [20]) inapplicable due to intractability, thus calling for a better model to handle them. This paper is the *first* to study the reliable social-sensing problem with interdependent variables, at scale.

Our solutions are based on the insight that although independence is uncommon in real applications, *conditional independence* does often arise. As stated in [12], dependencies usually expose some structure in reality. The dependency structure encodes conditional independence, that can be leveraged to greatly simplify the estimation of values of variables. In the previous application example, given that the weather is snowy, the resulting traffic congestion on two road segments can be assumed to be conditionally independent. Both are caused by snowy weather but neither is affecting the other (assuming they are sufficiently far apart). However, without knowing the state of the weather, we are not able to assume that congestion on both segments is independent. Measuring congestion on those segments, it will tend to be correlated (in the presence of snow events).

In this paper, we model dependencies among variables by a Bayesian network [13]. The underlying structure of a Bayesian network is a directed acyclic graph (DAG) in which each node V corresponds to a variable, v, and each arc $U \rightarrow V$ denotes a conditional dependence such that the value of variable v is dependent on the value of u. The Bayesian network is a natural way to model causal relations between variables. Since Bayesian networks are well-established tools for statistical inference, we can leverage prior results to solve our reliable social-sensing problem.

Of course, in some cases, the underlying dependences can form a complete graph in which any pair of variables are *directly* interdependent. In this extreme situation, there would be no efficient inference algorithm with computational complexity inferior to $\Theta(2^N)$, where N denotes the total number of variables (i.e., $|\mathcal{C}|$). All inferences should be made by considering the joint distribution of all variables. However, as stated in [12], the complete graph structure does not often happen in real applications, and so we are not interested in this extreme case.

2.2 Categorized Source Reliability

Although previous work in social sensing assumes that sources have different reliability, for a specific source, its reliability is assumed to be fixed (e.g., [22, 20, 21]. This fixed-reliability assumption does not hold in many practical scenarios. For example, a diabetic person who is in need of insulin might be a better source to ask about pharmacies that remained open after a natural disaster, than a person who is not in need of medication. The same diabetic person might not be a good source to ask about gas stations that are open, if the person does not own a car. In the above scenario, if we assume that a single source has the same reliability in reporting all types of variables, the performance of estimating the ground truth of these variables might be degraded. To make the source reliability model more practical, and thus the estimation more accurate, we assume that source reliability differs depending on the variable reported. Measured variables are classified into different categories. Source reliability is computed separately for each category. We call it *categorized source reliability*.

With the categorized-source-reliability model, the reliability of each source is represented by a vector (where each element is corresponding to the reliability for some reported category of variabls, rather than a scalar as in previous work. Please note that the previous reliability model is a special case of our model as a single-element vector.

2.3 Problem Definition

Next, we formally define our reliable social-sensing problem with interdependent variable at scale. We denote the j-th measured variable by C_j, and C_j is assumed to be binary. More specifically, $C_j \in \{T, F\}$ where T represents True (e.g., "Yes, the room is warm"), and F represents False (e.g., "No, the room is not warm"). One can think of each variable as the output of a different application-specific True/False predicate about physical world state. Each variable C_j belongs to some category ℓ, denoted by ${}^{\ell}C_j$. We use $\mathcal{L}$ to denote the category set.

In social-sensing, a source reports the values of variables. We call those reports, *claims*. We use a matrix SC to represent the claims made by all sources $\mathcal{S}$ about all variables $\mathcal{C}$. We call it the *source-claim matrix*. In the source-claim matrix, an element $SC_{i,j} = v$ means that the source S_i claims that the value of variable C_j is v. It is also possible that a source does not claim any value for some variable, in which case the corresponding item in the source-claim matrix $SC_{i,j}$ is assigned value U (short for "Unknown") meaning that the source did not report anything about this variable. Therefore, in the source-claim matrix, SC, each item $SC_{i,j}$ has three possible values T, F and U.

We define the reliability of source S_i in reporting values of variables of category ℓ as the probability that variables belonging to that category indeed have the values that the source claims they do. In other words, it is the probability that ${}^{\ell}C_j = v$, given that $SC_{i,j} = v$. In the following, we shall use the short notation X^v to denote that the variable X is of value v (i.e., $X = v$). Let ${}^{\ell}t_i$ denote the reliability of source S_i in reporting values of variables of category ℓ. We formally define the source reliability as follows:

$$ {}^{\ell}t_i = \Pr\left({}^{\ell}C_j^v | SC_{i,j}^v\right). \tag{1} $$

Let ${}^{\ell}T_i^v$ denote the probability that source S_i reports the value of variable ${}^{\ell}C_j$ correctly. In other words, the probability that S_i reports value v for variable ${}^{\ell}C_j$ given that its value is really v. Furthermore, let ${}^{\ell}F_i^v$ denote the probability of an incorrect report by S_i. In other words, it is the probability that S_i reports that ${}^{\ell}C_j$ has value $\bar{v}$ given that its value is v. Here $\bar{x}$ is the complement of x ($\bar{T} = F$ and $\bar{F} = T$). ${}^{\ell}T_i^v$ and ${}^{\ell}F_i^v$ are formally defined below:

$$ {}^{\ell}T_i^v = \Pr\left(SC_{i,j}^v | {}^{\ell}C_j^v\right), \ {}^{\ell}F_i^v = \Pr\left(SC_{i,j}^{\bar{v}} | {}^{\ell}C_j^v\right). \tag{2} $$

Note that, ${}^{\ell}T_i^v + {}^{\ell}F_i^v \leq 1$, since it is possible that the source S_i does not report anything of a variable. Therefore, we have:

$$ 1 - {}^{\ell}T_i^v - {}^{\ell}F_i^v = \Pr\left(SC_{i,j}^U | {}^{\ell}C_j^v\right). \tag{3} $$

Table 1: The summary of notations

Set of sources	$\mathcal{S}$
Set of variables	$\mathcal{C}$
Binary variable, j	C_j
Variable X of category ℓ	${}^{\ell}X$
Binary value set	$\{T, F\}$
Source-claim matrix	SC
Source reliability	${}^{\ell}t_i = \Pr\left({}^{\ell}C_j^v \mid SC_{i,j}^v\right)$
Correctness probability	${}^{\ell}T_i^v = \Pr\left(SC_{i,j}^v \mid {}^{\ell}C_j^v\right)$
Error probability	${}^{\ell}F_i^v = \Pr\left(SC_{i,j}^{\bar{v}} \mid {}^{\ell}C_j^v\right)$

We denote the prior probability that source S_i makes a positive claim (i.e., claims a value T) by s_i^T and denote the prior probability that source S_i makes a negative claim (i.e., claims a value F) by s_i^F. We denote the prior probability that variable C_j is of value v by d^v. By the Bayesian theorem, we have:

$$ {}^{\ell}T_i^v = \frac{{}^{\ell}t_i \cdot s_i^v}{{}^{\ell}d^v}, \; {}^{\ell}F_i^v = \frac{(1 - {}^{\ell}t_i) \cdot s_i^{\bar{v}}}{{}^{\ell}d^v}. \tag{4} $$

Table 1 summarizes the introduced notations.

