A Semi-Supervised Active-learning Truth Estimator for Social Networks

Hang Cui
Department of Computer Science,
University of Illinois at
Urbana-Champaign
hangcui2@illinois.edu

Tarek Abdelzaher
Department of Computer Science,
University of Illinois at
Urbana-Champaign
zaher@illinois.edu

Lance Kaplan
Army Research Laboratories
Adelphi, MD
lance.m.kaplan.civ@mail.mil

ABSTRACT

This paper introduces an active-learning-based truth estimator for social networks, such as Twitter, that enhances estimation accuracy significantly by requesting a well-selected (small) fraction of data to be labeled. Data assessment and truth discovery from arbitrary open online sources are a hard problem due to uncertainty regarding source reliability. Multiple truth finding systems were developed to solve this problem. Their accuracy is limited by the noisy nature of the data, where distortions, fabrications, omissions, and duplication are introduced. This paper presents a semi-supervised truth estimator for social networks, in which a portion of inputs are carefully selected to be reliably verified. The challenge is to find the subset of observations to verify that would maximally enhance the overall fact-finding accuracy. This work extends previous passive approaches to recursive truth estimation, as well as semi-supervised approaches where the estimator has no control over the choice of data to be labeled. Results show that by optimally selecting claims to be verified, we improve estimated accuracy by 12% over unsupervised baseline, and by 5% over previous semi-supervised approaches.

CCS CONCEPTS

• **Information systems → Collaborative and social computing systems and tools**.

KEYWORDS

Social Sensing, Truth Discovery, Semi Supervision, Active Learning , Maximum Likelihood Estimation

ACM Reference Format:
Hang Cui, Tarek Abdelzaher, and Lance Kaplan. 2019. A Semi-Supervised Active-learning Truth Estimator for Social Networks. In *Proceedings of the 2019 World Wide Web Conference (WWW'19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313712

1 INTRODUCTION

This paper presents an active-learning based truth estimator for data collected on social media. We envision scenarios, such as disaster response, where a large number of unvetted local sources make claims about the state of events, possibly combining fact with rumor. A claim, in this paper, refers to the statement of a tweet, or a similar expression of fact (on social media) that may be correct or not. A major challenge in consuming such social media feeds lies in uncertainty regarding source reliability and thus difficulty assessing correctness of claims. Online sources who contribute the information are often unknown or have little prior reputation to consider. The problem becomes: is it possible to correctly filter true observations from the collective output of online sources of unknown reliability?

Our estimator computes, for each reported claim, the probability that the claim is correct. The estimator is novel in that it selects some well-chosen small subsets of claims for verification. We assume that another mechanism is available to verify the selected claims (i.e., give the estimator their ground-truth true/false label). Presumably, this external mechanism is costly and so has to be applied with care, which motivates the problem of selecting the best claims to verify from the perspective of maximally improving the overall accuracy of our estimator.

We extend a large body of work on fact-finding [5, 12, 14, 15, 19, 20] that estimates the reliability of sources iteratively with assessing the veracity of claims over the *entire* data set. We build on a state of the art estimator [3] that considers sources with significant error correlations. Those correlations arise from the *open* nature of online data. That is to say, sources are not independent. They can see each other's outputs and may occasionally copy others. Source non-independence creates opportunities for the spread of data contamination, as bad data (e.g., rumors) can be copied from one source to another. The non-independence of online open sources calls for estimators, such as the one used in this paper, that explicitly take source dependencies into account.

Finally, our estimator is unique in finding pivot observations whose true/false values, if verified, would maximally improve overall fact-finding accuracy. It selects a small set of pivot observations from the large volume of inputs for an external party (e.g., a human) to label. These labeled observations are then reinserted into the estimator to maximally improve its accuracy. The size of the pivot set can be either pre-defined or adjusted on the run. As shown in the evaluation, the above active selection of best observations to verify improves over semi-supervised approaches that do not have control over which claims are verified, or that use uncertainty in truth value alone as the basis for selection. To the authors' knowledge, ours is the first *semi-supervised* truth estimator for social media that *actively learns* the most judicious data claims to label (while accommodating non-independent sources).

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313712

We evaluate our truth estimator using both simulations and real data. We show that our approach significantly outperforms several baselines in a wide range of experimental conditions.

The rest of this paper is organized as follows. Section 2 describes the data model and problem statement. Section 3 presents the active-learning-based truth estimation algorithm. Section 4 evaluates the accuracy of estimation results using simulations. Section 5 presents empirical evaluation results. Observations and limitations are discussed in Section 6. Section 7 summarizes related work.

2 MODEL AND PROBLEM STATEMENT

Our active-learning truth estimator integrates social sensing data with human supervision to estimate the probability of correctness of individual claims. Unsupervised truth estimators of social network posts are constrained by uncertainty in reliability of arbitrary online sources. To improve accuracy over previous work, we allow for verifying a small faction (< 5%) of social media posts. Our algorithm adopts an active learning structure that automatically determines the best set of posts to verify.

Let the set of all data sources be denoted by $\mathcal{S}$, where $|\mathcal{S}| = n$. The observations they make constitute the set of assertions, denoted by C, where $|C| = m$. The ith source and the jth assertion are referred to as $S_i \in \mathcal{S}$ and $C_j \in C$, where $i \in \{1, 2, \cdots, n\}$ and $j \in \{1, 2, \cdots, m\}$. The act of making an assertion by a specific source is called a *claim*. If source S_i makes an assertion C_j, we say logically that S_i claims C_j = *True*, denoted by claim $S_iC_j = 1$. The same assertion can be made by multiple sources at different times. The set of all claims made by all sources is represented by source claim matrix $\mathcal{SC}$, where each element is of the form $S_iC_j = value$, where $value = 1$ if source S_i makes claim $C_j = 1$ and $value = 0$ otherwise. Note that, the only natural language processing needed in this paper is to identity similar claims made by multiple sources and cluster them into one assertion. In this paper, we use simple cosine similarity between tweets as a distance metric. Previous work suggested that cosine similarity is sufficient for Twitter data[18]. Hence, clusters of tweets that are sufficiently close by this metric are considered to make the same assertion.

Our active-learning truth estimator selects a set of assertions to label by an external (accurate) entity in order to improve accuracy. Note that, we only verify a small ratio (< 5%) of total assertions. Verification can incur human labor and cost, which should be kept as low as possible. Let us denote the set of labeled assertions as set $\bar{C}$, the size of labeled assertions as V, verification result as $\mathcal{V}$, in which $\mathcal{V}_j = 1$ and 0 means the verification result of assertion j is *True* and *False*, respectively. In this paper, we assume that human participants who verify $\bar{C}$ have full access to ground truth, such that all the labeled data are accurate. Our algorithm automatically estimates the best set of assertions to verify, which means the set $\bar{C}$ is not preset and is derived on the run.

Since we assume that sources are open (and hence their outputs can be seen by other sources), in general, a source may report something original or may copy an observation posted by others. This copying behavior is very common today on social media, causing fast spread of rumors and misinformation. If source S_a occasionally copies from source S_b, we say that S_a is *influenced* by S_b. Prior literature described algorithms for empirically detecting the existence of influence (manifesting in copying behavior) among sources [9]. Intuitively, these algorithms infer influence from unusual correlations among source outputs much in the same way cheating can be detected from correlated answers on exams. Since data are time-stamped, the direction of influence can be estimated, assuming that data published earlier influences data published later.

