Latent Credibility Analysis

Jeff Pasternack
Facebook, Inc.
1601 Willow Road
Menlo Park, California 94025
jeffp@fb.com

Dan Roth
University of Illinois, Urbana-Champaign
201 North Goodwin
Champaign, Illinois 61801
danr@illinois.edu

ABSTRACT

A frequent problem when dealing with data gathered from multiple sources on the web (ranging from booksellers to Wikipedia pages to stock analyst predictions) is that these sources *disagree*, and we must decide which of their (often mutually exclusive) claims we should accept. Current state-of-the-art information credibility algorithms known as "fact-finders" are transitive voting systems with rules specifying how votes iteratively flow from sources to claims and then back to sources. While this is quite tractable and often effective, fact-finders also suffer from substantial limitations; in particular, a lack of transparency obfuscates their credibility decisions and makes them difficult to adapt and analyze: knowing the mechanics of how votes are calculated does not readily tell us what those votes *mean*, and finding, for example, that a source has a score of 6 is not informative. We introduce a new approach to information credibility, *Latent Credibility Analysis* (LCA), constructing strongly principled, probabilistic models where the truth of each claim is a latent variable and the credibility of a source is captured by a set of model parameters. This gives LCA models clear semantics and modularity that make extending them to capture additional observed and latent credibility factors straightforward. Experiments over four real-world datasets demonstrate that LCA models can outperform the best fact-finders in both unsupervised and semi-supervised settings.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval—*Information filtering*; I.2.m [**Computing Methodologies**]: Artificial Intelligence

General Terms

Algorithms, Experimentation, Measurement, Reliability

Keywords

Credibility, Graphical Models, Trust, Veracity

1. INTRODUCTION

Conflicts among information sources are commonplace: Twitter users debate the effects of healthcare reform, Wikipedia authors provide differing populations for the same city, online retailers offer discordant descriptions of the same product, financial analysts disagree on the future price of securities, and medical blogs prescribe different courses of treatment. Consequently, we need a means of discerning which of the asserted claims are true, especially on the web, where three of our four experimental datasets (from current, real problems in information credibility) originate. Presently this is addressed by simple or weighted voting or, with more sophisticated fact-finder algorithms (e.g. [4, 18, 14]), transitive voting, but these methods tend to be ad hoc and difficult to analyze and extend. Latent Credibility Analysis is a new method of approaching the credibility problem by instead modeling the joint probability of the sources making claims and the unseen (latent) truth of those claims. Finding the probability that a particular claim is true is then performed via inference in a probabilistic graphical model using one of the many extant exact and approximate inference algorithms. Unlike those of fact-finders, the resulting credibility decisions and the parameters capturing the credibility of the sources are distributions and probabilities with clear semantics: for example, in the SimpleLCA model we reason that a claim is likely to be true because the probability that everyone who asserted it was lying (as given by the Honesty parameters of the sources) is relatively small.

This transparency is important both when we need to explain the model's decisions to users (who might otherwise distrust the system itself) and when we adapt an LCA model to real-world problems; in our experiments, we are able to formulate reasonable priors and anticipate (to a degree) the most appropriate, best performing models by understanding the domain. Such clarity is a common trait of probabilistic models, but a substantial improvement over fact-finders, where the closest analog to priors is typically the number of "votes" each claim is initialized with; further, fact-finders in general have few, if any, other tunable parameters that can be adjusted, and where present (like the Investment fact-finder's "growth rate" value [13]) they tend to be both ad hoc and opaque—it is rarely possible to anticipate what values are suitable for a particular problem before evaluating them on labeled data. LCA models are also much simpler to modify on a more substantial level: there is a straightforward path from a "generative story" about why sources assert the claims that they do to the joint distribution, and augmenting this core (e.g. to incorporate the idea that observed attributes of the sources, like academic degrees, influence their credibility) is as simple as finding a product across several independent components. Even in experiments ig-

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

norant of such factors and using the fact-finders' standard unsupervised setting, LCA models substantially outperform fact-finders in establishing the credibility of city population, book authorship, stock predictions, and predictions of the Supreme Court of the United States. Perhaps surprisingly, this needn't come at an exorbitant cost: two of our models scale linearly, as fact-finders do, and the remaining two, while not linear time, nonetheless proved tractable even over relatively large (tens of thousands of sources and claims) datasets in our experiments.