The dependencies between variables are given by a Bayesian network. In the underlying dependency structure of the Bayesian network (i.e., a DAG, denoted by G), each vertex V_j corresponds to a variable C_j, and each arc $V_j \to V_k$ corresponds to a causal relation between variables C_j and C_k in which the value of C_k is dependent on that of C_j. For any variable C_j, we use $\mathrm{par}(C_j)$ to denote the set of variables on whom the value of C_j directly depends (i.e., not including transitive dependencies). Since the causal relation is encoded in the Bayesian network, G, there is an arc from each node denoting a variable in $\mathrm{par}(C_j)$ to the node denoting C_j in G. Please note that each node V_j in the Bayesian network is associated with a probability function that takes, as input, a particular set of values of $\mathrm{par}(C_j)$, and gives, as output, the probability of C_j being true. In other words, given the Bayesian network, we know the conditional probability $\Pr\left({}^{\ell}{C_j}^v | \mathrm{par}(C_j)\right)$ for any event variable C_j. We assume that the Bayesian network is known from application context (e.g., we might have a map that says which outlet depends on which suppliers), or can be empirically learned from historic data by the algorithms such as those introduced in [13]. Hence, our estimation algorithm assumes that the Bayesian network is an input.

Finally, we formulate our reliable social-sensing problem as follows: *Given a source-claim matrix SC, a category label ℓ for each reported variable, and a Bayesian network G, encoding the dependencies among variables, how to jointly estimate both the reliability of each source and the true value of each variable in a computationally efficient way?* Here an algorithm is defined as efficient if its time complexity is sublinear to the exponential (i.e., $o(2^{|G|})$ for a Bayesian network that is not a complete graph, where $|G|$ is the total number of nodes in the Bayesian network.

3. GENERALIZATION OF PREVIOUS MODELS

Before we propose our estimation algorithm, in this section, we show that our social-sensing model is more general than those proposed in our previous work [22, 20, 21, 23]. In other words, the social-sensing models proposed by the previous work are all special cases of ours. Therefore, thanks to the general model, the estimation algorithm proposed in our paper can be directly applied to any of the problems defined in the previous work.

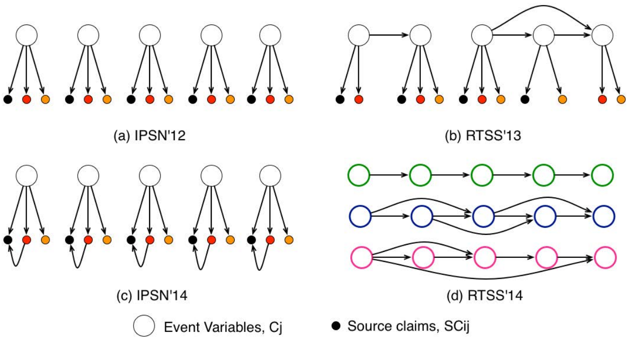


Figure 1: Model connections with previous work.

First, we show that the model proposed by Wang et al. [22] is a special case of our model. In their model, both the event variables and the sources were assumed independent. Thus, the structure of a Bayesian network for this model is just a DAG with arcs only connecting the event variable and its corresponding claims from the sources, as shown in Figure 1(a).

In [20], the model was extended to consider physical constraints of the sources (i.e., a variable might not be observed by some source), as well as correlated variables that fall into a bunch of independent groups. The structure of a Bayesian network for this model has disjoint cliques (complete subgraphs) where each clique has a constant number of nodes, as shown in Figure 1(b). In this figure, there are two cliques; one has two nodes and the other has three. Furthermore, since the physical constraints of the sources are considered, there are some variable that can only be observed by a subset of sources. Therefore, in the corresponding Bayesian network, if the variable is not observed by some source, then there is no arc between them in the DAG (such as the rightmost one that is observed only by the red source and the orange source).

Source dependencies were considered in [21], where a claim made by a source can either be original or be re-tweeted from some other source. The variables are assumed independent. The corresponding Bayesian network for this model is shown as in Figure 1(c). If a source i is dependent on some other source j, which means that the claim made by j actually affects that made by i as shown in Figure 1(c). In Figure 1(c), the black source is dependent on the red source, therefore there is an arc from the red node to the black node for each event variable. The arc from the $SC_{j\cdot}$ to $SC_{i\cdot}$ is enough to model this dependence.

Recently, Wang et al. [23] further extended the previous model by considering time-varying ground truth, in which the value of each variable could vary over time. They proved that given the evolving trajectory of each variable, by considering a sliding window of past states, the estimation result is greatly improved compared with estimators that only consider the current state. Their model can be represented by a dynamic Bayesian network with time-varying dependency structures. Figure 1(d) gives an example of a Bayesian network representation of their model. Here, we omit the vertices corresponding to claims made by sources $SC_{i,j}$. In the figure, the variable nodes with the same color are corresponding to a variable in different time-slots. The evolving trajectory of each variable can be represented by some dependency structure among all its history states, as shown in Figure 1(d).

The above examples illustrate how previous models can be special cases of our model. Therefore, once we solved the problem with the general model, using the same algorithm, we are able to solve all the previous problems as defined in [22, 20, 21, 23]. We propose our estimation algorithm in the following section.

4. ESTIMATING THE STATES OF INTERDEPENDENT VARIABLES

In this section, we describe our ground truth estimation algorithm for social-sensing applications with the interdependent variables at scale. Our algorithm follows the Expectation-Maximization (EM) framework [4] that jointly estimates (1) the reliability of each source, and (2) the ground truth value of each reported variable. Here we assume that sources independently make claims; for dependent sources, we can apply the algorithm proposed in [21]. We call the proposed algorithm EM-CAT (EM algorithm with CATegory-specific source reliability.

4.1 Defining Estimator Parameters and Likelihood Function

EM is a classical machine-learning algorithm to find the maximum-likelihood estimates of parameters in a statistical model, when the likelihood function contains latent variables [4]. To apply the EM algorithm, we first need to define the likelihood function $L(\theta; x, Z)$, where θ is the parameter vector, x denotes the observed data, and Z denotes the latent variables. The EM algorithm iteratively refines the parameters by the following formula until they converge:

$$\theta^{(n+1)} = \arg\max_{\theta} E_{Z|x,\theta^{(n)}}[\log L(\theta; x, Z)] \tag{5}$$

The above computation can be further partitioned into an *E-step* that computes the conditional expectation of the latent variable vector Z (i.e., $Q(\theta) = E_{Z|x,\theta^{n}}[\log L(\theta; x, Z)]$), and an *M-step* that finds the parameters θ that maximize the expectation (i.e., $\theta^{(n+1)} = \arg\max_{\theta} Q(\theta)$). In our problem, we define the parameter vector θ as:

$$\theta = \{({}^{\ell}T_i^v, F_i^v) | \forall i \in \mathcal{S}, v \in \Lambda, \ell \in \mathcal{L}\}$$

where $\Lambda = \{T, F\}$ denotes the set of binary values and $\mathcal{L}$ is the set of event categories. The data x is defined as the observations in the source-claim matrix SC, and the latent variable vector Z is defined as the values of the event variables.