Table 1: List of Parameters

$\mathcal{S}$	Set of all sources
C	Set of all assertions
n	# sources
m	# claims
$\mathcal{SC}$	Source-claim matrix
$\bar{C}$	Set of labeled assertions
$\tilde{\bar{C}}$	Set of unlabeled assertions
$\mathcal{V}$	Set of verification results
V	Number of labeled assertions
$\mathcal{Z}$	Estimated values of assertions
$\mathbf{A}$	Source original claim probability matrix
$\mathbf{P}$	Source dependent claim probability matrix
G	Influence network
$\mathbf{D}$	Dependency matrix
$\boldsymbol{\theta}$	Set of unknown parameters

We therefore assume the existence of an observed influence network, G, where nodes represent individual sources and directional edges denote the direction of influence between them. Note that, the existence of an edge in G between a parent node and a child node does not mean that the child always copies from the parent. It merely means that *some* claims of the child may be copied from the parent.

A claim S_iC_j is called original if no ancestors of S_i in the influence network, G, made the same claim with an earlier timestamp. Otherwise, we say the claim is dependent (meaning it might have been copied because an ancestor of node S_i in the influence graph made the same claim earlier). We use the indicator D_{ij} to denote claim dependencies. Specifically, $D_{ij} = 0$ indicates that claim S_iC_j is original. $D_{ij} = 1$ means the claim is dependent.

We use $\tau(C_j)$ to denote the actual truth value of assertion C_j. The goal is to identify the truth value of all observed assertions and select the best set of assertions to verify, given the source claim matrix $\mathcal{SC}$ and influence graph G.

3 EXPECTATION MAXIMIZATION FRAMEWORK OVERVIEW

3.1 Solution Overview

The estimator uses an expectation maximization framework that maximizes a pre-defined likelihood function. The algorithm jointly estimates source reliability and claim correctness, then uses estimates to select a set of assertions for human participants to verify. The labeled assertions are then combined with unlabeled data to compute improved estimates via EM framework.

Let $\mathcal{SC}$ denote the set of claims made in the interested interval. $\mathcal{S}$ and C are the corresponding sets of sources (who reported observations) and measured assertions (for which values were reported in time-slot), respectively. Our goal is to estimate the truth value $\tau(C_j)$ for every assertion C_j, as well as to update the reliability of each source. We introduce two sets of parameters to characterize the behavior of sources. Specifically, for each source i:

- Let $\mathbf{A}^i$ denotes the *original claim probability* matrix, in which the entry $\mathbf{A}^i_{pq} = P(S_iC_j = p|\tau(C_j) = q, \mathbf{D}_{ij} = 0)$ denotes the probability that source S_i claims that assertion C_j has value p, when the ground truth value is q and the claim is original.
- Let $\mathbf{P}^i$ denotes the *dependent claim probability* matrix, in which the entry $\mathbf{P}^i_{pq} = P(S_iC_j = p|\tau(C_j) = q, \mathbf{D}_{ij} = 1)$ denotes the probability that source S_i claims that assertion C_j has value p, when the ground truth is q, and given that some ancestor of S_i in the influence graph, G, claimed $C_j = p$ earlier.

The estimated value of assertion C_j is denoted by the vector $\mathcal{Z}_j$. For a binary assertion, $\mathcal{Z}_j = \{z_{j,0}, z_{j,1}\}$, where $z_{j,q} = P(\tau(C_j) = q)$, is the probability that C_j has value q. The active-learning truth estimator proceeds in three stages:

- Estimating the true value of each assertion C_j and the reliability of each source S_i by EM framework: the estimator adapts an expectation maximization (EM) algorithm to jointly estimate reliability parameters of sources and truth values of variables from data. Given the claims made by sources, denoted by matrix $\mathcal{SC}$, and the source dependency matrix $\mathbf{D}$, the EM algorithm outputs the reliability for each source $S_i \in \mathcal{S}$ and the estimated truth value of each assertion $C_j \in C$. Note that, those estimates are unsupervised (No verification is involved in this step). They remain to be fused with a set of verified assertions, which are selected in next step.
- Selecting a set of assertions of size V to verify: Our truth estimator is blinded to ground truth (We do not have access to ground truth until we output the verification set). To estimate the optimal set of assertions to verify, we use the likelihood function as metric: the set which maximizes the likelihood function is selected. Note that, we do not have access to ground truth in this step, which means we have no information on verification results. Thus, for each assertion to verify, we need to estimate the effect of both *True* and *False.* Expected likelihood is then computed by averaging over all verification results. The probability of verification results is derived from truth estimates in step 1. The algorithm iterates through every possible sets of assertions to derive the optimal set. In later section, we also propose an heuristic approximation to dramatically reduce computation cost.
- Re-estimating truth value of assertions and reliability of sources with labeled data. After obtaining verification results from human participants, we combine labeled assertion with unlabeled ones to update our estimation via semi-supervised EM framework.

The above three steps are discussed in the following three subsections respectively.

3.2 Estimating Truth Value and Source Reliability via EM Framework[14]

Let us define the concatenation of source parameter matrices, $\mathbf{A}^i$ and $\mathbf{P}^i$, as the unknown parameter set $\boldsymbol{\theta}$ to be estimated. We adapt an expectation maximization algorithm to estimate these parameters based on matrix $\mathcal{SC}$ collected in the interested interval. The expectation maximization likelihood is formulated as log likelihood of observation matrix:

$$\theta_{MAX} = argmax_{\boldsymbol{\theta}}\{\log(\mathbb{P}(\mathcal{SC}|\boldsymbol{\theta}, \mathbf{D}))\} \tag{1}$$

The expected likelihood of observation matrix is formulated as multiplication of assertions:

$$\begin{aligned} \mathcal{L} = \mathbb{E}_{\boldsymbol{\theta}}[\mathbb{P}(\mathcal{SC}|\boldsymbol{\theta}, \mathbf{D})] &= \prod_{C_j \in C} \sum_{q=0}^{v} \mathbb{P}(SC_j, C_j = q|\boldsymbol{\theta}, \mathbf{D}) \\ &= \prod_{C_j \in C} \sum_{q=0}^{v} \mathbb{P}(SC_j|C_j = q, \boldsymbol{\theta}, \mathbf{D})\mathbb{P}(C_j = q|\boldsymbol{\theta}, \mathbf{D}) \end{aligned} \tag{2}$$

where v is the largest value for an assertion. For example, in the binary case, an assertion can take either the value 0 or 1. Hence, $v = 1$. SC_j is the jth column of SC, representing all the claims made on assertion j. $\mathbb{P}(SC_j|C_j = q, \boldsymbol{\theta}, \mathbf{D})$ represents the collective probability of all claims that said assertion $C_j = q$. It can be expressed as multiplication of all sources who made claims to assertion j:

$$\mathbb{P}(SC_j|C_j = q, \boldsymbol{\theta}, \mathbf{D}) = \prod_{S_i \in \mathcal{S}_j} \mathbb{P}(S_iC_j|C_j = q, \boldsymbol{\theta}, \mathbf{D}_{ij}) \tag{3}$$

where

$$\mathbb{P}(S_iC_j|C_j, \boldsymbol{\theta}, \mathbf{D}_{ij}) = \begin{cases} \mathbf{A}^i_{pq} & C_j = q, S_iC_j = p, \mathbf{D}_{ij} = 0 \\ \mathbf{P}^i_{pq} & C_j = q, S_iC_j = p, \mathbf{D}_{ij} = 1 \end{cases} \tag{4}$$

are the source reliability parameters.