In the remainder of this paper we first provide a more detailed description of fact-finders. We subsequently discuss the fundamentals of LCA before introducing, in order of increasing sophistication, four specific LCA models: SimpleLCA, GuessLCA, MistakeLCA, and LieLCA, and then explore the performance of these models in comparison to fact-finders in our experiments.

2. BACKGROUND: FACT-FINDERS

A fact-finder takes as its input a list of assertions of the form "source s asserts claim c" and a list of disjoint *mutual exclusion sets* of claims [14]. Exactly one of the claims in each mutual exclusion set is true, and this is what the fact-finder endeavors to identify. This is done via an iterative transitive voting system: starting from some initial belief score in all the claims, the algorithm calculates the trustworthiness of each information source (e.g. a Wikipedia editor, a financial analyst, a website, a classifier, etc.) based on the claims it makes, and then in turn calculates the belief of the claims based on the trustworthiness of the sources asserting it; this process then repeats for a fixed number of iterations or until convergence.

Fact-finders are differentiated by their various update rules, whereby the trustworthiness of sources and belief in claims is calculated. For example, the "Sums" fact-finder is derived from Hubs and Authorities [9], where source trustworthiness can be considered the "hub" score and claim belief the "authority" score; at each iteration i we calculate the trustworthiness of each source as the sum of the belief in its claims, $T^i(s) = \sum_{c:s\to c} B^{i-1}(c)$, and then the belief score of each claim as the sum of the trustworthiness of the sources asserting it, $B^i(c) = \sum_{s:s\to c} T^i(s)$. Of course, fact-finders can be considerably more complex and varied; in the Investment and PooledInvestment [13] algorithms, sources "invest" their credibility in the claims they make, and claim belief is then non-linearly grown and apportioned back to the sources based on the size of their "investment".

Several fact-finders have probabilistic elements. TruthFinder [19] calculates claim belief as $1 - \prod_{s:s\to c} 1 - T(s)$, with the idea that $T(s)$ is the probability that s tells the truth, so the probability that a claim is wrong is the probability that all the (independent) sources are liars. However, these semantics are problematic: the pseudoprobabilities over all the claims in a mutual exclusion set will not sum to 1 and cannot be readily normalized since the trustworthiness of a source is calculated as the arithmetic mean of those claims it makes. [17] explicitly seeks to create a fact-finder with an (approximate) Bayesian justification, but relies on substantial assumptions, the most important being that $P(s \to c|True(c)) \approx P(s \to c)$, i.e. the probability a source asserts a claim is independent of the truth of that claim (which does not hold in practice). [21] is something of an anomaly, as it, like Latent Credibility Analysis, models the credibility problem as a graphical model (a Bayesian network), but specializes in situations where the truth is a collection of entities (e.g. identify all the authors of a book) and the model has the advantage of reasoning about these directly; other approaches (including LCA) instead simply treat these as binary claims (is "John Smith" an author of "Book" or not?). More importantly, the model makes an implicit assumption (as noted by the authors) that each source is predominately honest, which often does not hold in real data (e.g. vandalism in Wikipedia).

Additionally, some fact-finders have incorporated aspects beyond source trustworthiness and claim belief into their update rules. 3-Estimates [4] adds parameters to attempt to capture the "difficulty" of a claim, an idea also present in our LCA models. Fact-finders have also been applied to instances where the claims are not extracted in a prior step but rather snippets of textual "evidence" are effectively clustered using similarity metrics, as applied by the Apollo system to tweets [10] or to news articles by [16]. AccuVote [3] attempts to identify source dependence (one source copying another) to give greater credence to more "independent" sources, an aspect that is important in certain domains (e.g. blog postings, which are routinely derivative) and could be incorporated in future LCA models, although we do not consider it here.

Finally, frameworks have been created capable of extending any fact-finder. [13] applies declarative prior knowledge (in the form of first-order logic) to fact-finders by using linear programming to constrain claim beliefs; in our experiments, we use this method in an extremely simple form to apply supervision to fact-finders (our constraints are of the type "claim c is true"), which are otherwise wholly unsupervised algorithms. For LCA models, declarative constraints may be enforced by one of several methods for constraining the posterior distributions of probabilistic models, such as Posterior Regularization [5] or Constraint Driven Learning [1]. Further, [14] introduces generalized fact-finders, which adapt the bipartite unweighted graphs of standard fact-finders to weighted, k-partite graphs, allowing such factors as source features (e.g. "source s has a doctorate in a relevant field") and uncertainty in information extraction to be incorporated, essentially changing how votes flow throughout the network. LCA models naturally support these forms of prior knowledge and data in a principled way, as we will discuss shortly, and can incorporate many others (such as priors over the honesty of sources and real-valued features) that generalized fact-finders cannot.