After defining θ, x and Z, the likelihood function is derived as follows:

$$\begin{aligned} & L(\theta; x, Z) = \Pr(x, Z|\theta) = \Pr(Z|\theta)\Pr(x|Z; \theta) \\ & = \Pr(Z_1, \cdots, Z_N) \cdot \Pr(x_1, \cdots, x_N | Z_1, \cdots, Z_N; \theta) . \end{aligned} \tag{6}$$

Here $N = |\mathcal{C}|$ is the number of event variables, and x_j denotes all the claims made by the sources about the j-th variable. In Equation (6), $\Pr(Z|\theta) = \Pr(Z)$ because the joint probability of the event variables $\Pr(Z)$ is independent from the parameters θ.

Next, we are going to simplify the likelihood function by proving that for any event variables j_1 and j_2, x_{j_1} and x_{j_2} are conditionally independent given the latent variables Z. We use $X \perp\!\!\!\perp Y$ to denote that X and Y are independent, and similarly $X \perp\!\!\!\perp Y|Z$ to denote that X and Y are conditionally independent given Z. Before proving $x_{j_1} \perp\!\!\!\perp x_{j_2}|Z$, we first introduce the definition of d-separation.

Definition 1 (d-separation). *Let G be a Bayesian network, and $X_1 \rightleftharpoons \cdots \rightleftharpoons X_n$ be a trail in G. Let Z be a subset of the observed variables. The trail $X_1 \rightleftharpoons \cdots \rightleftharpoons X_n$[1] is active given Z if*

- *Whenever we have a V-structure $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$ in the trail, then X_i or one of its descendants are in Z, and*
- *no other node along the trail is in Z.*

If for any trail between X_1 and X_n is not active, then X_1 and X_n are d-separated in G by Z [10].

Here a trail between X_1 and X_n is an undirected path that is computed by simply ignoring the directions of the directed edges in the Bayesian network G. Note that if X_1 or X_n is in Z, the trail is not active. Next, we introduce a classical lemma showing that the d-separation implies conditional independence.

Lemma 1. *If X_i and X_j are d-separated in the Bayesian network G given Z, then $X_i \perp\!\!\!\perp X_j|Z$ [10] .*

Now we are ready to prove that x_{j_1} and x_{j_2} are conditionally independent given the latent variables Z for any $j_1, j_2 \in \mathcal{C}$ in Theorem 1.

Theorem 1. *For any pair of event variables j_1 and j_2, x_{j_1} and x_{j_2} are conditionally independent given the latent variables Z, i.e., $\forall j_1, j_2 \in \mathcal{C}$, $x_{j_1} \perp\!\!\!\perp x_{j_2}|Z$.*

Proof. To prove the theorem, we first need to define the causal relationship between a claim $SC_{i,j}$ and the value C_j of event j. Obviously, the value C_j of the event is independent of how a source claims it, but the claim $SC_{i,j}$ made by a source does rely on the value C_j of event j. Therefore, it is clear this causal relationship between $SC_{i,j}$ and C_j should be modeled by an arc from Z_j to $x_{i,j}$ in the Bayesian network, as illustrated in Figure 2. Here we do not distinguish the variable Z_j and its corresponding vertex in G, and the same for $x_{i,j}$ and its corresponding vertex.

Therefore, for any pair of x_{i_1,j_1} and x_{i_2,j_2}, and for whatever event dependency graph G of the event variables, we can find two vertices Z_{j_1} and Z_{j_2} in G such that all the trails have the same structure: $x_{i_1,j_1} \leftarrow Z_{j_1} \rightleftharpoons \cdots \rightleftharpoons Z_{j_2} \rightarrow x_{i_2,j_2}$, as shown in Figure 2. Since Z_{j_1} and Z_{j_2} are in $Z = \{Z_1, \cdots, Z_N\}$, and by Definition 1, we know that x_{i_1,j_1} and x_{i_2,j_2} are d-separated by Z. Please note that this d-separation is valid for any pair of sources i_1 and i_2, thus $x_i \perp\!\!\!\perp x_j|Z$ by Lemma 1. □

By Theorem 1 and the independent source assumption, the likelihood function in (6) can be simplified as:

$$L(\theta; x, Z) = \Pr(Z_1, \cdots, Z_N) \prod_{j \in \mathcal{C}} \prod_{i \in \mathcal{S}} \Pr(x_{i,j} | Z_j; \theta) . \tag{7}$$

[1] We use $X \rightarrow Y$ to denote the directed edge (arc) that points from X to Y in G, and $X \rightleftharpoons Y$ to denote the arc that connects X and Y whose direction, however, is not of interest.

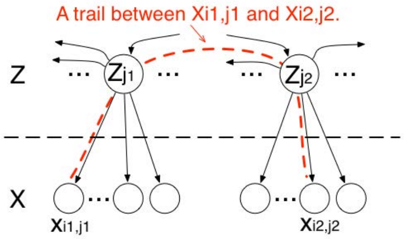


Figure 2: An illustration of the Bayesian network.

4.2 The EM Algorithm

Given the likelihood function, following (5), we can derive the EM algorithm. We omit the detailed mathematical derivations here since it is a standard procedure, and directly show the final results of how to update the parameters $\theta = \{{}^{\ell}T_i^v, {}^{\ell}F_i^v | \forall i \in \mathcal{S}, v \in \Lambda, \ell \in \mathcal{L}\}$ in (8).

$$\begin{aligned} {}^{\ell}T_i^{v\,(n+1)} &= \frac{\sum_{j \in {}^{\ell}\mathcal{C}_i^v} \Pr\left(Z_j = v | x; \theta^{(n)}\right)}{\sum_{j \in {}^{\ell}\mathcal{C}} \Pr\left(Z_j = v | x; \theta^{(n)}\right)}, \\ {}^{\ell}F_i^{v\,(n+1)} &= \frac{\sum_{j \in {}^{\ell}\mathcal{C}_i^{\bar{v}}} \Pr\left(Z_j = v | x; \theta^{(n)}\right)}{\sum_{j \in {}^{\ell}\mathcal{C}} \Pr\left(Z_j = v | x; \theta^{(n)}\right)}. \end{aligned} \tag{8}$$

In (8), $\theta^{(n)}$ denotes the parameters in the n-th iteration of the EM algorithm, ${}^{\ell}\mathcal{C}$ denotes the set of event variables with label ℓ, and ${}^{\ell}\mathcal{C}_i^v$ is a subset of ${}^{\ell}\mathcal{C}$ with each element that the source i claims its value being v (i.e. ${}^{\ell}\mathcal{C}_i^v = \{j | SC_{i,j} = v, L(j) = \ell\}$, where $L(j)$ denotes the label of event variable j).