Using E-step and M-step of standard EM-algorithm, we get the following iterative updates:

$$\begin{aligned} \mathcal{Z}_{j,p} &= \frac{B_p}{\sum_{q=1}^{v} B_q} \\ \mathbf{A}^i_{pq} &= \frac{\sum_{S_iC_j=p, \mathbf{D}_{ij}=0} z_{j,q}}{\sum_{S_iC_j, \mathbf{D}_{ij}=0} z_{j,q}} \\ \mathbf{P}^i_{pq} &= \frac{\sum_{S_iC_j=p, \mathbf{D}_{ij}=1} z_{j,q}}{\sum_{S_iC_j, \mathbf{D}_{ij}=1} z_{j,q}} \end{aligned} \tag{5}$$

where

$$B_p = \prod_{q=1}^{v} \prod_{S_iC_j=q} \mathbb{P}(S_iC_j|C_j = p) \tag{6}$$

The EM framework then iteratively updates the above formula until they converge.

3.3 Selecting Set of Assertions to Verify

Providing the estimated truth values via expectation maximization framework, we derive the set of verified assertions that maximize the expectation maximization likelihood. Let $\bar{C}$ denote the set of labeled claims, $\tilde{C}$ be the set of unlabeled claims, $\tilde{C} = C - \bar{C}$. $\mathcal{V}$ denotes the set of verification result, in which $\mathcal{V}_j \in \mathcal{V}$ is the verified

Algorithm 1 Semi-supervised Expectation Maximization

```
Input: Source-Claim matrix $SC$
Initialize $\boldsymbol{\theta}^{(0)}$, $\mathcal{Z}^{(0)}$, $\bar{\boldsymbol{\theta}}^{(0)}$, $\mathcal{L}$ and $\bar{C}$
while $\boldsymbol{\theta}^{(r)}$ does not converge do
    for $C_j$ in $C$ do
        Compute probability: $\mathbb{P}(S_iC_j|C_j, \boldsymbol{\theta}^{(r+1)}, \mathbf{D})$ according to (3)
    end for
    for $S_i$ in $S$ do
        Estimate parameters: $\mathbf{A}_{pq}^{i,(r+1)}$, $\mathbf{P}_{pq}^{i,(r+1)}$ and $\mathcal{Z}_j^{(r+1)}$ according to (5)
    end for
end while
Update EM likelihood $\mathcal{L}$ according to (2)
for $C' \subset C, |C'| = V$ do
    Compute expected likelihood estimation $\mathcal{L}_{C'}$ according to (12)
    if $\mathcal{L}_{C'} > \mathcal{L}$ then
        $\mathcal{L} = \mathcal{L}_{C'}$
        $\bar{C} = C'$
    end if
end for
Label $\bar{C}$ by human participants
while $\bar{\boldsymbol{\theta}}^{(r)}$ does not converge do
    for $C_j$ in $C$ do
        Update parameters $\mathbf{A}_{pq}^{i,(r+1)}$, $\mathbf{P}_{pq}^{i,(r+1)}$ and $\mathcal{Z}_j^{(r+1)}$ according to (13)(14)
    end for
end while
Output: $\mathbf{A}_{pq}^{i}$, $\mathbf{P}_{pq}^{i,(r+1)}$ and $\mathcal{Z}_j^{k}$
END
```

result of assertion j. The expected complete data log likelihood conditioning on verification result $\mathcal{V}$ can be written as:

$$\begin{aligned} Q_{\mathcal{V}} &= \mathbb{E}_{\boldsymbol{\theta}_{\mathcal{V}}}[\log \mathbb{P}(SC|\boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V})] \\ &= \sum_{C_j \in C} \sum_{q=0}^{v} \mathbb{P}(C_j = q|\boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) \log \mathbb{P}(SC_j|C_j = q, \boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) \\ &= \sum_{C_j \in \bar{C}} \sum_{q=0}^{v} \mathbb{P}(C_j = q|\boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) \log \mathbb{P}(SC_j|C_j = q, \boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) \\ &+ \sum_{C_j \in \tilde{C}} \sum_{q=0}^{v} \mathbb{P}(C_j = q|\boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) \log \mathbb{P}(SC_j|C_j = q, \boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) \end{aligned} \tag{7}$$

in which first term is the log likelihood of labeled assertions, the probability of assertion j of value q is

$$\mathbb{P}(C_j = q|\boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) = \begin{cases} 1 & q = \mathcal{V}_j \\ 0 & q \neq \mathcal{V}_j \end{cases} \tag{8}$$

The second term is the log likelihood of unlabeled assertions, it is derived according to EM algorithm described in section 3.2. The semi-supervised expectation maximization likelihood is then formulated as:

$$\begin{aligned} Q_{\mathcal{V}} = &\sum_{C_j \in \tilde{C}} \sum_{q=0}^{v} \mathbb{P}(C_j = q|\boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) \log \mathbb{P}(SC_j|C_j = q, \boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) \\ &+ \sum_{C_j \in \bar{C}} \mathbb{P}(C_j = \mathcal{V}_j|\boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) \log \mathbb{P}(SC_j|C_j = \mathcal{V}_j, \boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) \end{aligned} \tag{9}$$

Our goal is to estimate the set of labeled assertions that achieve best semi-supervision results. To maximize the above likelihood over all possible combinations of labeled assertions, we adopt linear programming to estimate semi-supervised EM. The main observation is: Given the bipartite $\mathcal{SC}$ graph, as we verify an assertion (change from estimated truth value to ground truth), the reliability estimates of sources that make claims on the verified assertion are explicitly influenced: those sources gain higher reliability estimates since they correctly claim the verified assertion, while other sources slightly lose reliability estimates depending on their speak rates. As source reliability estimates change, other assertions are indirectly influenced. It can in turn affects more sources, however the changes are minimal. Thus in verification selection, we assume the estimated truth of non-verified assertions remain unchanged, which enables linear programming to simplify the above EM likelihood. The simplified log likelihood conditioning on $\mathcal{V}$ is:

$$\begin{aligned} Q_{\mathcal{V}} = &\sum_{C_j \in \tilde{C}} \sum_{q=0}^{v} \mathbb{P}(C_j = q) \log \mathbb{P}(SC_j|C_j = q, \boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) \\ &+ \sum_{C_j \in \bar{C}} \mathbb{P}(C_j = \mathcal{V}_j|\boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) \log \mathbb{P}(SC_j|C_j = \mathcal{V}_j, \boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) \\ = &\sum_{C_j \in \tilde{C}} \sum_{S_i \in \mathcal{S}C_j} \sum_{q=0}^{v} \mathbb{P}(C_j = q) \log \mathbb{P}(S_iC_j|C_j = q, \boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) \\ &+ \sum_{C_j \in \bar{C}} \sum_{S_i \in \mathcal{S}_j} \mathbb{P}(C_j = \mathcal{V}_j|\boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) \log \mathbb{P}(S_iC_j|C_j = \mathcal{V}_j, \boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) \end{aligned} \tag{10}$$

where $\mathbb{P}(C_j = q)$ is estimated in previous section 3.2, and

$$\mathbb{P}(S_iC_j|C_j, \boldsymbol{\theta}_{\mathcal{V}}, \mathbf{D}, \mathcal{V}) = \begin{cases} \mathbf{A}^i_{pq|\mathcal{V}} & C_j = q, S_iC_j = p, \mathbf{D}_{ij} = 0 \\ \mathbf{P}^i_{pq|\mathcal{V}} & C_j = q, S_iC_j = p, \mathbf{D}_{ij} = 1 \end{cases} \tag{11}$$

$\mathbf{P}^i_{pq|\mathcal{V}}$ describes dependent behaviours between two sources (follow or retweet) given verification result $\mathcal{V}$. $\mathbf{A}^i_{pq|\mathcal{V}}$ is the probability of making independent assertions providing the verification result is $\mathcal{V}$. Note that, the verification result $\mathcal{V}$ is unknown in this section. We hence need to compute the expected likelihood by averaging over all possible verification results. The probability of $\mathcal{V}$ is derived from truth estimation $\mathcal{Z}$ in previous section 3.2, such that $\mathbb{P}(\mathcal{V}) = \prod_{v_j \in \mathcal{V}} z_{j,v_j}$. The expected likelihood is then formulated as:

$$Q = \sum_{\mathcal{V}} \mathbb{P}(\mathcal{V}) Q_{\mathcal{V}} \tag{12}$$

The simplified EM likelihood can be maximized as a linear programming, with constraint $\sum_q \mathbf{A}^i_{pq|\mathcal{V}} = 1$, $\sum_q \mathbf{P}^i_{pq|\mathcal{V}} = 1$ and $|\bar{C}| = V$, where V is the number of assertions to be verified.