3. LATENT CREDIBILITY ANALYSIS

3.1 Fundamentals

A Latent Credibility Analysis model is a probabilistic model where the true claim $\overline{c}$ in each mutual exclusion set of claims m is a (multinomial) latent variable, y_m. An observed assertion is the probability of c as claimed by s, $b_{s,c}$, typically $\{0, 1\}$ (e.g. "John claims Obama was born in Hawaii"), but distributional claims are also possible (e.g. "John is 95% certain Obama was born in Hawaii and 5% certain he was born in Alaska"). Note that $\forall_s,\ \sum_{c\in m} b_{s,c} = 1$. Every source s also has a $[0, \infty)$ confidence in his assertions over the claims in m, $w_{s,m}$, again typically $\{0, 1\}$ (0 if the source makes no assertion about m, 1 if it does), but other values may be used to express degrees of confidence with straightforward seman-

Notation	Description	Examples / Definition
s	An information source	Amazon.com; Dan Rather
c	A claim	President Barack Obama born in 1953
m	A mutually exclusive (ME) set of claims	Claimed Birth Years of Barack Obama
y_m	The true claim in m	President Barack Obama was born in 1961
$b_{s,c}$	The (observed) probability of c asserted by s	0; 1; 0.7
$w_{s,m}$	$[0, \infty)$ confidence of s in the distribution asserted over all $c \in m$	0; 1; 4.5
H_s	The probability s makes an honest, accurate assertion	0.4; 0.9
$D_{g/m/s}$	The probability s knows y_m (global, per-ME set, or per-source)	0.3; 0.7
S	Set of all sources s	$= \{s\}$
C	Set of all claims c	$= \{c\}$
M	Set of all mutual exclusion sets m	$= \{m\}$
B	$\|S\| \times \|C\|$ matrix of all observed assertions b	$= \{b_{s,m}\}$
W	$\|S\| \times \|M\|$ matrix of all assertion confidences w	$= \{m\}$
Y	Set of all true claims	$= \{y_m : m \in M\}$
Y_U	Set of all latent true claims	$\subseteq Y$
Y_L	Set of all observed true claims (labels)	$\subset Y$
X	Set of all observations (including B)	$= B \cup \{$ all other features $\}$
θ	Set of all latent model parameters	e.g. $\{H_s : s \in S\} \cup \{D_m : m \in M\}$

Table 1: LCA Notation

tics: as can be seen from the joint distributions of our LCA models, a $w_{s,m}$ of 0.5 causes assertions made by s about claims in m to affect the log-likelihood only half as much as sources with $w_{s,m} = 1$, and $w_{s,m} = 2$ is equivalent to making the same assertions twice. This can be useful if, for example, a source expresses abundant or reduced confidence in his assertion, e.g. "John is 50% confident that Obama was born in Hawaii with 95% probability...", comparable in function and purpose to belief and plausibility in Dempster-Shafer theory [20, 15] and uncertainty in subjective logic [8, 7].

Since we are not interested in modeling why a source decides to make an assertion about the claims in a mutual exclusion set (and with what confidence), the confidence matrix $W = \{w_{s,m}\}$ is taken as a given constant rather than an observation. Our observations are the assertion matrix $B = \{b_{s,c}\}$, together with whatever observed features (such as attributes of the sources) are relevant to the particular model; we will collectively refer to these observed variables as X. Similarly, we will refer to our latent variables as $Y = \{y_m\}$, and the model parameters (in the models we describe later these include the honesty of each source and the "difficulty factor" of identifying the true claim) as θ. Finally, when we write the joint probabilities, *we assume all mutual exclusion sets contain at least two claims*; this is a notational convenience, since any uncontested claim must be true (there is no alternative) and the probability of a source asserting it is thus 1 and it does not affect the joint probability.