In Equation (8), the key step for refining the parameters is to compute $\Pr\left(Z_j = v | x; \theta^{(n)}\right)$ for each $j \in \mathcal{C}$ and $v \in \Lambda$. Once we have computed this value, the rest of the computation for updating the parameters becomes trivial. Since we are using a Bayesian network to encode the dependences between variables, we know the conditional probability for each variable given a particular set of values of its parent variables. These are given as an input of our algorithm. Hence, we can compute $\Pr\left(Z_j = v | x; \theta^{(n)}\right)$, the marginal probability of variable Z_j given the evidence x. Similarly, $\Pr\left(Z_j = v | x; \theta^{(n)}\right)$ can be computed. The pseudocode of our estimation algorithm is given in Algorithm 1.

5. EVALUATION

In this section, we study the performance of our EM-CAT algorithm through extensive simulations as well as a real-world data set. While empirical data is always better, such data often constitutes isolated points in a large space of possible conditions. The simulation, in contrast, can extensively test the performance of our algorithm under very different conditions that are impractical to cover exhaustively in an empirical manner. The limitation of simulation is that read-world data might not follow exactly the assumed model. Therefore, we also evaluate our algorithm with a real-world data set. Evaluation results show that the new algorithm offers better estimation accuracy compared to other state-of-the-art solutions.

5.1 Simulation Study

In this simulation study, we build a social-sensing simulator in Matlab R2013b. For Bayesian network inference, we exploit an existing Bayesian network toolbox developed by Kevin Murphy [8]. Below we present our simulation setup and results.

Algorithm 1 EM-CAT: Expectation-Maximization Algorithm with Category-specific Source Reliability

Input: The source-claim matrix SC, the Bayesian network G, and event category $\ell \in \mathcal{L}$ for each event $j \in \mathcal{C}$.
Output: The estimated variable values, and the reliability vector of each source.

```
Initialize $\theta^{(0)}$ with random values between 0 and 0.5.
$n \leftarrow 0$
repeat
    $n \leftarrow n + 1$
    for Each $j \in \mathcal{C}$, each $v \in \Lambda$ do
        Compute $\Pr\left(Z_j = v | x; \theta^{(n)}\right)$ from the Bayesian network $G$.
    end for
    for Each $i \in \mathcal{S}$, each $v \in \Lambda$ and each label $\ell \in \mathcal{L}$ do
        Compute ${}^{\ell}T_i^{v\,(n)}$ and ${}^{\ell}F_i^{v\,(n)}$ from (8)
    end for
until $\theta^{(n)}$ and $\theta^{(n-1)}$ converge
for Each $j \in \mathcal{C}$ and $v \in \Lambda$ do
    $Z(j, v) \leftarrow \Pr\left(Z_j | x; \theta^{(n)}\right)$.
    if $Z(j, T) > Z(j, F)$ then
        Assign variable $j$ with value $T$
    else
        Assign variable $j$ with value $F$
    end if
end for
for Each $i \in \mathcal{S}$, each category $\ell \in \mathcal{L}$ do
    Compute its reliability ${}^{\ell}t_i$ from (4)
end for
```

5.1.1 *Methodology*

We simulated 100 interdependent binary variables. The underlying dependency graph is a random DAG that changes in each simulation run. The Bayesian network is created with the dependency structure defined by the DAG and parameters randomly generated using the toolbox. The expected ground truth for all variables is set to 0.5 (i.e., with probability 0.5, the variable will be True). The actual (marginal) probability distribution for each variable is defined by the Bayesian network.

The ground truth values of variables are generated based on the Bayesian network in a topologically-sorted order. That is, we wait to generate the value of variable v until the values of all of its parents, par(v), have been generated. Therefore, the ground truth value distribution of our variables follows the Bayesian network. Each event variable is also assigned a label ℓ randomly from a label set $\mathcal{L}$ to simulate the event category.

The simulator randomly assigns a reliability vector for each source. We randomly select a set of the sources to be "experts" at some category. Hence, for each, we choose a category, ℓ, and give the source a high reliability value in reporting variables of that category. The other values in the reliability vector are assigned lower values, making the average reliability of each source roughly the same. We use t_i to denote the average reliability of source S_i. In the simulation, t_i is in the range $(0.5, 1)$. We also simulate the "talkativeness" of the sources, which denotes the probability that a source would make a claim, denoted by s_i.

The source-claim matrix SC is then generated according to the reliability vector of each source, t_i, and the talkativeness of each source, s_i. For each source S_i and each event variable C_j, we first decide whether the source will make a claim about the variable by flipping a biased coin with probability s_i that the source will claim something. If it does not claim, then $SC_{i,j} = Unknown$, otherwise we generate the value of $SC_{i,j}$ based on T_i^v and F_i^v which can be computed from the reliability vector of the source.

The source-claim matrix SC is the evidence x in the Bayesian network. To include the claim nodes in the Bayesian network, we extend the DAG such that for each vertex V_j in the DAG G we create one vertex V_j' and add an arc (V_j, V_j') to G. We tried to add one vertex $V_{i,j}$ for each $x_{i,j}$ and directly set the parameters of $V_{i,j}$ to the corresponding T_i^v and F_i^v in the parameter vector $\theta^{(n)}$. This implementation is straightforward. However, it adds too many extra (that is $|\mathcal{S}| \times |\mathcal{C}|$) vertices to the Bayesian network, which greatly slows down the inference computation. Therefore, in our implementation, we just add one vertex V_j' for each variable V_j in G, and set the evidence (the observed value) of V_j' to False (which means $\Pr\left(V_j' = False|V_j\right) = p(x_j|Z_j;\theta^{(n)}) = \prod_{i\in\mathcal{S}} p(x_{i,j}|Z_j;\theta^{(n)})$). In this implementation, we only double the size of the DAG, which makes the inference computation of the Bayesian network much more efficient.

The default values of the simulation parameters are as follows: the number of sources is 40, the expected source (average) reliability t_i is 0.6, the talkativeness of the source s_i is 0.6. The number of event variables is set to 100, and we randomly generate the Bayesian network parameters such that, in expectation, the probability that each variable is True, is set to 0.5. The number of edges in the Bayesian network is 100. There are 2 categories by default.

We compare our algorithm to the algorithm proposed in [22] and two intermediate extensions towards the current solution. We also include a simple baseline. We use **EM-CAT** to denote our algorithm, and **EM-REG** to denote the algorithm proposed in [22]. Note that, EM-REG assumes that variables are independent, and all the variables share the same category (i.e., it assumes that there is only one category). The first extension of EM-REG is to add the Bayesian dependency structure to the event variables. We call this extension **EM-T** (EM algorithm with sTructed variables). The second extension of EM-REG is to consider event categories, that is called **EM-C**. The simple baseline algorithm is just voting, and is denoted by **VOTING**. VOTING estimates each variable to be equal to the majority vote. Each simulation runs 100 times and each result is averaged over the 100 executions.