3.4 Re-estimating Truth Value and Source Reliability with Labeled Assertions

The optimal set of assertions is then verified by human participants. After obtaining the verification results, we re-compute truth estimates from semi-supervised EM likelihood formulated in (9). Using E-step and M-step described in section 3.2, we can derive iterative updates:

$$\mathcal{Z}_{j,p} = \begin{cases} \frac{B_p}{\sum_{q=1}^{v} B_q} & j \notin \bar{C} \\ 1 & j \in \bar{C}, q = v_j \\ 0 & j \in \bar{C}, q \neq v_j \end{cases}$$

$$\mathbf{A}^i_{pq} = \frac{\sum_{S_iC_j=p,\mathbf{D}_{ij}=0} z_{j,q}}{\sum_{S_iC_j,\mathbf{D}_{ij}=0} z_{j,q}}$$

$$\mathbf{P}^i_{pq} = \frac{\sum_{S_iC_j=p,\mathbf{D}_{ij}=1} z_{j,q}}{\sum_{S_iC_j,\mathbf{D}_{ij}=1} z_{j,q}} \tag{13}$$

where

$$B_p = \prod_{q=1}^{v} \prod_{S_iC_j=q} \mathbb{P}(S_iC_j|C_j = p) \tag{14}$$

Algorithm 2 Greedy Approximation

```
Input: Source-Claim matrix $SC$ and Number of assertions to label $\mathcal{V}$
Initialize $\bar{C} = \emptyset$,$\tilde{C} = C$, $\mathcal{L}$ and $d$
Compute EM estimation according to (5)(6)
while $|\bar{C}| < V$ do
    for $C_j \in \tilde{C}$ do
        Compute expected likelihood $\mathcal{L}_j$ according to (16)
        if $\mathcal{L}_j > \mathcal{L}$ then
            $\mathcal{L} = \mathcal{L}_j$
            Set $j$ to be the verification index $d$
        end if
    end for
    Add $C_j$ to $\bar{C}$
end while
Re-Compute EM estimations according to (13)(14)
END
```

3.5 A Greedy Approach

The above verification algorithm takes V^d set of parameters to cover all d possible verification results of V assertions. Although each verification estimation takes minimal computation cost, the number of combinations can grow exponentially as the dataset. In this section, we describe a greedy approach to significantly reduce number of parameters on such large dataset. Start with $\bar{C} = \emptyset$, the approach adds one assertion to $\bar{C}$ via greedy approach (select the assertion that maximizes expected likelihood). We run the above iteration V times, getting one verified claim in each iteration. We may also redo expectation maximization to update source reliability and truth after certain number of verifications.

In this section, we analyze the greedy approach to demonstrate it is a good estimation. We study the effect of verifying one assertion. Denote the verified assertion as C_{ve}, estimated correctness before verification is z_{ve}. $\bar{C}$ is the set of labeled assertions (not including C_{ve}), $\tilde{C}$ is the set of unlabeled assertions. The likelihood before verifying C_{ve} is:

$$\begin{aligned} Q = & \sum_{C_j \in \bar{C}} \sum_{q=0}^{v} \mathbb{P}(C_j = q|\boldsymbol{\theta}_{ve}, \mathbf{D}) \log \mathbb{P}(SC_j|C_j = q, \boldsymbol{\theta}_{ve}, \mathbf{D}) \\ & + \sum_{C_j \in \tilde{C}} \sum_{q=0}^{v} \mathbb{P}(C_j = q) \log \mathbb{P}(SC_j|C_j = q, \boldsymbol{\theta}_{ve}, \mathbf{D}) \\ & = \sum_{C_j \in \bar{C}} \sum_{q=0}^{v} \mathbb{P}(C_j = q|\boldsymbol{\theta}_{ve}, \mathbf{D}) \log \mathbb{P}(SC_j|C_j = q, \boldsymbol{\theta}_{ve}, \mathbf{D}) \\ & + \sum_{C_j \in \tilde{C} - C_{ve}} \sum_{q=0}^{v} \mathbb{P}(C_j = q) \log \mathbb{P}(SC_j|C_j = q, \boldsymbol{\theta}_{ve}, \mathbf{D}) \\ & + \sum_{q=0}^{v} \mathbb{P}(C_{ve} = q) \log \mathbb{P}(SC_j|C_{ve} = q, \boldsymbol{\theta}_{ve}, \mathbf{D}) \end{aligned} \tag{15}$$

On verifying C_{ve}, with probability $\mathbb{P}(C_{ve} = q_{ve})$, the result is $C_{ve} = q_{ve}$. The expected likelihood after verifying C_{ve} is:

$$\begin{aligned} Q' = & \sum_{q_{ve}=0}^{v} \mathbb{P}(C_{ve} = q_{ve}) \times \\ & \{ \sum_{C_j \in \bar{C}} \sum_{q=0}^{v} \mathbb{P}(C_j = q|\boldsymbol{\theta}_{q_{ve}}, C_{ve}, \mathbf{D}) \log \mathbb{P}(SC_j|C_j = q, C_{ve}, \boldsymbol{\theta}_{q_{ve}}, \mathbf{D}) \\ & + \sum_{C_j \in \tilde{C} - C_{ve}} \sum_{q=0}^{v} \mathbb{P}(C_j = q) \log \mathbb{P}(SC_j|C_j = q, C_{ve}, \boldsymbol{\theta}_{q_{ve}}, \mathbf{D}) \\ & + \log \mathbb{P}(SC_j|C_{ve}, \boldsymbol{\theta}_{q_{ve}}, \mathbf{D}) \} \end{aligned} \tag{16}$$

Since we assume the estimated truth of non-verified assertions remain unchanged, thus for any assertion $j \in \tilde{C}, j \neq ve$, we have:

$$\mathbb{P}(C_j = q|\boldsymbol{\theta}_{C_{ve},q_{ve}}, \mathbf{D}) = \mathbb{P}(C_j = q) \tag{17}$$

The effect of verifying an assertion is two-fold: Change on estimated correctness of verified assertion and change on estimated reliability scores of related sources.

Change on estimated correctness of verified assertion

First, we study the effect on estimated correctness of verified assertion C_{ve}. Estimates of C_{ve} changes from z_{ve} before verification, to 1 with probability z_{ve} and 0 with probability $1 - z_{ve}$. This part of likelihood is represented as the last item of (15) and (16). We can see the overall expected likelihood does not change, since the probability of verified results follows the correctness estimation of C_{ve}. Thus, multiple verified assertions are independent in terms of influence of estimated correctness.