As an example, consider a problem with two mutual exclusion sets, m_p ="Obama's Birthplace" and m_d ="Obama's Birthdate", where we observe a source s_j = "John" make a single assertion c_h = "Obama was born in Hawaii". Then $b_{s_j,c_h} = 1$, $\forall_{c \in m_p \setminus c_h} b_{s_j,c} = 0$, $w_{s_j,m_p} = 1$, and $w_{s_j,m_d} = 0$ (rendering the values of $\{b_{s_j,c} : c \in m_d\}$ irrelevant). Latent variables y_{m_p} and y_{m_d} are Obama's true birthplace and birthdate, respectively, so y_{m_p} = "Hawaii" and y_{m_d} = "August 4th, 1961".

3.2 Inference

Information credibility problems can be classed as unsupervised or semi-supervised; in the unsupervised case, we are only given observations X and none of the y_m are known, so $Y_U = Y$ and $Y_L = \emptyset$ (Y_U and Y_L are the sets of unlabeled [latent] and labeled [observed] true claims, respectively). Alternatively, when semi-supervised, we know the true claims in some mutual exclusion sets, $Y_L \subset Y$, already and only need to determine the remaining $Y_U = Y \setminus Y_L$. In both cases, our goal is to infer:

$$P(Y_U|X, Y_L) = \frac{\int_\theta P(Y_U, Y_L, X|\theta)P(\theta)}{\sum_{Y_U} \int_\theta P(Y_U, Y_L, X|\theta)P(\theta)}$$

This is the distribution over the possible true claims for each mutual exclusion set where the true claim is not already known, given the observations and true claims already identified. In our experiments we solve this approximately, by using EM [2] to find the maximum a posteriori (MAP) point estimate of the parameters, $\theta^* = \text{argmax}_\theta P(X|\theta)P(\theta)$, and then simply calculating:

$$P(Y_U|X, Y_L, \theta^*) = \frac{P(Y_U, X, Y_L|\theta^*)}{\sum_{Y_U} P(Y_U, X, Y_L|\theta^*)}$$

The expectation and maximization update rules used to find the maximum a posteriori point estimate θ^* are:

Expectation – Step :

$$\forall_m \ : \ P(y_m = c|X, \theta^t) = \frac{P(y_m = c, X|\theta^t)}{\sum_{v \in m} P(y_m = v, X|\theta^t)}$$

Maximization – Step :

$$\theta^{t+1} = \underset{\theta}{\text{argmax}}\, \mathbb{E}_{Y|X,\theta^t}[\log(P(X, Y|\theta)P(\theta))]$$

In LCA models, the E-step is always easy, since the y_m values are independent given the observations X and the parameters θ^t at iteration t. The M-step can be more difficult:

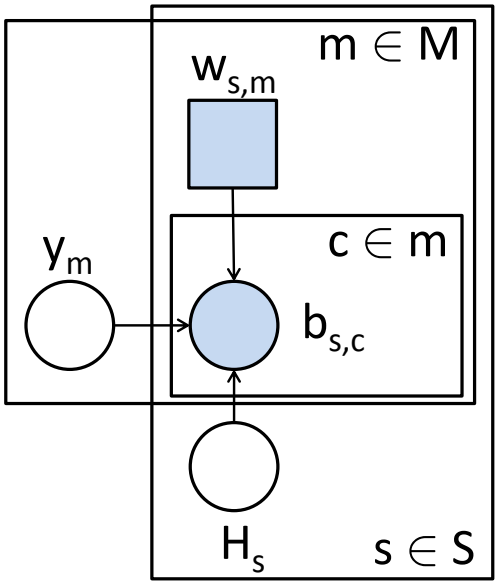

Figure 1: A plate diagram of a basic SimpleLCA model with observed assertions as the sole features ($X = B$).

in SimpleLCA, θ^{t+1} can be calculated in closed form provided that $P(\theta)$ is uniform; otherwise, gradient ascent must be used. Where this can be done parameter-by-parameter, the time required for the M-step scales linearly in the number of parameters; in MistakeLCA and LieLCA, joint gradient ascent requires a number of steps increasing linearly in the number of dimensions [12] (since the Lipschitz constant and squared diameter both increase linearly) while the cost to compute the gradient and the function value also increase linearly (provided the number of assertions per source and claims per mutual exclusion set remains constant), yielding $O(|\theta|^2)$ complexity. However, even on our largest experiments, MistakeLCA and LieLCA took no more than $\sim$ 200 times as long as SimpleLCA and GuessLCA, far less than suggested by this worst case quadratic bound. Exact runtimes varied, but for concreteness LieLCA took approximately 20 minutes on the population dataset, 30 minutes on the stock dataset (per time interval), and from 25-80 minutes on the books dataset (single-threaded on a 3GHz Core 2 Duo E8400); by comparison, GuessLCA was 40 seconds, one minute, and 3-4 minutes, respectively.