5.1.2 Evaluation Results

Figure 3 shows the performance of our EM-CAT algorithm as the number of sources varies from 20 to 80. In Figure 3(a), we observe that our EM-CAT algorithm has the lowest estimation error, and EM-T and EM-C work better than the regular EM algorithm which is better than simple baseline voting. The reason is that when the underlying event variables follow some dependency structure, exploiting this piece of information will result a better estimator. EM-CAT also considers the category-specific reliability of each source. For each event category, EM-CAT will always select the sources with higher reliability for the category. Therefore, it achieves higher accuracy. Please note that as the number of sources increases, the accuracy of all the estimators increases. More data sources will result in more data. Therefore, the accuracy of the learning algorithm will be improved.

Figure 3(b) shows the error in estimating source reliability. Both the EM-CAT and EM-C are better in estimating source reliability than the other two algorithms. The reason is that the other algorithms ignore event categories. Thus, the information regarding differences in source reliability across different categories of observations is not exploited.

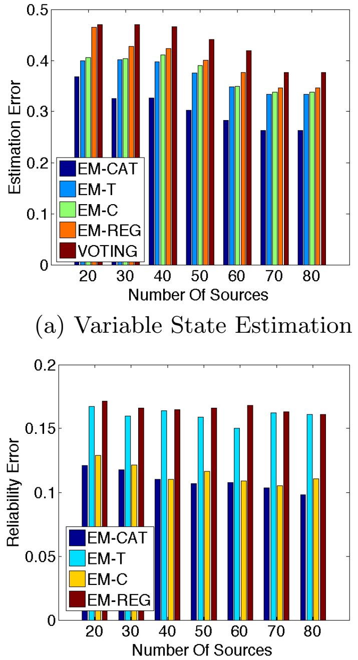
Figure 4 shows the performance of the estimators as a function of source reliability. From the figure, we observe that with more reliable sources, the accuracy of the estimators is greatly improved. Even the voting can result in very reliable estimates when source reliability is 0.9 or above (i.e., 90% of their reports are true). Among all the estimators, our new EM-CAT is the best at both estimating the ground truth values of reported variables and the reliability of sources.

Figure 5 explores the effect of "talkativeness," s_i, of the sources on estimation accuracy. As mentioned earlier, the talkativeness of a source denotes the probability that the source will make a claim regarding some variable. In the experiment, talkativeness is varied from 0.4 to 0.8. With higher talkativeness, we have more data. This is the reason why accuracy of the estimators improves as s_i increases. Again, our EM-CAT algorithm is the best among all the estimators.

In Figure 6, we study the performance of the estimators when the number of edges in the dependency structure (a DAG) varies. A larger number of edges in the DAG means more dependencies among variables. Figure 6 shows that performance of the state estimators does not change much with the number of dependencies. The reason could be that the parameters of the Bayesian network are generated *uniformly* at random in $(0, 1)$. Therefore, in expectation, the bias of the ground truth is around 0.5. However, if the bias of the ground truth is skewed, we will observe a difference among different dependency structures, since the value of a variable will be affected more by its depended variables. Our algorithm is the best among all the algorithms since we exploit the dependency structure of the variables. Although the accuracy of estimators does not vary much as the structure changes, exploiting this information leads to a more accurate state estimator.

We study the performance of the estimators as the number of category labels varies from 2 to 5 in Figure 7. From Figure 7(a), we observe that as the number of labels increases, the performance of EM-CAT becomes worse. This is because we end up with fewer and fewer data in each category. With fewer data, the parameters of the estimator cannot be learned accurately. Therefore the performance of estimation degrades. This figure suggests that when the data size is small, it is better to ignore category, but with a large data size, it would be better to exploit it.

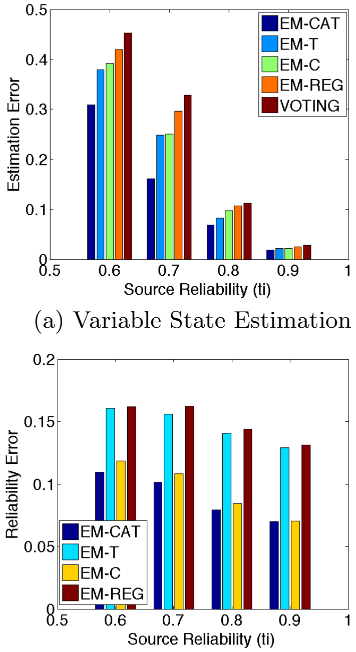
Figure 8, we study the performance of the estimators as the number of variables varies from 80 to 110. From the figure, we observe that the accuracy of the estimators improves as the number of variables increases. The reason is that we have more data to learn the estimation parameters more accurately. Actually, the voting algorithm does not vary much, since in voting each source has the same weight as the others. Therefore, even when the number of variables increases, the weight of the sources does not change, leaving performance the same. Again, our algorithm EM-CAT



(a) Variable State Estimation

(b) Source Reliability Estimation

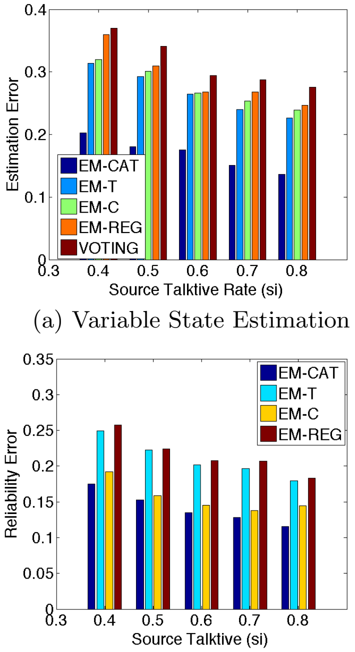
Figure 3: Performance as the number of sources varies.



(a) Variable State Estimation

(b) Source Reliability Estimation

Figure 4: Performance as the source reliability varies.



(a) Variable State Estimation

(b) Source Reliability Estimation

Figure 5: Performance as the source talkativeness varies.

is the best among all the algorithms in both the estimation of ground truth values of variables and estimation of source reliability.

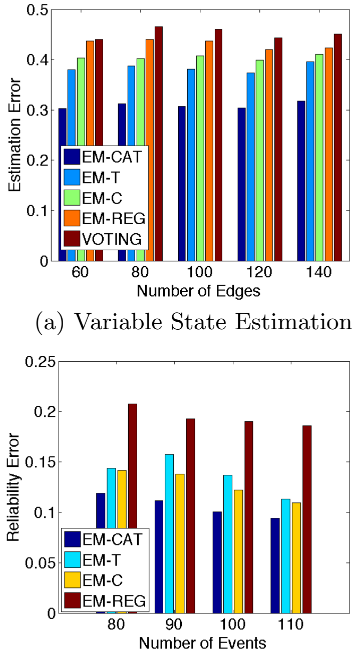
Next, we study the scalability of our EM-CAT algorithm. We mainly compare our algorithm with the algorithm proposed in [20]. Their algorithm considers the full joint distribution of all the correlated variables.