Change on estimated reliability scores of related sources
Verifying an assertion changes source reliability scores, which implicitly affects estimated truth value of unlabeled assertions. Original claim probability matrix of sources are estimated as:

$$\mathbf{A}^i_{pq} = \frac{\sum_{S_i C_j = p} z_{j,q}}{\sum_{S_i C_j} z_{j,q}} \tag{18}$$

After verification, estimated reliability matrices become:

$$\mathbf{A}^i_{pq} = \begin{cases} \frac{\sum_{j \in C - C_{ve}, S_i C_j = p} z_{j,q} + 1}{\sum_{j \in C - C_{ve}, S_i C_j} z_{j,q} + 1} & \mathbb{P} = z_{ve} \\ \frac{\sum_{j \in C - C_{ve}, S_i C_j = p} z_{j,q}}{\sum_{j \in C - C_{ve}, S_i C_j} z_{j,q}} & \mathbb{P} = 1 - z_{ve} \end{cases}$$

Same applies to dependent claim probability matrix. Since verification affects both nominator and denominator in reliability estimates, it does change the expected likelihood. Multiple verifications interact dependently on source reliability scores. Providing we are only verifying a small set of assertions, we can make a further approximation that the changes on source reliability are positive (With high probability, a verification overall improves source reliability estimation). With this assumption, the problem becomes submodular and monotone, and thus greedy algorithm gives a $(1-1/e)$-approximation solution to our problem. However, in sparse networks, the above assumption does not hold and will cause slight performance drop.

4 EVALUATION USING SYNTHETIC DATA

We evaluate our algorithm using both real and synthetic data sets. This section describes evaluation using synthetic data. Evaluation using real-world data is presented in the next section.

Our use of synthetic data admittedly misses realistic qualities of target applications. The advantage, however, lies in exercising full control over ground truth, nature of assertions, and nature of sources, hence allowing evaluation under a broad set of conditions. By changing parameters of synthetic data generation, we are able to comment on general trends and on robustness of our results across a wide range of conditions. This broader picture complements our more localized (if accurate) understanding derived from the point cases represented by the specific collected data sets (in the next section).

We develop a synthetic data generator that creates n sources $\{S_1, S_2, \cdots, S_n\}$, of which some fraction are not reliable. Each source makes a subset of assertion selected from some larger set maintained by the generator. The generator works as follows:

- First, it generates m assertions $\{C_1, C_2, \cdots, C_m\}$ to choose from, and chooses source reliability and speak rates for each of the n sources. Reliability matrices and speak rates are generated as Gaussian distributions of preset parameters. A control parameter P_{true} determines the percentage of true assertions in the overall set. Hence, turning that parameter down would increase the fraction of "rumors" compared to true statements among the data reported.
- Second, the generator generates claims in accordance with the chosen source parameters and populates the set $\mathcal{SC}$ for the period of simulation. It loops over all sources. In each round, the speak rate of a source, P_{speak}, decides if the source will make a claim. If so, another probability, P_{depend}, decides whether or not this claim will be a repeat of a previous one (e.g., retweet). A repeat is generated in accordance with the dependency graph by repeating some claim of an ancestor in the graph. If P_{depend} decides that the claim is an original claim, then either a true or a false assertion is chosen according to reliability parameters of this source.
- The generated claims using the above method are given to our set of algorithms to analyze.

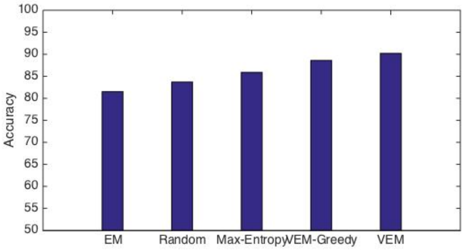

Figure 1: Simulation accuracy

In the simulation, reliable sources have expected 70% chance of making correct claims, while unreliable sources have expected 40%. We evaluate five different methods:

- EM: The pure *EM* uses expectation maximization but without any verification. It jointly estimates source reliability and assertion truth values from source claim matrix *SC* and dependency graph *G*. The original algorithm was published in [14].
- Random: It augments *EM* by randomly selecting the assertions to verify.
- Max-Entropy: The *Max-Entropy* algorithm picks the assertions that have the closest probability of true and false values (i.e., the most uncertainty in estimated value).
- VEM: *VEM* is the active-learning algorithm described in section 3.2 - 3.4.
- VEM-Greedy: It is the Greedy approximation of *VEM*, described in 3.5.

We conduct 50 independent experiments for each data point. For data generation, unless mentioned otherwise, we use number of sources $n = 50$, speak rate $P_{speak} = 0.2$, expected reliability of good sources = 0.7 and expected reliability of bad sources = 0.4. A total of $m = 250$ claims are generated in each simulation. We limit the assertions to verify to 5%. In general, we expect our algorithm to work better when sources are more predictable (e.g., good sources are nearly 100% accurate, and bad sources are always inaccurate). The above settings (of 70% and 40% reliability) were meant to increase entropy in order to stress test the algorithm.

4.1 Simulation Accuracy

We compare the simulation results of the five algorithms to ground truth. From Fig. 1, we see that our *VEM* algorithm outperforms all other algorithms. The *Random* selection of assertions to verify can occasionally correct a claim but is inferior to others. The *Max-Entropy* algorithm selects more uncertain assertions, but it is not the best choice, since it does not take into account the global

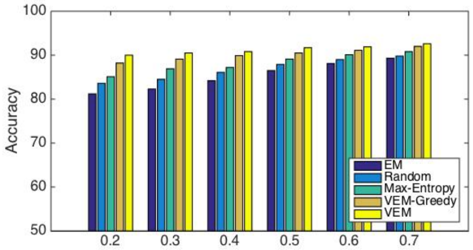

Figure 2: Vary Claim Prob.

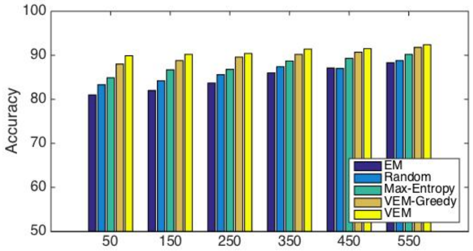

Figure 3: Vary # of Sources

interaction between assertions. Our algorithm achieves better accuracy than the above two methods. The reason is two-fold. First, *VEM* jointly considers an assertion's value uncertainty and its interactions with other assertions. For example, if only one unknown source made this assertion then knowing its true value has a more limited effect (on understanding source reliability) than if several unknown sources made this assertion (in which case the verification will yield information pertinent to several sources at once). Second, *VEM* selects the best set of assertions to verify by iteratively estimating verification outcome. *VEM-Greedy* provides a low cost approximation to our *VEM* algorithm to reduce computational complexity. It achieves comparable accuracy but is much lighter. Next, we study the effect of varying data set generation parameters. In each experiment, we vary one parameter while keeping others unchanged. The simulator generates a new dataset of each set of parameters. First, we vary the parameter P_{speak}, the probability of a source making claim, from 0.2 to 0.7 with increments of 0.1. This has the effect of modulating the density of the source claim matrix. In Fig. 2, our *VEM* achieves the best accuracy for every speak rate setting. *VEM-Greedy* has comparable performance to *VEM* and still beats the other three algorithms. Note that, although *VEM* has better accuracy than *VEM-Greedy*, it iterates through all possible combinations of assertions, which is computationally expensive in a large dataset. Another observation is that, at low speak rates, our algorithm significantly outperforms *EM*, *Random* and *Max-Entropy*. The reason is, in a sparse source claim matrix, *EM* produces more uncertain results: more truth estimations fall close to 0.5 than 0 or 1. *VEM* provides better improvements over baselines on such sparse graphs. In the second experiment, we change the number of sources n from 50 to 550 in increments of 100. In Fig. 3, our algorithm outperforms the others over the entire range. Adding sources has a similar effect on accuracy as increasing speak rate. In our default setting, with $P_{speak} = 0.2$, the graph is sparse, which impairs the performance of the *EM* algorithm. When we increase the number of sources, the number of edges connecting to each assertion is increased. This leads to an improvement in performance. Again, our algorithm outperforms others for every setting of parameters. In third experiment, we vary the number of assertions from 250 to 750 in increments of 100. The speak rate P_{speak} and the number of sources are unchanged. In Fig. 4, the baseline *EM*, *Random* and *Max-Entropy* have similar trends as in the previous two simulations. Our *VEM* and *VEM-Greedy* improve slightly more when increasing the number of assertions. The reason is: since we always verify 5% of the total number of assertions, having more assertions allows us to verify more as well, deriving better value due to their judicious selection of claims to verify. The immediate implication is that our algorithm fits well with larger data sets.

