Provenance-assisted Classification in Social Networks

Dong Wang*, Md Tanvir Al Amin*, Tarek Abdelzaher*, Dan Roth*,
Clare Voss†, Lance Kaplan†, Stephen Tratz†, Jamal Laoudi†, Douglas Briesch†
*Department of Computer Science, University of Illinois at Urbana Champaign, Urbana, IL 61801
Email: {dwang24, maamin2, zaher, danr}@illinois.edu
†US Army Research Laboratory, Adelphi, MD 20783
Email: {clare.r.voss.civ,lance.m.kaplan.civ,stephen.c.tratz.civ, jamal.laoudi.ctr, douglas.m.briesch.civ}@mail.mil

***Abstract*—Signal feature extraction and classification are two common tasks in the signal processing literature. This paper investigates the use of source identities as a common mechanism for enhancing the classification accuracy of *social signals*. We define social signals as outputs, such as microblog entries, geotags, or uploaded images, contributed by users in a social network. Many classification tasks can be defined on such outputs. For example, one may want to identify the dialect of a microblog contributed by an author, or classify information referred to in a user's tweet as true or false. While the design of such classifiers is application-specific, social signals share in common one key property: they are augmented by the explicit identity of the source. This motivates investigating whether or not knowing the source of each signal (in addition to exploiting signal features) allows the classification accuracy to be improved. We call it *provenance-assisted* classification. This paper answers the above question affirmatively, demonstrating how source identities can improve classification accuracy, and derives confidence bounds to quantify the accuracy of results. Evaluation is performed in two real-world contexts: (i) fact-finding that classifies microblog entries into true and false, and (ii) language classification of tweets issued by a set of possibly multi-lingual speakers. We also carry out extensive simulation experiments to further evaluate the performance of the proposed classification scheme over different problem dimensions. The results show that provenance features significantly improve classification accuracy of social signals, even when no information is known about the sources (besides their ID). This observation offers a general mechanism for enhancing classification results in social networks.**

I. Introduction

The emergence of social networks in recent years opens myriad new opportunities for extracting information from artifacts contributed by social sources. A significant amount of data mining literature has recently concerned itself with social network analysis. While much of that literature explores clever heuristics, recent work demonstrated that some classes of such data mining problems (such as fact-finding [1], [2]) have a rigorous estimation-theoretic formulation, amenable to well-understood solutions that use maximum-likelihood estimation techniques to accurately assess the quality of analysis results [3], [4].

This paper explores the link between estimation theory and social networks by addressing the problem of social signal classification. Generalizing from the special case of fact-finding [1]–[4], we define social signals as outputs, such as microblog entries, geotags, or uploaded images, contributed by users in a social network. We then consider the classification problem of such outputs.[1] Unlike signals generated by the physical environment (such as magnetic field or sound), where the source of the signal is often a physical object yet to be identified, in social networks the source of a social signal is usually explicitly indicated. For example, microblog entries uploaded on Twitter include the user ID. So do images uploaded on Flickr and videos uploaded on YouTube. The ubiquity of source ID information begs the question of whether it can assist with classification tasks defined on social signals such as identifying the location depicted in an uploaded image, the language used in a tweet, or the veracity of a claim.

Current classifiers address their classification tasks by exploiting domain-specific features, such as visual clues in an image and linguistic features in text, to perform the classification. The question posed in this paper is whether (and to what degree) using source identity will enhance classification results. Clearly, the more one knows about the source, the better the enhancement. To compute a worst case, we assume that one does not know *anything* about the sources other than their IDs. This assumption is often true when users find content on social networks that comes from arbitrary sources. The research question addressed in this paper is to understand to what degree knowledge of source ID alone, and without any additional information about the sources, may enhance classifier performance. Such an enhancement can then be generally applied to any classification task in social networks where source IDs are available.

One approach for incorporating source identity into the classification problem is to add it as a feature into the classifier. This may be cumbersome, however, as different classifiers are usually employed for different types of signals. For example, image classifiers and language classifiers are quite different, which may require different solutions for incorporating source information. Rather than having to change different classifiers by incorporating source identities as features, the approach we take is a general one, where the original domain-specific classifier remains unchanged. Instead, source identities are considered in a separate step that is independent of the domain-specific classifier design. This step operates on classifier output

[1] The fact-finding problem can be thought of as a special case of classification where one needs to classify claims into true and false.

with the aim of improving classification results. We show that this refinement step can be formulated as a maximum-likelihood estimation problem.

We evaluate the approach in two real world application scenarios: (i) a fact-finding application where noisy microblog data are classified into true and false facts, and (ii) a language classification application where Arabic microblogs from Twitter are classified into different dialects. Our evaluation results show that the scheme proposed in this paper significantly improves classification accuracy of conventional classifiers by leveraging the provenance information. We also carry out extensive simulation experiments to examine the performance of our classification enhancement scheme in different scenarios. The results verify its scalablility and robustness over several key problem dimensions.

The rest of the paper is organized as follows. In Section III, we present our signal classification model in social networks. We discuss the proposed maximum likelihood estimation approach to improve the classification accuracy in Section IV. The theoretical accuracy bounds that are used to quantify the quality of the results are derived in Section V. Experimental evaluation results are presented in Section VI. We review related work in Section II. Finally, we discuss the limitations of the current model and future work in Section VII, and conclude the paper in Section VIII.

II. Related Work

Classification is an fundamental problem that has been extensively studied in machine learning, data mining, statistics and pattern recognition. A comprehensive overview of different classification schemes is described in [5], [6]. Our work augments prior classification literature in the context of classifying social signals. The current work studied the classification of nodes and relationships in social networks [7], [8] as well as human related features [9]. In contrast, we take advantage of the fact that signals in social networks, unlike physical signals in other application scenarios, explicitly mention source ID. Hence, we develop a new provenance-assisted scheme for enhancing classification results by taking into account provenance information in a separate step using a maximum-likelihood estimation approach. Our approach explicitly improves classification accuracy by jointly uncovering classes of artifacts and affinities of sources to generating artifacts of specific classes.

One application of our classification scheme has been fact-finding. Techniques for classifying true facts from false ones are traced back to data mining and machine learning literature. One of the early works is Hubs and Authorities [10] that used a basic fact-finder where the belief in a claim and the truthfulness of a source are jointly computed in a simple iterative way. 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* [11]. Yin et al. introduced *TruthFinder* as an unsupervised fact-finder for trust analysis on a providers-facts network [12]. Other fact-finders enhanced the basic framework by incorporating analysis on properties or dependencies within claims or sources. Galland et al. [13] took the notion of hardness of facts into consideration by proposing their algorithms: *Cosine, 2-Estimates, 3-Estimates*. Similar iterative algorithms have also been studied in the context of crowdsourcing applications to minimize the budget cost while optimizing overall quality of answers from crowdsourced workers [14]. While such prior work was essentially heuristic in nature, an optimal solution to (a simplified version of) the problem was recently proposed [1] in the context of a simple social sensing model, demonstrating improved performance. In contrast, this paper solved a more general classification problem beyond fact-finding where the possible values of artifacts are not limited to *binary* values.

Our classifier enhancement scheme is based on expectation maximization. In estimation theory, Expectation Maximization (EM) is a general optimization technique for finding the maximum likelihood estimation of parameters in a statistic model where the data are "incomplete" or involve latent variables in addition to estimation parameter and observed data [15]. EM is frequently used for data clustering in data mining and machine learning [16], [17]. For language modeling, the EM is often used to estimate parameters of a mixed model where the exact model from which the data is generated is unobservable [18]–[20]. EM is also used in many other estimation tasks involving mixture distributions including parameter estimation for hidden Markov models with applications in pattern recognition, image reconstruction, error correction codes, etc [21], [22].

III. Problem Formulation

Consider a social network of M sources, who collectively generate the social signals we want to classify. We henceforth call such signals the *artifacts*. Let there be a total of N artifacts, where each artifact can have one of K possible classes. The classification problem is to determine the class of each artifact. Many problems fall into the above category. Below, three examples are presented:

- *Fact-finders:* A fact-finder considers M sources, who collectively generate N claims (the artifacts). Each claim is in one of two possible classes (i.e., $K = 2$), *true* or *false*. The fact-finder must determine which claims are true and which are false.
- *Language classification:* In a language classification problem, there may be M authors (the sources), who collectively write N words (the artifacts). Each word could be in one of K languages. The goal is to identify the language of each word.
- *Automated geo-tagging of text:* In a geo-tagging problem, there may be M bloggers (the sources), who collectively describe a set of N events (the artifacts). Each event may take place at one of K locations, not explicitly marked in the blog. The goal is to identify the location associated with each event implicitly from the text.