We compare three inference algorithms: 1. the **junction tree algorithm (JTree)**, 2. the **variable elimination algorithm (VarElim)**, and 3. the method used in [20], i.e., inference with the **full joint distribution (Total)**. Please note that all the three algorithms compute the exact inference probility of the Bayesian network, there are also algorithms that compute the approximate inference probability [10] which compromises the inference accuracy but has a better computational complexity.

In Figure 9, we fixed the expected node degrees to be 2, and varied the number of nodes in the Bayesian network from 10 to 25. Note that, the y-axis is in log-scale. From Figure 9, we clearly observe that the computation time of the Total algorithm increases exponentially as the number of nodes increase linearly. The computation time of both the JTree algorithm and the VarElim algorithm grows in a much less rapid way. The time complexity of both the algorithms actually depends on the size of the largest clique (the complete sub-graph in the Bayesian network); since the expected node degree is 2, it is possible that the resulting graph has a clique of size n, where n is less than the total number of nodes N in the Bayesian network. As the total number of nodes N increases, the gap between n and N will also increase as shown in the Figure 9. The JTree algorithm is more efficient than the VarElim algorithm, since it maintains a data structure which can simultaneously update the potential of local cliques but the the VarElim algorithm eliminates the variables sequentially. Furthermore, the time complexity of the VarElim algorithm also depends on the order of the variables to eliminate. It is NP-hard to find the optimal order based on which the VarElim algorithm eliminates the variables. When the total number of nodes N is 25, we can observe that the Total algorithm needs 20 seconds in average, while VarElim needs 2 seconds and JTree needs only 0.2 seconds in average. We can use JTree algorithm in our EM-VTC algorithm for the Bayesian inference computation, compared with the previous solution that does not exploit the dependence structure of the variables [20] (i.e. using the Total algorithm), it is not hard to observe how scalable our algorithm is as the number of total variables varies.

Figure 10 shows the CPU time of the three inference algorithm when the number of nodes is fixed at 24 but the expected degree of each node varies from 1 to 3. From the figure, we can observe that the CPU time of the VarElim algorithm increases as the number of node degree increases. This is because that with a larger node degree, the chance that the VarElim algorithm selects a bad variable eliminating order become larger. Thus, the time complexity of VarElim increases. However, the time complexity of the JTree algorithm only depends on the size of each local clique, when the node degree is small such as less than 3 the complexity of JTree actually changes very little which does not show in Figure 10. Figure 10 shows the scalability of our algorithm compared with previous solution [20] as the expected node degree varies in the Bayesian network.

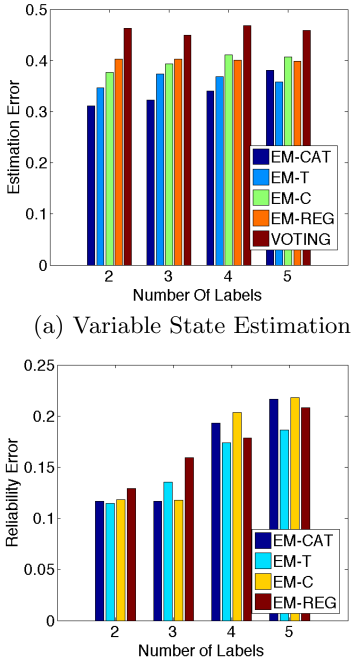
Next, we evaluate our algorithm using data from a real disaster scenario. In addition to estimation accuracy, we evaluate its efficacy at hypothesis testing.



(a) Variable State Estimation

(b) Source Reliability Estimation

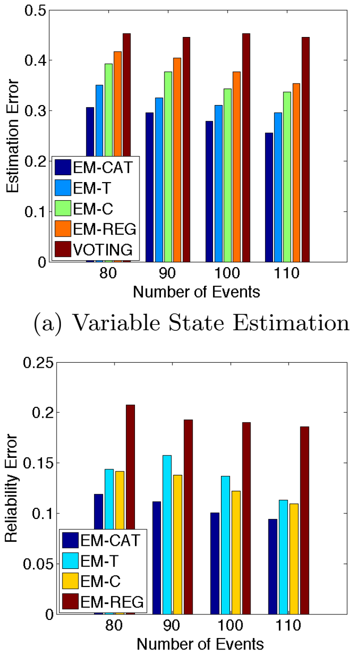
Figure 6: Performance as the number of edges in the Bayesian network varies.



(a) Variable State Estimation

(b) Source Reliability Estimation

Figure 7: Performance as the number of event categories varies.



(a) Variable State Estimation

(b) Source Reliability Estimation

Figure 8: Performance as the number event variables varies.

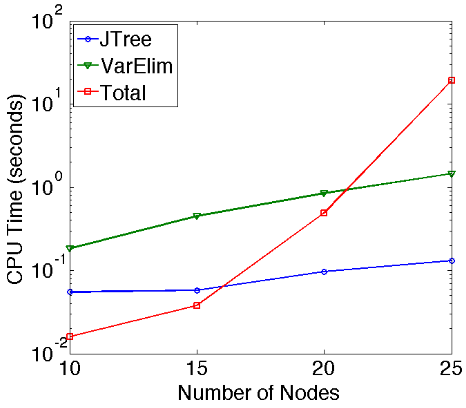


Figure 9: Computation time comparison with fixed node degree.

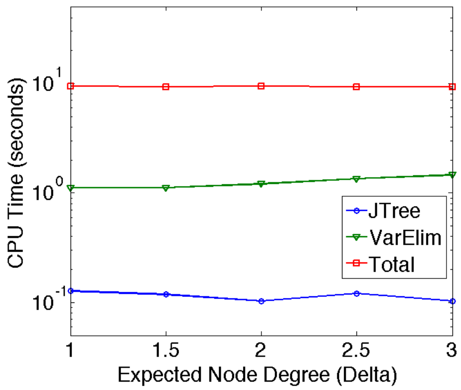


Figure 10: Computation time comparison with fixed number of nodes.

5.2 Performance with Real Data

Here, we test the estimation accuracy of our algorithm with a real-world data set about the availability of groceries, pharmacies, and gas stations during Hurricane Sandy in November 2012[2]. Sandy was reported as the second-costliest hurricane in the history of the United States (surpassed only by hurricane Katrina). It caused widespread shortage of gas, food, and medical supplies, as gas stations, grocery retail shops and pharmacies were forced to close. Some were closed for as long as a month. The data set was documented by the All Harzard Consortium (AHC)[2], a state-sanctioned non-profit organization fucused on homeland security, emergency management, and business continuity issues in the mid-Atlantic and northeast regions of the US. The data covered states including WV, VA, PA, NY, NJ, MD, and DC, and the information was updated daily. Figure 11 shows a portion of the location distribution of the gas stations in the data set.