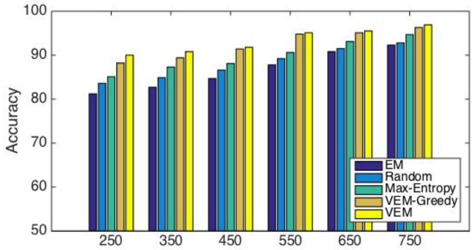

Figure 4: Vary # of Assertions

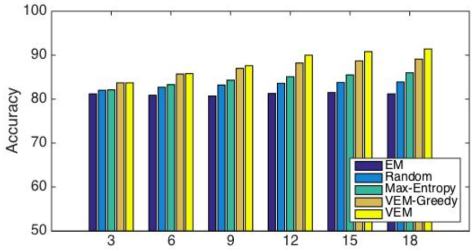

Figure 5: Vary # of verified assertions

In the next simulation, we study the effect of varying number of verified assertions. We change the this number from from 3 to 18 with increment of 3, keeping the total at 250 (i.e., roughly 1.2% to 7.5%). In Fig. 5, our algorithm is shown to outperform the others in all 6 scenarios. There are two observations:

- We observe an increasing performance gap between *VEM* (*VEM-Greedy*) and other algorithms as we increase number of verified assertions. *Random* and *Max-Entropy* use a passive learning approach, which has no control over labeled data. Thus, they are susceptible to selecting less advantageous assertions. Our *VEM* algorithm determines the best assertions to verify offering a better advantage as the number of verified assertions increases.
- The improvements in total accuracy achieved slow down after 12 assertions. This demonstrates 5% is a good fraction for pivot assertions.

4.2 Compare Computation Time

Here we compare the computation overheads of different algorithms. We increase the size of the source claim matrix by increasing the number of sources and assertions as indicated on the horizontal axis. We then average the time needed to process the data sets over

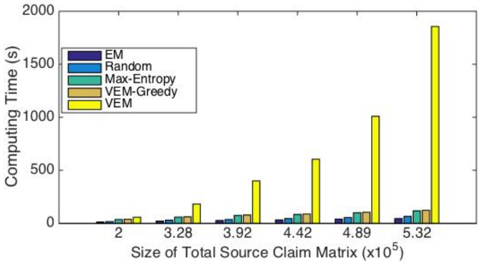

Figure 6: Computation Time Comparison

20 independent runs. The result is shown in Fig. 6. *Random*, *Max-Entropy* and *VEM-Greedy* have slight overheads over baseline *EM*. Our *VEM*, however, grows exponentially with the number of claims. The simulation indicates that *VEM* is good for smaller datasets but does not scale well to larger ones. However, *VEM-Greedy*, its greedy approximation, has a roughly similar overhead as *Random* and *Max-Entropy*, yet still achieves better results than other methods.

Results in this section confirm two main observations. First, the accuracy improvement attributed to the new algorithm holds true across a broad set of conditions on sources, assertions, percentage of verified assertions, and overall sparsity of the Source-Claim matrix. Overall, better advantages are seen in larger experiments. Second, the quality of the results is fairly robust to changes in synthetic data generation. Hence, when discussing specific data sets in the next section, we have reason to believe that the observed performance advantages are not coincidental to only those specific data sets and configurations. With that, we turn to results obtained when experimenting with real data.

5 EMPIRICAL EVALUATION

In this section, we evaluate our algorithm on four empirical data sets collected on Twitter in 2018 and 2015. The data were collected using Twitter API. Each data set was collected by specifying event keywords or a geographic location. The data collection tool would then collect tweets that contain the indicated keywords or originate from the indicated location. Four datasets were created, labeled: (1) Hurricane, (2) Facebook, (3) Kirkuk, (4) LA Marathon, representing the events of: (1) Hurricane Florence in 2018, (2) Facebook under federal investigation, (3) Battle with ISIS in Kirkuk city, and (4) A marathon in LA:

- Hurricane Florence: In September 2018, Hurricane Florence hit the US East Coast, including North Carolina, South Carolina and Maryland. The disaster caused 53 deaths and over 38 *billion* USD dollars in damage over its course.
- Facebook Federal Investigation: In 2018, Facebook faced a federal investigation into its sharing of user data with the political consulting firm Cambridge Analytica. Some 87 million facebook users' data were claimed to be improperly shared.
- Kirkuk: On March 10th 2015, Kurdish forces in Northern Iraq attacked the self proclaimed “Islamic State" (ISIS) outpost, west of the city of Kirkuk. Battles raged and a lot of commentary followed on social media about state of affairs on the ground.
- LA Marathon: This data set collects tweets about the Los Angeles Marathon on March 15th, 2015. The runners started from Dodger Stadium and ran a course to the coastline in Santa Monica, passing through downtown Los Angeles, Hollywood, the Sunset Strip, Santa Monica Blvd, and Ocean Avenue, among other landmarks. Much Twitter activity followed the event as it unfolded.

Table 2: Information Summary of Twitter Dataset

	#Assertions	#Sources	#Claims
Hurricane Florence	9052	7645	9378
Facebook	10238	8236	11395
Kirkuk	6172	4816	6188
LA Marathon	7072	5174	7148

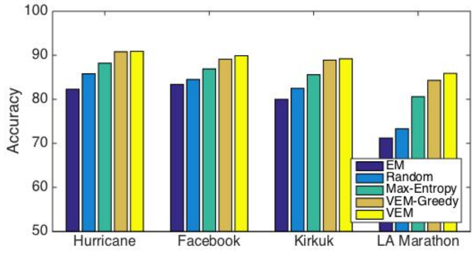

Figure 7: Empirical evaluation

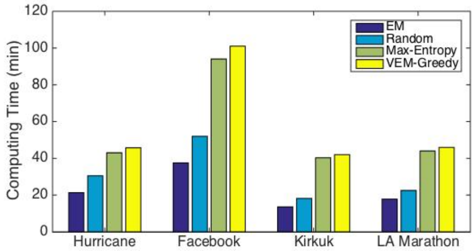

Figure 8: Computation Time

Our intent was to use data sets that differ significantly in time (e.g., today versus three years ago), and differ significantly in topic (e.g., social event, political event, or natural disaster). To evaluate our algorithm, the most popular 1000 assertions were selected (i.e., those supported by the highest number of claims from sources). Each algorithm then computes the probability of each of these 1000 assertions to be true. To avoid variations in the number of assertions returned, the highest-probability 200 assertions (ranked true) by each algorithm are computed. We then manually grade the correctness of those tweets by human graders. To prevent bias, the graders have no information on which algorithm generated which output. Each tweet is marked as "True", "False", or "Unconfirmed" by the grader. An "Uncomfirmed" tweet is either non-verifiable or is an opinion (a subjective assessment, such as 'Facebook is user-friendly'). We then compute accuracy as the percentage of assertions labeled "True" by the grader in the top 200 returned by each algorithm: $\#True/(\#True + \#False + \#Unconfirmed)$. Note that our accuracy analysis presents a pessimistic performance evaluation, since we count unconfirmed assertions as "False".