The traditional classification approach is to take the artifacts in isolation and find the best class for each using specialized domain knowledge. For example, a language classifier can use linguistic features to identify the language of a word.

This paper augments that with the exploitation of provenance information. By provenance, for purposes of this work, we refer to the identity of the source(s) of each artifact. We do not assume that we know any information about the sources other that their IDs. While, clearly, knowing some background about the sources will help, this is not the point of this work. The paper investigates to what degree the knowledge of source ID alone helps classification outcomes.

The intuitive reason why provenance information (i.e., source IDs) should help with classification is that sources have *affinity* to generating artifacts of particular types. For example, a person from Egypt might have an affinity to writing in Arabic, a truthful person might have an affinity of generating tweets of type “true”, and a person who commutes in Los Angeles might have an affinity to complaining about LA traffic. Said differently, sources constrain the probability distribution of the classes of artifacts they generate. These constraints are automatically estimated and explicitly accounted for in the mathematical formulation of our algorithm, which then forces the solution to obey them.

According to our terminology, multiple sources can “generate” the same artifact. For example, multiple tweeters can make the same claim, multiple authors can use the same word, and multiple bloggers can describe the same event. The input to our problem describes which sources generated which artifacts. This input is given by a bipartite graph as shown in Figure 1, where nodes represent sources and artifacts, and where a link exists between a source and an artifact if and only if the source generated that artifact. We call this set the observed input, X. The class of each artifact is unknown and is represented by a latent variable. The vector of all such latent variables is called Z.

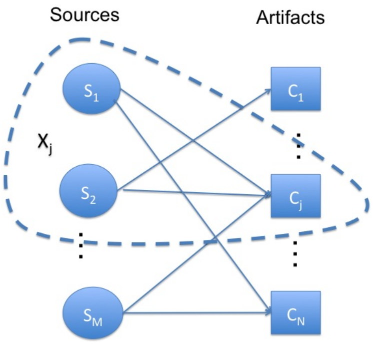


Fig. 1. Input Bipartite Graph

We also define $a_{i,k}$ as the (unknown) probability that source S_i generates an artifact C_j given that C_j is of class k (e.g., the odds that source S_i speaks word C_j given that C_j is from a certain language k). Formally, $a_{i,k}$ is defined as follows:

$$a_{i,k} = P(S_i \rightarrow C_j | Class(C_j) = k) \tag{1}$$

We also define $X_{i,j} = 1$ if S_i generates artifact C_j (i.e, $S_i \rightarrow C_j$), and $X_{i,j} = 0$ otherwise. Moreover, let us denote the probability that an artifact is of class k, given that it was generated by source i, as $t_{i,k}$. These probabilities represent source affinities, referred to above. Formally, $t_{i,k}$ is given as:

$$t_{i,k} = P(Class(C_j) = k | S_i \rightarrow C_j) \tag{2}$$

Using Bayes theorem, $t_{i,k}$ is related to $a_{i,k}$ as follows:

$$t_{i,k} = \frac{a_{i,k} \times d_k}{s_i} \tag{3}$$

where $s_i = P(S_i \rightarrow C_j)$ represents the probability of a source to generate artifacts (i.e., artifact production rate) and $d_k = P(Class(C_j) = k)$ is the overall prior of a randomly chosen artifact to be of class k.

Our problem is to jointly estimate (i) the latent variable vector Z (i.e., the class of each artifact), and (ii) the source affinities, $t_{i,k}$, that can be computed from the estimation parameter vector $\theta = (\theta_1, \theta_2, ..., \theta_K)$, where $\theta_k = (a_{1,k}, a_{2,k}, ..., a_{M,k})$.

IV. Solution

In this section, we cast the problem of jointly (i) classifying the values of artifacts and (ii) computing source affinities as a maximum likelihood estimation problem. A maximum-likelihood estimator is then derived to solve it. The maximum likelihood estimator finds the values of the unknown parameters (i.e., θ) that maximize the probability of observed input X. Hence, we would like to find θ that maximizes $P(X|\theta)$. The probability $P(X|\theta)$ depends on which artifacts belong to which classes (i.e., the values of latent variables z). Using the total probability theorem, we can now rewrite the expression we want to maximize, namely $P(X|\theta)$, as follows:

$$P(X|\theta) = \sum_z P(X, z|\theta) \tag{4}$$

We solve this problem using the Expectation Maximization (EM) algorithm that starts with some initial guess for θ, say θ_0 and iteratively updates it using the formula:

$$\theta_{t+1} = argmax_\theta \{E_{z|X,\theta_t} \{\log P(X, z|\theta)\}\} \tag{5}$$

The above breaks down into three quantities that need to be derived:

- The log likelihood function, $\log P(X, z|\theta)$
- The expectation step (E-step), $Q\left(\theta|\theta^{(t)}\right) = E_{z|X,\theta_t}\{\log P(X, z|\theta)\}$
- The maximization step (M-step), $\theta_{t+1} = argmax_\theta Q\left(\theta|\theta^{(t)}\right)$

Note that, the E-step and M-step are computed iteratively until the algorithm converges. The above likelihood functions are derived below.

A. Deriving the Likelihood

To compute the log likelihood, we first compute the function $P(X, z|\theta)$. Let us divide the source and artifact bipartite graph X into sub-graphs, X_j, one per artifact C_j. The sub-graph describes which sources generate the artifact and which did not. Since artifacts are independent, we can re-write:

$$P(X, z|\theta) = \prod_{j=1}^{N} P(X_j, z_j|\theta) \tag{6}$$

which can in turn be re-written as:

$$P(X, z|\theta) = \prod_{j=1}^{N} P(X_j|\theta, z_j)P(z_j) \tag{7}$$

where $P(X_j|\theta, z_j)$ is the joint probability of all observed input involving artifact C_j.

Considering each artifact could have K possible class values, Equation (8) can be further rewritten as follows:

$$P(X, z|\theta) = \prod_{j=1}^{N} \sum_{j=1}^{K} P(X_j|\theta, z_j = k)P(z_j = k) \tag{8}$$

Hence, the likelihood function, denoted by $L(\theta; X, z)$, is given by:

$$\begin{aligned} & L(\theta; X, z) = p(X, z|\theta) \\ & = \prod_{j=1}^{N} \Big\{ \sum_{k=1}^{K} \prod_{i=1}^{M} a_{i,k}{}^{X_{i,j}} (1 - a_{i,k})^{(1-X_{i,j})} \times d_k \times z_j^k \Big\} \end{aligned} \tag{9}$$

where $a_{i,k}$ is defined in Equation (1) and $X_{i,j} = 1$ if source S_i generate artifact C_j and $X_{i,j} = 0$ otherwise. d_k represents the overall prior probability that an arbitrary artifact is of class k. Let $z_j^1, z_j^2, ..., z_j^K$ be a set of indicator variables for artifact C_j, where $z_j^k = 1$ when C_j is of class k and $z_j^k = 0$ otherwise. We now formulate an expectation maximization algorithm (EM) that jointly estimates the parameter vector θ and the indicator variables, z_j^k.