5.2.1 *Methodology*

In this evaluation, the reported variables denote whether a given gas station has gas, whether a given grocery store is open, and whether a given pharmacy is open on a given

[2] The data set is available at: http://www.ahcusa.org/hurricane-Sandy-assistance.htm

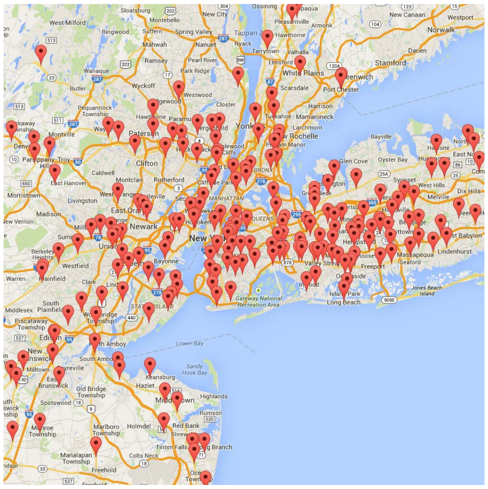


Figure 11: The location distribution of the gas station data.

day following the hurricane. Hence, the total number of variables is equal to the total number of gas stations, grocery shops and pharmacies, combined, multiplied by the period of observation in days. All variables are binary. The ground truth for these variables is given in the AHC data set. The experiment has two goals. The first goal is to test the accuracy of our algorithm at reconstructing the ground truth values of these variables from observations reported by unreliable sources. The second goal is to determine which of several hypothesized dependency structures among the variables is borne out by data reported by these unreliable sources (which is an instance of hypothesis testing). In particular, it is interesting to see whether a hypothesis that corresponds to the ground-truth dependency structure will actually be picked out despite observation noise. Since we had ground truth data only, we added noise artificially by simulating 100 unreliable sources. The average reliability of a source was set to 0.7. The expected talkativeness of a source was set 0.8. Obsevations reported by these simulated sources were then used.

The hypothesis testing experiment exemplifies the type of analysis a decision-maker might perform to understand the cause of failures in a large system. For example, when power is lost or restored, when flooding occurs or is drained, and when available resources are depleted in a neighborhood due to local demand, correlated changes in the state of our aforementioned variables occur. The dependency structure among these variables depends on the cause of outage. This observation allows us to compare hypotheses regarding the cause of failure. Each hypothesis would correspond to a different dependency graph between failures. A dependency graph that results in a higher value for the likelihood function when the EM algorithm converges would imply that the corresponding hypothesis is better supported by data. Hence, by comparing the converged values of the likelihood functions when running EM with different hypothesized dependency graphs, we could determine which hypothesis is more likely to be the case. For illustration, we propose three hypotheses regarding the dependency structures among the observed variables in the hurricane Sandy scenario:

1. *Independent hypothesis:* This hypothesis trivially states that the variables are independent.

2. *Supply line hypothesis:* This hypothesis states that all variables located in the same state are connected by a directed path; the supply line.[3]

3. *Exact hypothesis:* Prior work on the same data set [5] did identify the exact dependency between our variables by observing which variables tend to change together. This real dependency structure was computed based on ground truth data. We include it here to test our EM algorithm. The algorithm, if correct and robust to the noise introduced by less reliable sources, should generate a higher final value for the likelihood function when this hypothesis is used for the dependency structure.

These three hypotheses are named **Indep**, **SupLine**, and **CoJump**, respectively. We test which one is the most probable. Our hypotheses lead to different DAGs from which we build the corresponding Bayesian networks. In our evaluation, we randomly select four days in November, 2012, and test our hypotheses based on data from these four days.

In addition to choosing a winning hypothesis, we study the performance of our EM-CAT algorithm in terms of estimation errors in the state of gas stations, pharmacies, and groceries, and error in estimating the reliability of the simulated sources. The evaluation is averaged over 20 executions to smooth out the noise.

5.2.2 Evaluation Results

Figure 12 shows the evaluation results on the pharmacy data. In this figure, we observe that the Indep hypothesis and the SupLine hypothesis lead to an inferior estimation accuracy compared to the CoJump hypothesis. This is consistent with the fact that the CoJump hypothesis did coincide with the real dependency structure seen in the data.

The estimation errors of the three hypotheses with our EM-CAT algorithm for the data of grocery retail stores are shown in Figure 13. We can observe that the CoJump hypothesis is the best of the three. SupLine works better than Indep on the data generally.

The evaluation results for the gas station data is shown in Figure 14. In general, the CoJump hypothesis is better than the other two.

The above results show that the “right” hypothesis regarding the dependency structure among variables does in fact result in better estimation accuracy, but how would a user know which of multiple hypotheses is the right hypothesis? The answer, as mentioned earlier, lies in computing the converged value of the likelihood function (when EM terminates) when dependencies among variables are given by each of the compared hypotheses. The hypothesis corresponding to the highest value of likelihood is the best hypothesis. These values are summarized in Table 2 for the hypotheses described above. From the table, we observe that the CoJump hypothesis has the highest likelihood according to the data set. This is what we hoped to see. It shows that indeed

[3] Ideally, we should have considered the real topology of supply lines or supply routes connecting the retailers under consideration. However, we did not have access to this information, so we just assumed, for the sake of an example, that we knew the supply line topology and that it was as we defined.

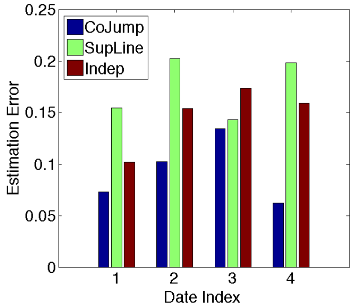


Figure 12: Estimation error with the pharmacy data.

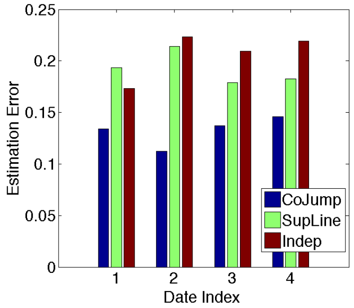


Figure 13: Estimation error with the grocery data.

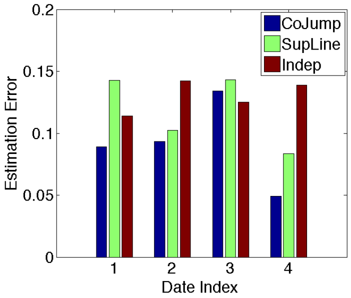


Figure 14: Estimation error with the gas station data.

comparing the converged values of the likelihood function computed by the EM algorithm identifies the right hypothesis regarding dependencies among variables, even in the presence of noise from unreliable observations. The evaluation illustrates use of the EM-CAT algorithm for hypothesis testing.

Table 2: Likelihood of the hypotheses for each data set.