The results of this evaluation are shown in Fig. 7. Consistently with simulations, our algorithm outperforms all baselines. The reason is: *VEM* iterates through every combination of assertions to estimate the semi-supervised EM likelihood, which is theoretically more accurate than other heuristic methods. *Random* selection and *Max-Entropy* fail to select the best assertions to verify.

Our algorithm improves accuracy by 8.6%, 6.5%, 9.2%, 12.7% respectively over baseline *EM* when only 5% of the total assertions are verified. Said differently, it reduces error roughly in half (from the neighborhood of 20% to the neighborhood of 10%). Thus, it does significantly improve truth finding quality in social sensing systems. In addition, the lighter version, *VEM-Greedy*, has comparable performance in all four datasets.

Lets take a deeper look at those algorithms. *EM* is the baseline approach. It allows us to understand the benefits of active learning. Prior work shows that *EM* outperforms traditional iterative algorithms (i.e., Sum, Average Log, Truth Finder, in social sensing system) [18]. The main sources of inaccuracy in the unsupervised *EM* are: (1) It requires relatively dense data to have better accuracy. (2) Our evaluation counts subjective opinions as False. Subjective assessments made by reliable sources get high credibility scores, which reduces accuracy in our evaluation.

Random reflects performance of semi-supervision if the set of labeled assertions was not controlled (i.e., when labeled data are arbitrarily chosen). It allows us to assess the performance improvement due to a good choice of assertions to verify. Similarly, the original *Max-Entropy* shows that a good choice depends on more than just the uncertainty in the value of the assertion. *Max-Entropy* improves accuracy over *Random* by verifying the most uncertain assertions. Our algorithm does better because it succeeds in finding assertions that are connected in the source claim graph in such a way that their verification potentially benefits other estimated values. Hence, by verifying a small set of assertions, we also improve the estimation of source reliability, which influences the estimation of other assertions. *VEM* uses linear programming to find the best assertions to verify. It does incur a high computational cost. Thus, the lighter version of *VEM*, *VEM-Greedy* is derived. Empirical results show that it still outperforms *Random* and *Max-Entropy*.

Finally, Fig. 8 gives a comparasion of computation time on real datasets. *Random* has similar execution time as baseline *EM* since the algorithm only does one round of EM estimation. *Max-Entropy* and *VEM-Greedy* does two rounds of EM because those two algorithms select assertions to verify based on EM estimates then re-estimate with labeled assertions. *VEM* takes 2276, 3365, 350 and 564 minutes to execute due to its high combinatorial computation cost. Note that, *VEM* does not need to iterate over every combinations in real datasets, since some subsets of assertions can be eliminated by heuristics.

The evaluation shows that across a diverse selection of Twitter data sets from different times and topics, our algorithm, *VEM*, improves accuracy, compared to the baselines in a manner largely consistent with simulation results. Moreover, *VEM-Greedy* offers nearly the same performance at a much lower overhead.

6 DISCUSSION AND FUTURE WORK

The work presented in this paper can be extended in several ways.

Errors and imperfections: The accuracy of our approach is contingent on the accuracy of several underlying components. For example, we group claims by cosine similarity, but semantically-aware clustering methods may yield better results. We assume an "oracle" that allows the selected claims to be reliably verified, but the oracle itself might be wrong. Not all assertions might be verifiable by an oracle. In some scenarios, only a subset can be verified. Our model relaxation that enables the greedy algorithm to perform well does not hold when networks are sparse. This deficiency partially explains the performance gap between the expensive form of inference and the greedy form. A more thorough analysis is needed in order to assess the sensitivity of our approach to these and other sources of error.

One-hot constraints: Our approach as discussed evaluates the truth values of individual assertions independently. In reality, there may be "one-hot" constraints on the value that is true. For example, different assertions might describe a suspect in a street shooting differently, but if the descriptions are conflicting, only one description is true. Our algorithm can be easily extended to accommodate such constraints in scenarios where it is possible to identify claims that are mutually conflicting. This might be easier to do for more structured data, such as crowdsensing data, where users report multiple-choice observations via an app.

Scalability: One important feature of crowdsourcing systems is scalability. Our algorithm runs under a batch setting, which means all data are considered at the same time in the estimation. In practice, this is not always a concern. Often one desires to verify information on specific claims or topics, which allows pre-filtering of larger datasets to the issue of concern. However, should one decide to run the algorithm on larger data, one possibility is to use an incremental form. Specifically, previous work presented online recursive truth estimators based on an EM framework that allow propagating information from one window to the next as priors [18] to be undated incrementally in the current window. This allows the algorithm to work on one slice of data at a time, making it more scalable.

Two changes must be made to support such an online extension. First, time should be slotted into intervals. The EM framework would only be applied to claims collected in the current time-slot. Posterior beliefs are then computed from EM estimates of current and previous time-slots, summarizing source reliability and claim correctness probabilities from previous history. Posterior beliefs then become prior beliefs in the next time-slot and are included in online EM estimation [18]. Second, since user queries can be made at any time-slot, an interesting challenge is whether the labeling overhead can be distributed more judiciously over time. For example, if no significant uncertainty exists, maybe spare some verification budget for the next window.

Collusion: As with many fact-finding and reputation systems, one challenge is malicious attacks. A group of collaborators or multiple replicas generated by a single user can attempt to fool the system to boost correctness of false assertions. This can be done by gradually gaining trust from providing true observations, then coordinating on delivering the same false message(s). While it is virtually impossible to detect a single such manipulation, repeated manipulation over time is detectable. Previous literature proposed mechanisms to detect such collaborations based on several observations on social networks. For example, collaborating groups tend to coincidentally make observations on the same set assertions repeatedly [10] and tend to have more friend/follow links among themselves than external links to honest users [8]. Our assertion selection can combine those features to better assess source reliability in the presence of malicious attacks.

Verification cost: We implicitly assume that every verification attempt has the same cost, leading to a formulation where only the total fraction of verified assertions is bounded. Our algorithm can be further extended to the settings where verification of different assertions has a different cost. The problem becomes to select an optimal set of assertions while keeping cost below a preset value. Extending *VEM* algorithm is straight-forward, since it iterates through every possible set of assertions and estimates their verification effect. *VEM-Greedy*, however, has to consider verification cost in some appropriate manner since it selects assertions greedily one at a time. A simple solution is to select the assertion which has highest approximate improvement per unit cost.

Natural language processing: Our approach minimizes the use of natural language processing and does not interpret the content of tweets, which makes it easily applicable to any language, dialect, or topic with no prior language training. If tools for natural language processing were to be used, they would improve our truth estimator in several ways. For example, the same observation can be expressed in multiple ways or languages, which we shall fail to recognize as the same assertion. Some natural language processing could mitigate this problem.

Applicability and real world scenarios: We are currently working on integrating our algorithm into our Apollo social sensing toolkit (http://apollo2.cs.illinois.edu). Apollo extracts tweets hourly using the Twitter API and processes them to discover real-world events related to a given broad topic, such as disasters, world conflicts, local traffic, or other categories, supplied by the user as query terms. It further filters descriptions of these events, retaining only what it believes to be the most trustworthy claims (tweets). Using the approach described in this paper, the tool can be extended by selecting a very small subset of tweets and asking the user to assess their veracity. In return, our algorithm will improve the quality of assessment and filtering of the majority of claims. This capability can be applied in many contexts. For example, in post-disaster scenarios, rumors often proliferate about the extent of damage, availability of help, and instructions for survivors. Apollo can help mitigate the confusion. More recently, actors on social media were blamed for misinformation campaigns, for purposes ranging from intimidation to political manipulation. Apollo may offer a countermeasure that reduces information contamination. Finally, in the absence of disasters and conflicts, the tool can be used simply as an online (near) real-time news source, distilling myriads of evolving events, on a topic of user choice, into a much smaller amount of higher-quality newsworthy headlines. In the authors' own lab, Apollo is being used by graduate students to catch up on the most salient events in their home countries, especially those that do not receive much coverage in the local news. This is an anecdotal proof of value of the tool as a new personalizable medium to keep up with evolving real-time events worldwide.