B. Deriving the E-step and M-step

Given the above formulation, substitute the likelihood function defined in Equation (9) into the definition of Q function of Expectation Maximization. The Expectation step (E-step) becomes:

$$\begin{aligned} & Q\left(\theta|\theta^{(t)}\right) = E_{Z|X,\theta^{(t)}}[\log L(\theta; X, Z)] \\ & = \sum_{j=1}^{N} \Bigg\{ \sum_{k=1}^{K} p(z_j = k|X_j, \theta^{(t)}) \\ & \times \left[\sum_{i=1}^{M} (X_{i,j} \log a_{i,k} + (1 - X_{i,j}) \log(1 - a_{i,k}) + \log d_k) \right] \Bigg\} \end{aligned} \tag{10}$$

where X_j represents the observed links from all sources to the j^{th} artifact. Let the latent variable z_j be defined for C_j such that: $z_j = k$ when C_j is of class k. Let $p(z_j = k|X_j, \theta^{(t)})$ be the conditional probability that the variable z_j is of class k given the observed data related to the j^{th} artifact and current estimate of θ. $p(z_j = k|X_j, \theta^{(t)})$ can be further derived as:

$$\begin{aligned} Z(t, j, k) &= p(z_j = k|X_j, \theta^{(t)}) \\ &= \frac{p(z_j = k; X_j, \theta^{(t)})}{p(X_j, \theta^{(t)})} \\ &= \frac{p(X_j, \theta^{(t)}|z_j = k)p(z_j = k)}{\sum_{k=1}^{K} p(X_j, \theta^{(t)}|z_j = k)p(z_j = k)} \\ &= \frac{A(t, j, k) \times d_k}{\sum_{k=1}^{K} A(t, j, k) \times d_k} \end{aligned} \tag{11}$$

where $A(t, j, k)$ is defined as:

$$\begin{aligned} A(t, j, k) &= p(X_j, \theta^{(t)}|z_j = k) \\ &= \prod_{i=1}^{M} a_{i,k}^{(t)X_{i,j}} (1 - a_{i,k}^{(t)})^{(1-X_{i,j})} \end{aligned} \tag{12}$$

Substituting Equation (11) into Equation (10), we get:

$$\begin{aligned} & Q\left(\theta|\theta^{(t)}\right) \\ & = \sum_{j=1}^{N} \Bigg\{ \sum_{k=1}^{K} Z(t, j, k) \\ & \times \left[\sum_{i=1}^{M} (X_{i,j} \log a_{i,k} + (1 - X_{i,j}) \log(1 - a_{i,k}) + \log d_k) \right] \Bigg\} \end{aligned} \tag{13}$$

For the Maximization step (M-step), we choose θ^* (i.e., $(a_{1,k}^*, a_{2,k}^*, ...a_{M,k}^*)$ for $k = 1, 2, ..., K$) that maximizes the $Q\left(\theta|\theta^{(t)}\right)$ function in each iteration to be the $\theta^{(t+1)}$ of the next iteration.

To get θ^* that maximizes $Q\left(\theta|\theta^{(t)}\right)$, we set the derivatives $\frac{\partial Q}{\partial a_{i,k}} = 0$, which yields:

$$\sum_{j=1}^{N} \left[Z(t, j, k)(X_{i,j} \frac{1}{a_{i,k}^*} - (1 - X_{i,j}) \frac{1}{1 - a_{i,k}^*}) \right] = 0 \tag{14}$$

Let us define SJ_i is the set of artifacts the source S_i actually generates, and $S\bar{J}_i$ is the set of artifacts S_i does not generate. Thus, Equation (14) can be rewritten as:

$$\sum_{j \in SJ_i} Z(t, j, k) \frac{1}{a_{i,k}^*} - \sum_{j \in S\bar{J}_i} Z(t, j, k) \frac{1}{1 - a_{i,k}^*} = 0 \tag{15}$$

Solving the above equations, we can get expressions of the optimal $a_{i,k}^*$:

$$a_{i,k}^{(t+1)} = a_{i,k}^* = \frac{\sum_{j \in SJ_i} Z(t, j, k)}{\sum_{j=1}^{N} Z(t, j, k)} \tag{16}$$

where N is the total number of artifacts we have. $Z(t, j, k)$ is defined in Equation (11).

Given the above, The E-step and M-step of EM optimization reduce to simply calculating Equation (11) and Equation (16) iteratively until they converge. The convergence analysis has been done for EM scheme and it is beyond the scope of this paper [23]. In practice, we can run the algorithm until the difference of estimation parameter between consecutive iterations becomes insignificant. We can then classify the classes of artifacts based on the converged value of $Z(t, j, k)$. Specially, C_j is of class k if $Z(t, j, k)$ is the largest for $k = 1, 2, ..K$. We can also compute the values of t_i^k from the values of the estimation parameters based on Equation (3). This completes the mathematical development. We summarize the resulting algorithm in the subsection below.

C. Final Algorithm

Algorithm 1 Provenance-Assisted General Classifier

```
Initialize parameter vector $\theta$
while $\theta^{(t)}$ does not converge do
    for $j = 1 : N$ do
        compute $Z(t, j, k)$ based on Equation (11)
    end for
    $\theta^{(t+1)} = \theta^{(t)}$
    for $i = 1 : M$ do
        compute $a_{i,k}^{(t+1)}$ based on Equation (16)
        update $a_{i,k}^{(t)}$ with $a_{i,k}^{(t+1)}$ in $\theta^{(t+1)}$
    end for
    $t = t + 1$
end while
Let $Z_{j,k}^c$ = converged value of $Z(t, j, k)$
Let $a_{i,k}^c$ = converged value of $a_{i,k}^{(t)}$
for $j = 1 : N$ do
    Let $k^*$ = the class of artifact $C_j$who has maximum$Z_{j,k}^c$
    $C_j$ is of Class $k^*$
end for
for $i = 1 : M$ do
    calculate $t_{i,k}^*$ from $a_{i,k}^c$ based on Equation (3)
end for
Return the computed optimal estimates of class $k^*$ for each artifact $C_j$ and the probability of a source to generate a specific class of artifacts (i.e.,$t_{i,k}^*$).
```

In summary of the EM classification scheme derived above, the input is the source artifact graph X describing which sources generate which artifacts and the output is an estimate of the class of each artifact, as well as an estimate of source affinities.

In particular, given the source artifact graph X, our algorithm begins by initializing the parameter θ. The algorithm then iterates between the E-step and M-step until θ converges. Specifically, we compute the conditional probability of an artifact to be of class k (i.e., $Z(t, j, k)$) from Equation (11) and the estimation parameter (i.e., $\theta^{(t+1)}$) from Equation (16). Finally, we can decide whether each artifact C_j is of class k based on the converged value of $Z(t, j, k)$ (i.e., $Z_{j,k}^c$). The pseudocode of the provenance assisted (PA) classification algorithm is shown Algorithm 1.

D. Enhancing an Arbitrary Classifier

The above algorithm can be executed as an enhancement stage for any arbitrary (domain-specific) classifier of social signals. There are two different ways that such an enhancement can be added.

In the first approach, the enhanced system runs the arbitrary (domain-specific) classifier first. Assuming that the original classifier can tell when it is very confident in its labels, and when it is not, we can import from that classifier only labels of those artifacts in which it is very confident. These labels are treated as the ground truth estimate of the corresponding subset of the indicator variables vector, Z, used by our algorithm. Remember that the indicator variable vector, Z, in our iterative algorithm states the class of each artifact. A subset of the indicator variables is thus determined by the domain specific classifier. The rest are initialized at random and the above iterations are carried out updating their values until they converge.

In the second approach, the enhanced system runs the domain-specific classifier to obtain an initial guess of the class of *all* artifacts. These results will presumably contain misclassifications. Hence, the labels generated by the domain classifier are used as initial values for the indicator variable vector Z. Our algorithm is then executed to update these initial estimates. The converged values of these variables should improve upon the initial guess (i.e., upon the domain classification results).

In the first scenario above, the improvement is obvious. Our algorithm fills in labels that the domain classifier was unsure of. In the second scenario, the intuitive reason why we achieve a performance improvement is that our algorithm starts with the output of the traditional classifier, which is already close to the right answer and "snaps it" to the locus of points that maximize likelihood of observations in view of constraints that relate the probability distributions computed for sources and the probabilities of the classes of their artifacts. This "snapping" therefore uses additional information on source-artifact relations, not furnished to the traditional classifier. Namely, it obeys laws of probability and Bayesian equations that relate source affinities and artifact classes.

V. Accuracy Bound

In the previous section, we derived a classification enhancement scheme that takes the provenance of artifacts into account. However, one important question remains: how to quantify the estimation accuracy of the resulting enhanced classifier? In particular, we are interested in obtaining the confidence intervals; namely, the error bounds on the estimation parameters of our model for a given confidence level. In this section, we derive such Cramer-Rao lower bounds (CRLB).

A. Deriving Error Bounds

We start with the derivation of Cramer-Rao lower bounds for our problem. The CRLB states the lower bounds of estimation variance that can be achieved by the maximum likelihood estimation (MLE).