	Pharmacy	Grocery	Gas Station
CoJump	0.83	0.74	0.79
SupLine	0.66	0.64	0.74
Indep	0.70	0.58	0.73

6. RELATED WORK

Inferring the structure of Bayesian Network is, in general, an NP-complete problem [3]. In order to learn a Bayesian Network in a tractable way, various algorithms are proposed. There are mainly two categories of approaches, score-based and constraint-based [19]. The former one tries to search for the optimum structure based on goodness-of-fit. The latter one utilizes conditional independence to build the network. Depending on the different data types and relationships, various hypothesis tests are available. For continuous data, if the relationship among variables is believed to be linear, tests based on Pearson's correlation are widely used. Asymptotic χ^2 tests can also be used to test independence between two continuous variables [6]. In cases of categorical variables, one of the most classical tests is Pearson's χ^2 test [1]. It works on the contingency table and tests if paired observations from two categorical variables are independent. In addition, likelihood-ratio statistic (or G^2) can also be used on either categorical or continuous variables; Jonckheere's trend test provides an independence test on ordinal variables [7]. Although in some cases separate test statistics can be used by different tests, they usually provide the same conclusions. If two variables are conditionally dependent, an edge between these two variables should be drawn in the Bayesian Network. Different from traditional work, where the inference is employed on the data provided by a single source, our proposed mechanism is able to conduct hypothesis testing upon crowdsourced data by accounting for a variety of sources of different and unknown reliability.

The problem studied in this paper bears some resemblance to the fact-finding problem that has been studied extensively in recent years. The goal of fact-finding, generally speaking, is to ascertain correctness of data from *sources of unknown reliability*. As one of the earliest efforts in this domain, Hubs and Authorities [9] presented a basic fact-finder, where the belief in a claim and the truthfulness of a source are computed in a simple iterative fashion. Later on, Yin et al. introduced TruthFinder as an unsupervised fact-finder for trust analysis on a providers-facts network [24]. Pasternack et al. extended the fact-finder framework by incorporating prior knowledge into the analysis and proposed several extended algorithms: Average.Log, Investment, and Pooled Investment [14]. Su et al. proposed semi-supervised learning frameworks to improve the quality of aggregated decisions in distributed sensing systems [16, 17]. Towards a joint estimation on source reliability and claim correctness, Wang et al. [22, 21] and Li et al. [11, 18] proposed expectation maximization and coordinate descent methods to deal with deterministic and probabilistic claims, respectively. Though yielding good performance in many cases, none of these approaches considers situations where the variables in question have wide-spread dependencies. To address this problem, Wang et al. [20, 23] further extended their framework to handle limited dependencies. However, their algorithm has exponential computational complexity in the number of correlated variables, and thus can only be applied in scenarios where the number of dependencies is small. In contrast to their work, we consider a model for more general scenarios where a considerable number of dependencies exists among the variables reported by the unreliable sources.

7. CONCLUSION

In this paper, we addressed the reliable crowd-sensing problem with interdependent variables. Crowd-sensing is a novel sensing paradigm in which human sources are treated as sensors. The challenge is that the reliability of sources is unknown in advance. Recently, several efforts tried to address this reliability challenge by formulating the problem given different source and event models. However, they did not address the problem when the reported variables are interdependent at large scale. In this paper, dependencies between reported variables were formulated as a Bayesian network. We demonstrated that our formulation is more general than previous work; previous models being special cases of ours. Evaluation results showed that our EM-CAT algorithm outperforms the state-of-the-art solutions. We also showed that the algorithm can be used for hypothesis testing on the dependency structure among variables.

Acknowledgments

We sincerely thank Jorge Ortiz for shepherding the final version of this paper, and the anonymous reviewers for their invaluable comments. Research reported in this paper was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement W911NF-09-2-0053, DTRA grant HDTRA1-10-10120, and NSF grants CNS 09-05014 and CNS 10-35736. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

8. REFERENCES

[1] A. Agresti. *An introduction to categorical data analysis*, volume 135. Wiley New York, 1996.

[2] All hazards consortium. http://www.ahcusa.org/.

[3] D. M. Chickering. Learning bayesian networks is np-complete. In *Learning from data*, pages 121–130. Springer, 1996.

[4] A. P. Dempster, N. M. Laird, D. B. Rubin, et al. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal statistical Society*, 39(1):1–38, 1977.

[5] S. Gu, C. Pan, H. Liu, S. Li, S. Hu, L. Su, S. Wang, D. Wang, T. Amin, R. Govindan, G. Aggarwal, R. Ganti, M. Srivatsa, A. Barnoy, P. Terlecky, and T. Abdelzaher. Data extrapolation in social sensing for disaster response. In *Proceedings of the 10th IEEE International Conference on Distributed Computing in Sensor Systems*. IEEE Press, 2014.

[6] R. I. Jennrich. An asymptotic $\chi 2$ test for the equality of two correlation matrices. *Journal of the American Statistical Association*, 65(330):904–912, 1970.

[7] A. R. Jonckheere. A distribution-free k-sample test against ordered alternatives. *Biometrika*, pages 133–145, 1954.

[8] Kevin Murphy. Bayes Net Toolbox for Matlab. https://code.google.com/p/bnt/.

[9] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[10] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[11] 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 2014 ACM SIGMOD international conference on Management of data*, pages 1187–1198. ACM, 2014.

[12] Mark Paskin. A short course on graphical models. http://ai.stanford.edu/ paskin/gm-short-course/.

[13] T. D. Nielsen and F. V. Jensen. *Bayesian networks and decision graphs*. Springer, 2009.

[14] J. Pasternack and D. Roth. Knowing what to believe (when you already know something). In *COLING*, 2010.

[15] C. S. Raghavendra, K. M. Sivalingam, and T. Znati. *Wireless sensor networks*. Springer, 2004.

[16] L. Su, J. Gao, Y. Yang, T. F. Abdelzaher, B. Ding, and J. Han. Hierarchical aggregate classification with limited supervision for data reduction in wireless sensor networks. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pages 40–53. ACM, 2011.

[17] 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*, pages 327–338, 2012.

[18] L. Su, Q. Li, S. Hu, S. Wang, J. Gao, H. Liu, T. Abdelzaher, J. Han, X. Liu, Y. Gao, and L. Kaplan. Generalized decision aggregation in distributed sensing systems. In *Real-Time Systems Symposium (RTSS)*, 2014.

[19] I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

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

[21] D. Wang, T. Amin, S. Li, T. A. L. Kaplan, S. G. C. Pan, H. Liu, C. Aggrawal, R. Ganti, X. Wang, P. Mohapatra, B. Szymanski, and H. Le. Humans as sensors: An estimation theoretic perspective. In *IPSN*, 2014.

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

[23] S. Wang, D. Wang, L. Su, L. Kaplan, and T. Abdelzaher. Towards cyber-physical systems in social spaces: The data reliability challenge. In *Real-Time Systems Symposium (RTSS)*, 2014.

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