7 RELATED WORK

This paper builds on advances in social sensing, which evolved from participatory sensing, initially introduced in [2]. Recent applications include BigRoad[6], RST[7], and Nutrilyzer[13].

Reliability and data quality are among the key problems of social sensing. To eliminate conflicts among multi-source data, the topic of truth-discovery has been studied extensively in machine learning/data mining [5, 19, 20] and social network literature [1, 15–17].

Early truth finders were based on the concepts of hubs and authority, in which the source reliability and claim trustworthiness updates each other in an iterative fashion. Yin et al. utilized interdependencies between source trustworthiness and fact confidence, introducing TruthFinder as an unsupervised fact finder. Later work extended TruthFinder by introducing heuristics into the analysis, including: small set of pre-labeled truth [20], incorporating prior knowledge [11], similarity between claim values [4] and extra parameters to source reliability [12]. Wang et al. [15] first formulated fact finding into a maximum likelihood estimation problem and jointly estimated source reliability and claim correctness using an expectation maximization approach. Subsequent work [14, 16] extended the original EM by taking into account dependencies.

The above work assumed a batch setting in which all data have to be available at once. However, the amount of data can grow exponentially over time, and the gap between staring and finishing of data collection can be large, which leads to high computation cost and slow response to queries. Yao et al. proposed a recursive estimator for streaming social media data[17]. The online recursive estimator adapted a batch EM framework by passing posterior belief across time windows. We are the first extension of the maximum-likelihood estimation approach for social media truth discovery to the case of active learning that investigates the selection of data to label for maximum improvement in results.

8 CONCLUSIONS

In this paper, we presented an active-learning based truth estimator that discovers truth values of observations made in social networks. Our algorithm achieves significantly higher accuracy compared with existing methods. We evaluated our approach in simulation as well as on real data collected from Twitter. Our algorithms significantly outperformed existing work. A light-weight version of the original approach was developed that offered the best compromise between quality of results and computation efforts. The paper is a first example of applying an active learning approach to improve the performane of truth discovery on social media.

ACKNOWLEDGMENT

Research reported in this paper was sponsored in part by the U.S. Army Research Laboratory under Cooperative Agreements W911NF-17-2-0196 and W911NF-09-2-0053, DARPA contract W911NF-17-C-0099, and NSF grants CNS 13-29886 and CNS 16-18627. 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.

REFERENCES

[1] Md Tanvir Al Amin, Charu Aggarwal, Shuochao Yao, Tarek Abdelzaher, and Lance Kaplan. 2017. *Unveiling polarization in social networks: A matrix factorization approach.* Technical Report. IEEE.

[2] Jeffrey A Burke, Deborah Estrin, Mark Hansen, Andrew Parker, Nithya Ramanathan, Sasank Reddy, and Mani B Srivastava. 2006. Participatory sensing. *Center for Embedded Network Sensing* (2006).

[3] Hang Cui, Tarek Abdelzaher, and Lance Kaplan. 2018. Recursive Truth Estimation of Time-Varying Sensing Data from Online Open Sources. In *International Conference on Distributed Computing in Sensor Systems (DCOSS)*. New York, NY.

[4] Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava. 2009. Integrating conflicting data: the role of source dependence. *Proceedings of the VLDB Endowment* 2, 1 (2009), 550–561.

[5] Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava. 2009. Truth discovery and copying detection in a dynamic world. *Proceedings of the VLDB Endowment* 2, 1 (2009), 562–573.

[6] Luyang Liu, Hongyu Li, Jian Liu, Cagdas Karatas, Yan Wang, Marco Gruteser, Yingying Chen, and Richard P Martin. 2017. Bigroad: Scaling road data acquisition for dependable self-driving. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 371–384.

[7] Chuishi Meng, Houping Xiao, Lu Su, and Yun Cheng. 2016. Tackling the Redundancy and Sparsity in Crowd Sensing Applications.. In *SenSys*. 150–163.

[8] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. 2007. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM, 29–42.

[9] Praneeth Netrapalli and Sujay Sanghavi. 2012. Learning the Graph of Epidemic Cascades. *SIGMETRICS Perform. Eval. Rev.* 40, 1 (June 2012), 211–222. https://doi.org/10.1145/2318857.2254783

[10] Praneeth Netrapalli and Sujay Sanghavi. 2012. Learning the graph of epidemic cascades. In *ACM SIGMETRICS Performance Evaluation Review*, Vol. 40. ACM, 211–222.

[11] Jeff Pasternack and Dan Roth. 2010. Knowing what to believe (when you already know something). In *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 877–885.

[12] Jeff Pasternack and Dan Roth. 2013. Latent credibility analysis. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 1009–1020.

[13] Tauhidur Rahman, Alexander Travis Adams, Perry Schein, Aadhar Jain, David Erickson, and Tanzeem Choudhury. 2016. Nutrilyzer: A Mobile System for Characterizing Liquid Food with Photoacoustic Effect.. In *SenSys*. 123–136.

[14] Dong Wang, Md Tanvir Amin, Shen Li, Tarek Abdelzaher, Lance Kaplan, Siyu Gu, Chenji Pan, Hengchang Liu, Charu C Aggarwal, Raghu Ganti, et al. 2014. Using humans as sensors: an estimation-theoretic perspective. In *Information Processing in Sensor Networks, IPSN-14 Proceedings of the 13th International Symposium on*. IEEE, 35–46.

[15] Dong Wang, Lance Kaplan, Hieu Le, and Tarek Abdelzaher. 2012. On truth discovery in social sensing: A maximum likelihood estimation approach. In *Information Processing in Sensor Networks (IPSN), 2012 ACM/IEEE 11th International Conference on*. IEEE, 233–244.

[16] Shiguang Wang, Dong Wang, Lu Su, Lance Kaplan, and Tarek F Abdelzaher. 2014. Towards cyber-physical systems in social spaces: The data reliability challenge. In *Real-Time Systems Symposium (RTSS), 2014 IEEE*. IEEE, 74–85.

[17] Shuochao Yao, Md Tanvir Amin, Lu Su, Shaohan Hu, Shen Li, Shiguang Wang, Yiran Zhao, Tarek Abdelzaher, Lance Kaplan, Charu Aggarwal, et al. 2016. Recursive ground truth estimator for social data streams. In *Information Processing in Sensor Networks (IPSN), 2016 15th ACM/IEEE International Conference on*. IEEE, 1–12.

[18] Shuochao Yao, Md Tanvir Amin, Lu Su, Shaohan Hu, Shen Li, Shiguang Wang, Yiran Zhao, Tarek Abdelzaher, Lance Kaplan, Charu Aggarwal, and Aylin Yener. 2016. Recursive Ground Truth Estimator for Social Data Streams. In *Proceedings of the 15th International Conference on Information Processing in Sensor Networks (IPSN '16)*. IEEE Press, Piscataway, NJ, USA, Article 14, 12 pages. http://dl.acm.org/citation.cfm?id=2959355.2959369

[19] Xiaoxin Yin, Jiawei Han, and S Yu Philip. 2008. Truth discovery with multiple conflicting information providers on the web. *IEEE Transactions on Knowledge and Data Engineering* 20, 6 (2008), 796–808.

[20] Xiaoxin Yin and Wenzhao Tan. 2011. Semi-supervised truth discovery. In *Proceedings of the 20th international conference on World wide web*. ACM, 217–226.