We follow similar derivation steps in [4] and the derived asymptotic CRLB of our problem is shown as follows:

$$(J^{-1}(\hat{\theta}_{MLE}))_{i,j} = \begin{cases} 0 & i \neq j \\ \frac{\hat{a}_{i,k}^{MLE} \times (1 - \hat{a}_{i,k}^{MLE})}{N \times d_k} & i = j \end{cases} \qquad (17)$$

Note that, the asymptotic CRLB is independent of M (i.e., number of sources) under the assumption that M is sufficient, and it can be quickly computed.

B. Confidence Interval

One of the attractive asymptotic properties about maximum likelihood estimator is called *asymptotic normality*: The MLE estimator is asymptotically distributed with Gaussian behavior as the data sample size goes up. The variance of estimation error on parameter $a_{i,k}$ is denoted as $var(\hat{a}_i^{MLE})$. For a problem with sufficient M and N (i.e., under asymptotic condition), $(\hat{t}_{i,k}^{MLE} - t_{i,k}^0)$ also follows a norm distribution with 0 mean and variance given by:

$$var(\hat{t}_{i,k}^{MLE}) = \left(\frac{d_k}{s_i}\right)^2 var(\hat{a}_{i,k}^{MLE}) \tag{18}$$

Thus, the confidence interval that can be used to quantify the probability a source S_i generates a given class k of artifacts (i.e., $t_{i,k}$) is given by the following:

$$(\hat{t}_{i,k}^{MLE} - c_p\sqrt{var(\hat{t}_{i,k}^{MLE})}, \hat{t}_{i,k}^{MLE} + c_p\sqrt{var(\hat{t}_{i,k}^{MLE})}) \tag{19}$$

where c_p is the standard score (z-score) of the confidence level p. For example, for the 95% confidence level, $c_p = 1.96$.

VI. Evaluation

In this section, we first evaluate the performance of the provenance-assisted (PA) classifier described in this paper through two real world application scenarios including a fact-finding application using geotagging data and an Arabic dialect classification application using Twitter data feeds. We then carry out extensive simulation experiments to study the performance of the PA classifier over different problem dimensions. The results show that our scheme significantly improves classification accuracy compared to traditional classifiers by using the source ID as additional information.

A. Fact-finding Example

Fact-finding is a common type of analysis applied to data (typically text) uploaded to social networks. The goal of fact-finding is to estimate the probability of correctness of claims made in the text. In this experiment, we generate a scenario where ground truth is known. Namely, we develop a "parking lot finder" application, that helps students identify free parking lots on campus (at the University of Illinois at Urbana Champaign). "Free parking lots" refer to parking lots that are free of charge after 5pm on weekdays (as well as weekends). The application allows volunteers to identify parking lots they think are free. This information is shared with others. It also runs our algorithm in the background to compute the right class for each parking lot: either "free" or "pay". For evaluation, we collected ground truth by visiting all parking lots in question and accurately inspecting their posted signs. Note that, a slightly different version of this application was published in the context of handling conflicting claims [2]. The current evaluation is different is that (interpreting fact-finding as a fact classification problem) we aim to understand the degree to which fact classification results are improved when our expectation maximization algorithm runs as a second stage after an initial solution is computed by another fact-finder.

In the experiment, 30 participants were recruited. Recruited volunteers were asked to mark any parking lots they thought were free. Participants were not asked to visit all parking lots in the area. Rather, they were asked to mark parking lots at will (e.g., those parking lots they are familiar with). Collectively, they surveyed 106 parking lots (46 of which were indeed free). There were a total of 423 reports (notations claiming a "free parking lot") collected from these participants.

We note that there are many different types of parking lots on campus: enforced parking lots with time limits, parking meters, permit parking, and others. Different parking lots have different regulations for free parking. Moreover, instructions and permit signs sometimes are easy to miss. Hence, our participants suffered both false positives and false negatives in their reports. Moreover, participants differed in their reliability (i.e., affinity to generating correct responses). Some actually visited the parking lots in person and carefully inspected the posted signs. Others, reported results from memory.

In our evaluation, three different fact-finding schemes are first employed. Our expectation maximization algorithm is then applied to their output. Specifically, Average-Log [11], Truth-Finder [12] and a Voting scheme were used to provide three different initial guesses regarding artifact classification. The voting scheme considered a parking lot to be "free" if it was reported free by at least a given number of volunteers. This threshold was varied in the evaluation results shown later.

To run our provenance-assisted (PA) expectation maximization algorithm, we generated the source to artifact bipartite graph (i.e., observed input X) taking the participants as sources and parking lots as artifacts. The artifacts were assigned class "free" or "pay" depending on the results of Average-Log [11], Truth-Finder [12], or the voting scheme, respectively. Our scheme then performed its iterations until they converged. The receiver operating characteristics (ROCs) curves computed by these schemes as well as the final solution of our provenance assisted (PA) classifier are shown in Figure 2.

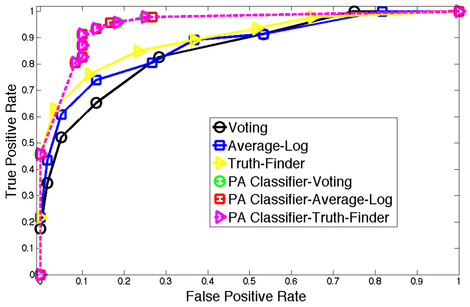


Fig. 2. ROCs of Different Fact-Finders

We observe that the PA classifier achieved the best ROCs performance among all schemes under comparison. The reason is that our PA classifier modeled the provenance information of the artifacts explicitly and used the MLE approach to find the value of each claim that is most consistent with the observations we had. We also observed that the EM algorithm converges to the ML solution given a reasonable initialization.

As a result, the PA classifier is insensitive to the initial guess provided by the other classifiers.

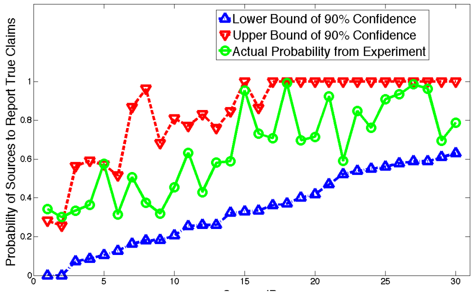


Fig. 3. Source probability to report true claims

We also evaluated the probability of a source to report true claims ("free parking lot") and the confidence bounds we derived to quantify its accuracy. We calculated the 90% confidence bounds based on the formula derived in Sections V. The results are shown in Figure 3. The sources are sorted based on the value of the lower bounds at 90% confidence. We observe that there are only 2 sources out of 30 whose probability to report true claims was outside the 90% confidence bounds, which matches quite well with the definition of a 90% confindence interval (which implies that no more than 3 sources out of 30 should be outside the interval). The results verified the correctness and accuracy of the confidence bounds we derived for our PA classifier.

B. Arabic Dialect Classification

In this application, the goal is to automatically distinguish two dialects of Arabic in a set of tweets. In particular, we used Twitter to collect Arabic tweets for our experiment. The two dialects we selected were Egyptian and Morrocan. For evaluation purposes, we used two sets of key words, representing Arabic words that only appear in either Egyptian or Morrocan dialect. Then, we used them as query words to collect tweets that originated from Egypt and Morocco respectively. We collected 2945 tweets in total, including 2000 Egyptian tweets and 945 Morrocan tweets. Note that, the choice of Egyptian and Moroccan was dictated simply by available language expertise on the team, to make groundtruthing of dialects possible.

In our experiment, we applied both a domain-specific dialect classifier [24] and the provenance-assisted (PA) classifier we developed in this paper on the collected tweets and compared their performance. To use the provenance-assisted classifier, we first broke the tweets into words and removed punctuation marks, non-linguistic symbols and tags in the tweets. We then built the source artifact graph by taking the users of the tweets as sources and the words they tweeted as artifacts. There is a link between a source and an artifact if the user tweeted that word. We first ran the domain classifier on the collected tweets and obtained the dialect classification results. We then used the dialect outputs to label all words and initialized our PA classifier accordingly.

The compared results are shown in Table I. We changed the threshold of the probability to decide whether a word is Egyptian or not from 0.5 to 0.95. From Table I, we observe that, compared to the domain Classifier, the PA classifier was able to increase the accuracy of Egyptian classification by more than 10% while keeping the accuracy of Morrocan tweet classification slightly better or similar. Such performance gain is obtained by leveraging the user ID information of tweets. We also observed that the PA classifier performance is consistent and robust when the threshold value was varied in the experiment.

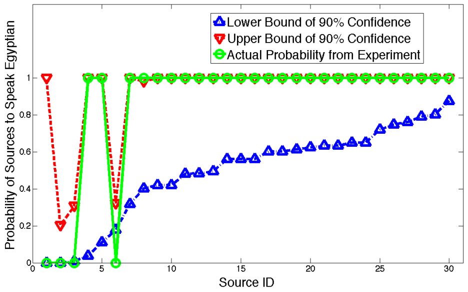


Fig. 4. Source probability to speak Egyptian

We also studied the probability of a source to speak a given dialect and the confidence bounds we derived to quantify its accuracy. For demonstration purposes, we randomly picked 30 sources and computed the probability that the source speaks Egyptian from the tweets he/she actually tweeted. We also calculated the 90% confidence bounds based on the formula derived in Sections V. The results are shown in Figure 4. We observe that in this case there are only 3 sources out of 30 whose probability to speak Egyptian was outside the 90% confidence interval, which means that indeed exactly 90% of the sources fall withing the interval.

C. Simulation Study

The above experiments represent only two points in the space of possible datasets to apply classifiers to. They feature datasets with only two classes and a limited number of sourses. To explore performance more broadly, in this subsection, we carried out extensive simulation experiments evaluating the provenance-assisted (PA) classification scheme along different problem dimensions.

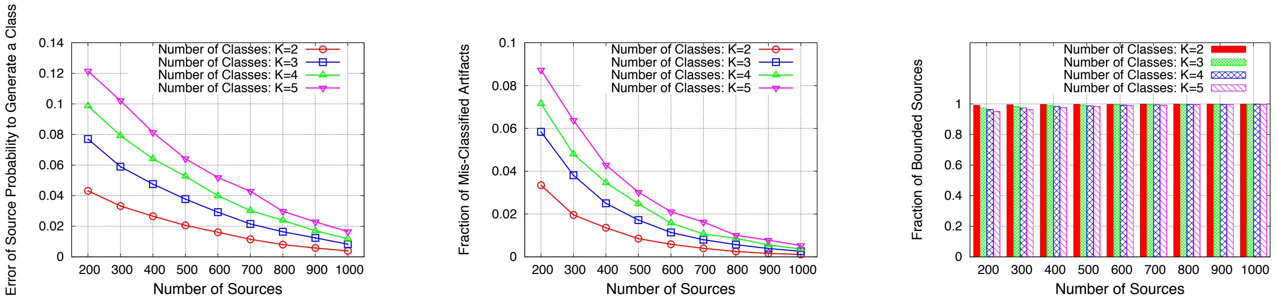
We built a simulator in Matlab 7.10.0 that generates a random number of sources and artifacts. A probability t_i^k is assigned to each source S_i representing his/her probability to generate artifacts of a given class k. For each source S_i, L_i artifacts are generated. We ensure that $\sum_{k=1}^{K} t_i^k = 1$.

In the evaluation, we mainly studied three metrics: (i) the average estimation error of t_i^k normalized by its mean value; (ii) the average classification error; (iii) the fraction of sources whose t_i^k are within the confidence bounds we derived in Section V.

1) Source-Artifact Graph Topology - Sources: In the first experiment, we evaluate the estimation accuracy of the PA classifier by varying the number of sources in the system. The number of generated artifacts was fixed at 3000. The average number of artifacts generated per source was set to

TABLE I
ARABIC DIALECTS CLASSIFICATION RESULTS

	Domain Classifier	**PA Classifier (Threshold=0.5)**	**PA Classifier (Threshold=0.75)**	**PA Classifier (Threshold=0.95)**
Correctly Classified Egyptian Tweets	1645	**1860**	**1855**	**1839**
Correctly Classified Morrocan Tweets	852	**840**	**850**	**861**
Egyptian Tweet Classification Accuracy	82.3%	**93%**	**92.8%**	**92%**
Morrocan Tweet Classification Accuracy	90.2%	**88.9%**	**90%**	**91.1%**



(a) Source Probability Estimation Error

(b) Fraction of Misclassified Artifacts

(c) Fraction of Sources within 90% Confidence Interval

Fig. 5. Changing the number of sources. PA classifier operates as an add-on to a domain classfier that leaves 50% of the artifacts unlabeled.

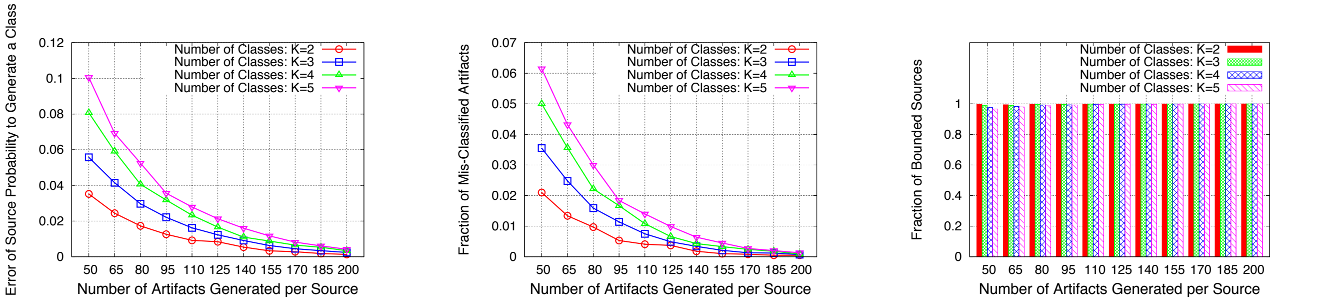
50. We assumed that a domain-specific classifier has already labeled half the artifacts with class labels. This is to emulate the case where the initial classifier was sure of only 50% of the data. Our PA classifier used the labeled artifacts to initialize the EM algorithm and figure out the classes of the *unlabeled* ones. The number of sources was varied from 200 to 1000. In this initial experiment, the probability that a source generates artifacts of a given class, t_i^k, was drawn at random from a uniform distribution. In some sense, this offers a worst-case for our classifier, as it indicates absence of a clear affinity between sources and artifacts. (Later, we show experiments with stronger affinity models.) The number of classes K was varied from 2 to 5. Reported results are averaged over 50 random distributions of t_i^k.

Results are shown in Figure 5. Observe that the PA classifier estimation accuracy improves as the number of sources in the system increases. Given sufficient sources, the estimation error in t_i^k, and artifact classification error are kept well below 5%. We also note the fraction of sources whose t_i^k are actually bounded by the 90% confidence interval is normally around or above 90%, which verifies the accuracy of the confidence intervals we derived. Additionally, we observe that the performance of the PA classifier increases as the number of classes K decreases. The reason is that the number of estimation parameters becomes smaller.

2) Source-Artifact Graph Topology - Artifacts: The second experiment studies the performance of the PA classifier when the average number of artifacts generated per source changes. As before, the number of generated artifacts was fixed at 3000. The average number of sources was set to 300. The fraction of labeled artifacts (presumably by a domain specific classifier) was set to 0.5. The number of artifacts generated per source was varied from 50 to 200. The number of classes K was varied from 2 to 5. Reported results are averaged over 50 random distributions of t_i^k. Results are shown in Figure 6. Observe that the PA classifier estimation accuracy improves as the number of generated artifacts per source increases. This is because more artifacts simply provide more evidence for the PA classifier to figure which artifact belongs to which class. Similarly, we note the fraction of sources whose t_i^k are bounded by the 90% confidence interval are indeed above 90%. Additionally, we also observe similar trend of performance increase of the PA classifier as the number of classes (i.e., K) decreases.

3) Fraction of Labeled Atrifacts: The third experiment examines the effect of changing the fraction of the labeled artifacts on the PA classifier. We vary the fraction of labeled artifacts by the domain classifier from 0.1 to 0.9, while fixing the total number of artifacts to 3000. The average number of artifacts generated per source was set to 50. The number of sources was set to 300. The number of classes K was varied from 2 to 5. Reported results are averaged over 50 random distributions of t_i^k. Results are shown in Figure 7. Observe that the PA classifier estimation error reduces as the fraction of labeled artifacts increases. This is intuitive: more correctly labeled artifacts will help the PA classifier converge to better results. Moreover, the 90% confidence bounds are shown to be tight even when the fraction of labeled artifacts is relatively small. We also observe that the estimation performance of the PA classifier increases as the number of classes decreases.

4) Imperfect Domain Classifiers: In the above experiments, we looked at the case where some fraction of artifacts are

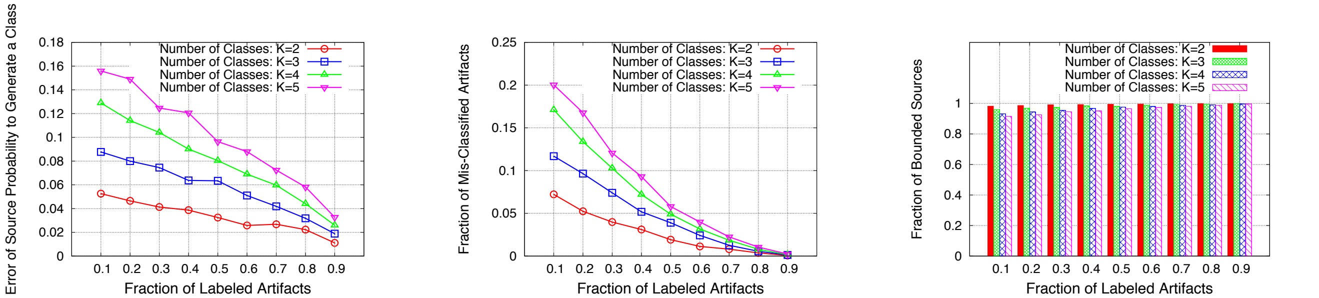


(a) Source Probability Estimation Error

(b) Fraction of Misclassified Artifacts

(c) Fraction of Sources within 90% Confidence Interval

Fig. 6. Changing the number of artifacts per source. PA classifier operates as an add-on to a domain classfier that leaves 50% of the artifacts unlabeled.



(a) Source Probability Estimation Error

(b) Fraction of Misclassified Artifacts

(c) Fraction of Sources within 90% Confidence Interval

Fig. 7. Changing the fraction of labeled artifacts

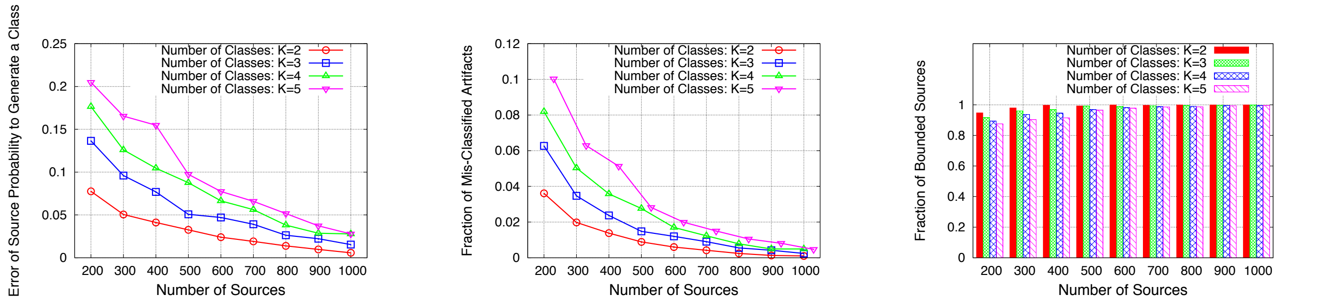
correctly labeled and the rest are not labeled. The other usage scenario for our PA classifier is one where a domain classifier labels everything, but a certain fraction of labels are wrong. In this case, the PA classifier does not view initial labels as ground truth. Instead, it simply uses them as initial values for the iterations.

We first repeated the first and second experiments using a domain classifier with imperfect class labels. The experiment setup was the same as before except that we assumed all artifacts are labeled by an imperfect classifier and the fraction of incorrect labels was set to 25%. The results are shown in Figure 8 and Figure 9. We observe that our PA classifier significantly improves the classification accuracy over the original classifiers. The fraction of mis-classified artifacts was reduced to below 10% (from 25%) for all cases we examined. These results demonstrate the capability of our classification enhancement scheme to improve the classification accuracy of imperfect classifiers.

Next we repeated experiment 3 above with an imperfect domain classifier. In this case, we assumed all artifacts are labeled and varied the fraction of incorrect labels from 0.05 to 0.5. The results are shown in Figure 10. We observed that our PA classifier performance is robust to the fraction of initial incorrect labels and is able to reduce the classification error significantly compared to the original labels. For example, when half of the initial labels are wrong, our PA classifier was able to reduce the fraction of mis-classified artifacts to about 12% for K=5. This result demonstrated the capability of the PA classifier to improve the classification accuracy of imperfect classifiers when the source information is available. In all cases, the reported results are averaged over 50 instances.

5) *Study of Source Affinity Models:* In the next experiment, we studied the effect of different source affinity models on the performance of our PA classifier. Sources may have affinity to generating certain types of artifacts (e.g., individuals living in Morocco may have an affinity to the Moroccan dialect and individuals living in Egypt may have an affinity to the Egyptian dialect). We studied three types of sources in our experiment: (i) *specialized sources*: each source produces only one class of artifacts (e.g., speaks only one language) regardless of how many classes are simulated in the data set; (ii) *semi-specialized sources*: each source uniformly produces some number of classes of artifacts that is less than the total number of classes; (iii) and *semi-specialized sources with dominant affinity*: same as semi-specialized sources, except that the odds of producing different classes of artifacts by a source are not uniform. There is a preferred class that dominates. The other classes share the remaining probability equally.

In the experiment, the number of sources was set to 300 and each source generated 50 artifacts on average. The total number of artifacts was set to 3000, and the number of classes was fixed at 5. We set the fraction of labeled artifacts to 0.5. The reported results are averaged over 50 experiments. Figure 11 showed the performance comparison between specialized sources and semi-specialized sources. We observe that the source specialization can improve the classification accuracy. The reason is that highly specialized sources have more concentrated distributions to generate given types of

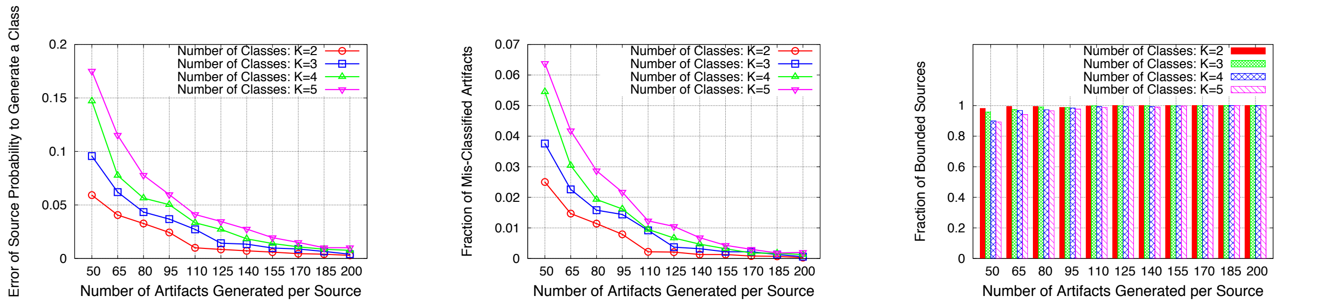


(a) Source Probability Estimation Error

(b) Fraction of Misclassified Artifacts

(c) Fraction of Sources within 90% Confidence Interval

Fig. 8. Changing the number of sources. PA classifier operates as an add-on to a domain classfier that misclassifies 25% of the artifacts.

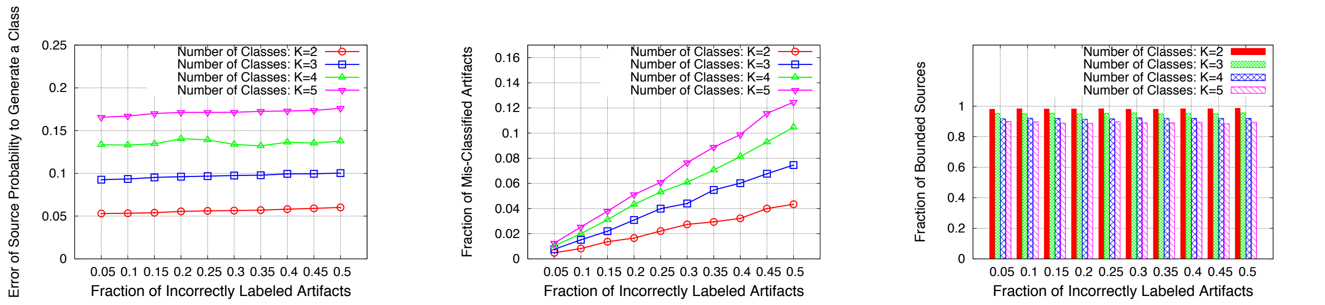


(a) Source Probability Estimation Error

(b) Fraction of Misclassified Artifacts

(c) Fraction of Sources within 90% Confidence Interval

Fig. 9. Changing the number of artifacts per source. PA classifier operates as an add-on to a domain classfier that misclassifies 25% of the artifacts.



(a) Source Probability Estimation Error

(b) Fraction of Misclassified Artifacts

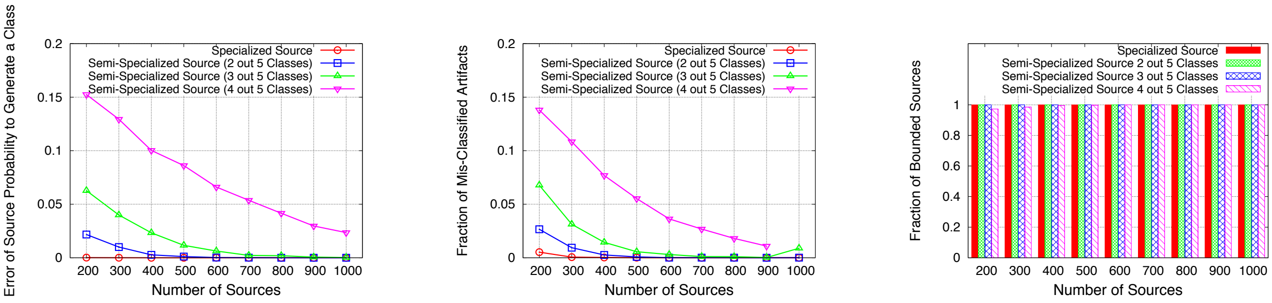
(c) Fraction of Sources within 90% Confidence Interval

Fig. 10. Changing the fraction of initially mislabeled artifacts

artifacts, which makes it easier for our classifier to differentiate artifacts of different classes. Figure 12 shows the effect of the affinity dominance of the semi-specialized sources. We observe that the classification performance of our PA classifier improves as the probability to generate the preferred class (i.e., the class that dominates) by semi-specialized sources increases. This is because the semi-specialized sources become more specialized as the probability to generate the preferred class increases.

In this experiment, we examined the performance of our classifier when the internal redundancy (represented by the average number of sources per artifact) changes. Similarly as before, we set the number of sources to 300, the average artifacts generated per source is set to 50. We varied the average number of sources per artifact by changing the total number of artifacts generated. We fixed the number of classes at 5 and set the fraction of labeled artifacts to 0.5. The reported results are averaged over 50 experiments and shown in Figure 14. We observed that the classification performance of our classifier improves as the average number of sources per artifact increases. This is intuitive: the more sources per artifact, the more redundancy is available to obtain better results. We also observed that the more specialized the sources are, the less sensitive our classifier will be to the changes in the average number of sources per artifact.

6) Study of Scalability: In the last subsection, we studied the scalability (in terms of execution time) of our classifier over several basic problem dimensions. In the first experiment, we fixed the number of artifacts as 3000 and the average num-

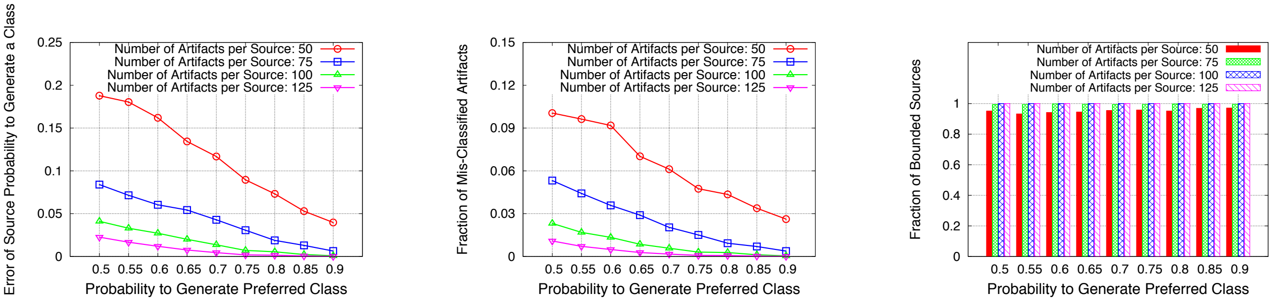


(a) Source Probability Estimation Error

(b) Fraction of Misclassified Artifacts

(c) Fraction of Sources within 90% Confidence Interval

Fig. 11. Changing the number of classes per source

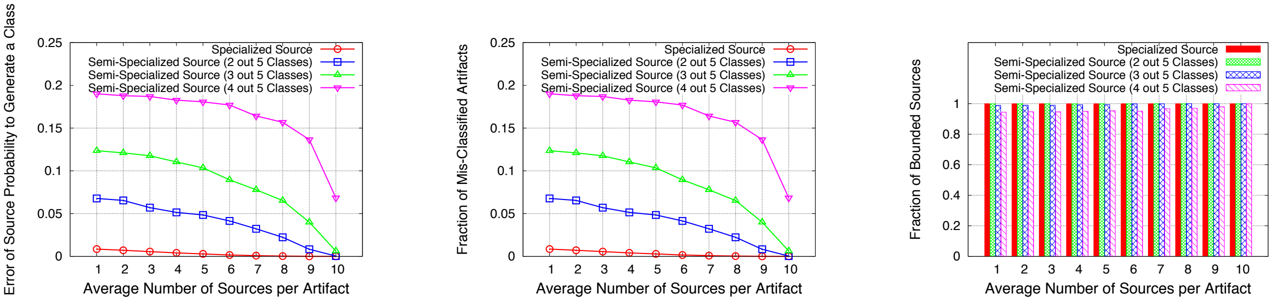


(a) Source Probability Estimation Error

(b) Fraction of Misclassified Artifacts

(c) Fraction of Sources within 90% Confidence Interval

Fig. 12. Changing degree of affinity

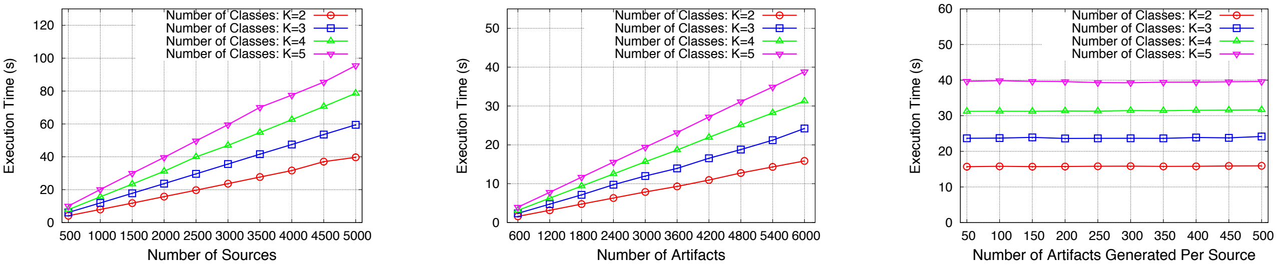


(a) Source Probability Estimation Error

(b) Fraction of Misclassified Artifacts

(c) Fraction of Sources within 90% Confidence Interval

Fig. 13. Changing number of sources per artifact



(a) Changing number of sources

(b) Changing number of artifacts

(c) Changing number artifacts generated per source

Fig. 14. Execution Time (s)

ber of artifacts generated per source as 100. We changed the number of sources from 500 to 5000. The results are averaged 50 experiments and reported in Figure 14(a). We observed the execution time of our classifier is *linearly* proportional to the number of sources in the system. In the second experiment, we fixed the number of sources as 1000 and the average number of artifacts generated per source as 100. We varied the number of total artifacts from 600 to 6000. Results are shown in Figure 14(b). We observed that our classifier also scales *linearly* to the number of artifacts of the problem. In the last experiment, we fixed the number of sources as 1000 and number of artifacts as 3000. We changed the average number of artifacts per source from 50 to 500. Results are shown in Figure 14(c). We noted that the execution time of our classifier is insensitive to the average number of artifacts generated per source.

This concludes our evaluation study. In this section, we evaluated the performance of the proposed PA classifier through two real world applications as well as extensive simulation experiments. The results verified that our PA classifier can significantly improve the classification performance of traditional classifiers by only using the source ID information. The performance of our classifier was shown to be robust and scalable over different problem dimensions. Additionally, it would also be interesting to examine the usage of source ID as a feature in domain classifiers. The authors would like to pursue this direction in the future work.

VII. Limitations and Future Work

This paper presented a general classifier enhancement scheme that uses source IDs to improve classification accuracy. Several simplifying assumptions were made that offer directions for future work.

In this paper, sources are assumed to be independent from each other in the sense that each source has their own independent affinities for generating artifacts of different types. In general, these affinities may be related. For example, if my friends in the social network speak a given language, there is a higher chance that I speak that language as well. This paper does not model such dependencies.

Several solutions have recently been proposed to model source dependencies in various special cases. One possible method is to detect the copy relationship between sources based on historical data [25], [26]. Another possible solution is to study the latent information dissemination graph between sources and understand how information are actually propagated through non-independent sources [27].

Related with the source independence assumption, the input to the proposed classifier in this paper is merely a set of artifacts labeled with source identities. The goal is to examine the performance improvement of the PA classifier by leveraging the provenance information. However, another distinguishing feature of social signals is the underlying social networks that capture relationships between nodes. It would be interesting to investigate the problem of incorporating the social network information (e.g, connections/linkages between sources) to further improve classification accuracy.

Another interesting direction for future work is to consider the Source ID as a feature in the domain classifiers and compare their performance with our PA classifiers. In that case, we might need to change the specific model of each domain classifier to incorporate the Source ID information. However, it would also be interesting to investigate if there will be another general way to consider the provenance information in classification problems without too much modification of the original models.

It is common to observe sources have some expertise in certain knowledge domains. For example, a biologist may generate artifacts mainly about phylogeny of organisms while a musician may generate artifacts regarding music genres. Although we studied the effect of source affinity on classification performance in the evaluation, we do not explicitly take into account prior knowledge on source expertise in our current classifier. It is interesting to extend our model to take into account more information about sources besides their ID. Furthermore, the affinities of sources to generate different artifacts may change in different situations or over time. In such case, we will need more more efficient estimation schemes to dynamically track the changes in the source affinity. We reserve this as a future work direction.

A few techniques have been proposed in fact-finding to consider the hardness of facts [13], which could be generalized and adapted for our scheme. In general, generating certain artifacts might require a lower degree of specialization than others. For example, in an application where artifacts are tweets describing events, and classes of artifacts refer to locations of these described events, many sources may tweet about worldwide events of common interest. In this case, such general-interest tweets give less information about their sources. However, other tweets may be about special locations and represent specialized local knowledge. Such specialized knowledge is a better indicator of the locations or special interests or their sources. Future extensions of the scheme can therefore estimate and take into account, for different classes of artifacts, the difficulty (or degree of specialization needed) to generate artifacts of that class.

VIII. Conclusion

This paper presented a scheme to improve classification accuracy of social signals by exploiting available source IDs. A maximum likelihood estimation model was built to jointly estimate source affinities and artifact classes, to assist classification tasks. An accuracy bound was derived along with the PA classification scheme to establish confidence in analysis results. The new scheme was evaluated through both real-world case studies and extensive simulation experiments. The results show that our scheme significantly improves classification accuracy compared to traditional domain classifiers and correctly computes confidence intervals. The work represents the first attempt at identifying a general methodology for improving performance of arbitrary classification tasks in social network applications.

Acknowledgements

Research reported in this paper was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. 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] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, "On truth discovery in social sensing: A maximum likelihood estimation approach," in *The 11th ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN 12)*, April 2012.

[2] D. Wang, L. M. Kaplan, and T. F. Abdelzaher, "Maximum likelihood analysis of conflicting observations in social sensing," *ACM Transaction on Sensor Networks*, to appear.

[3] D. Wang, L. Kaplan, T. Abdelzaher, and C. C. Aggarwal, "On scalability and robustness limitations of real and asymptotic confidence bounds in social sensing," in *The 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 12)*, June 2012.

[4] D. Wang, L. M. Kaplan, T. F. Abdelzaher, and C. C. Aggarwal, "On credibility estimation tradeoffs in assured social sensing," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 6, pp. 1026–1037, 2013.

[5] J.Han, M.Kamber, and J. Pei, *Data Mining: Concepts and Techniques, Third Edition*. Morgan Kaufman, 2011.

[6] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[7] S. Bhagat, G. Cormode, and S. Muthukrishnan, "Node classification in social networks," *CoRR*, vol. abs/1101.3291, 2011.

[8] S. Tang, J. Yuan, X. Mao, X.-Y. Li, W. Chen, and G. Dai, "Relationship classification in large scale online social networks and its impact on information propagation," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 2291–2299.

[9] A. Vinciarelli, M. Pantic, and H. Bourlard, "Social signal processing: Survey of an emerging domain," 2008.

[10] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[11] J. Pasternack and D. Roth, "Knowing what to believe (when you already know something)," in *International Conference on Computational Linguistics (COLING)*, 2010.

[12] X. Yin, J. Han, and P. S. Yu, "Truth discovery with multiple conflicting information providers on the web," *IEEE Trans. on Knowl. and Data Eng.*, vol. 20, pp. 796–808, June 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1399100.1399392

[13] A. Galland, S. Abiteboul, A. Marian, and P. Senellart, "Corroborating information from disagreeing views," in *WSDM*, 2010, pp. 131–140.

[14] D. R. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., 2011, pp. 1953–1961.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, vol. 39, no. 1, pp. 1–38, 1977.

[16] M. H. C. Law, M. A. T. Figueiredo, and A. K. Jain, "Simultaneous feature selection and clustering using mixture models," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 26, pp. 1154–1166, 2004.

[17] J. V. Graca, L. Inesc-id, K. Ganchev, B. Taskar, J. V. Graa, L. F. Inesc-id, K. Ganchev, and B. Taskar, "Expectation maximization and posterior constraints," in *In Advances in NIPS*, 2007, pp. 569–576.

[18] C. Zhai, "A note on the expectation maximization (em) algorithm," *Department of Computer Scinece, University of Illinois at Urbana Champaign*, 2007.

[19] J. Bilmes, "A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," *Technical Report, University of Berkeley, ICSI-TR-97-021*, 1997.

[20] G. J. McLachlan and T. Krishnan, "The em algorithm and extensions." *John Wiley and Sons, Inc.,*, 1997.

[21] T. Moon, "The expectation-maximization algorithm," *Signal Processing Magazine, IEEE*, vol. 13, no. 6, pp. 47–60, 1996.

[22] J. Gunther, D. Keller, and T. Moon, "A generalized bcjr algorithm and its use in iterative blind channel identification," *Signal Processing Letters, IEEE*, vol. 14, no. 10, pp. 661–664, 2007.

[23] C. F. J. Wu, "On the convergence properties of the EM algorithm," *The Annals of Statistics*, vol. 11, no. 1, pp. 95–103, 1983. [Online]. Available: http://dx.doi.org/10.2307/2240463

[24] S. Tratz, D. Briesch, J. Laoudi, and C. Voss, "Tweet conversation annotation tool with a focus on an arabic dialect, moroccan darija," in *In Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse, Association for Computational Linguistics, Sofia, Bulgaria.*, 2013.

[25] X. Dong, L. Berti-Equille, and D. Srivastava, "Truth discovery and copying detection in a dynamic world," *VLDB*, vol. 2, no. 1, pp. 562–573, 2009. [Online]. Available: http://portal.acm.org/citation.cfm?id=1687627.1687691

[26] X. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava, "Global detection of complex copying relationships between sources," *PVLDB*, vol. 3, no. 1, pp. 1358–1369, 2010.

[27] P. Netrapalli and S. Sanghavi, "Learning the graph of epidemic cascades," in *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '12. New York, NY, USA: ACM, 2012, pp. 211–222. [Online]. Available: http://doi.acm.org/10.1145/2254756.2254783