3.3 SimpleLCA

SimpleLCA, as with all our models, is a joint distribution that reflects a "story" of how sources decide which claims to assert. For both this and subsequent LCA models, we assume that each $b_{s,c} \in \{0, 1\}$ and each $w_{s,m} \in \{0, 1\}$; this matches our experimental domains (where sources assert a single claim in a mutual exclusion set with full certainty) and simplifies the equations for the joints by avoiding a cumbersome normalization factor. If these assumptions are relaxed, the joint "distributions" as written will no longer be distributions and must be normalized.

In SimpleLCA, each source s has a probability of being honest, H_s. A source then decides to assert the true claim $\overline{c}$ in mutual exclusion set m with probability H_s; otherwise, it chooses uniformly at random from the other claims in m with probability $\frac{1-H_s}{|m|-1}$. From this intuitive idea, we can immediately derive a joint distribution over y_m and X:

$$\begin{aligned} &P(y_m, X|H_s) \\ &\quad = P(y_m)\left((H_s)^{b_{s,y_m}} \prod_{c \in m \setminus y_m} \left(\frac{1-H_s}{|m|-1}\right)^{b_{s,c}}\right)^{w_{s,m}} \\ &\quad = P(y_m)\left((H_s)^{b_{s,y_m}} \left(\frac{1-H_s}{|m|-1}\right)^{(1-b_{s,y_m})}\right)^{w_{s,m}} \end{aligned}$$

Here, $P(y_m)$ is our prior probability of y_m being the true claim in m, and $w_{s,m}$ will be 1 if the source asserts (with full certainty) a claim in m, or 0 if the source says nothing about m. In the second equation we have simplified the expression by noting that $\sum_{c \in m} b_{s,c} = 1$, so $\sum_{c \in m \setminus y_m} b_{s,c} = 1 - b_{s,y_m}$.

Observing that all sources make their assertions independently and taking $\theta = \{H_s\}$ we can write the full joint as:

$$P(Y, X|\theta) = \prod_m P(y_m) \prod_s \left((H_s)^{b_{s,y_m}} \left(\frac{1-H_s}{|m|-1}\right)^{(1-b_{s,y_m})}\right)^{w_{s,m}}$$

The expected log-likelihood maximized in the M-step is then $\mathbb{E}_{Y|X,\theta^t}[\log(P(X, Y|\theta)P(\theta))] =$

$$\begin{aligned} &\log(P(\theta)) + \sum_Y P(Y|X, \theta^t) \log\left(\prod_m P(y_m)\right. \\ &\quad \left. \cdot \prod_s \left((H_s)^{b_{s,y_m}} \left(\frac{1-H_s}{|m|-1}\right)^{(1-b_{s,y_m})}\right)^{w_{s,m}}\right) \end{aligned}$$

$$\begin{aligned} &= \log(P(\theta)) + \sum_m \sum_{y_m} P(y_m|X, \theta^t)\left(\log(P(y_m))\right. \\ &\quad \left. + \sum_s w_{s,m}\left(b_{s,y_m}\log(H_s) + (1-b_{s,y_m})\log\left(\frac{1-H_s}{|m|-1}\right)\right)\right) \end{aligned}$$

Finding the derivative with respect to each $H_s \in \theta$,

$$\begin{aligned} &\frac{\delta}{\delta H_s}\mathbb{E}_{Y|X,\theta^t}[\log(P(X, Y|\theta)P(\theta))] = \\ &\quad \frac{\delta P(H_s)}{\delta H_s} P(H_s)^{-1} \\ &\qquad + \frac{\sum_m \sum_{y_m} P(y_m|X, \theta^t) w_{s,m}(b_{s,y_m} - H_s)}{H_s - (H_s)^2} \end{aligned}$$

Now we can maximize each H_s independently in our M-step using gradient ascent to find the new, maximizing θ^{t+1}. However, when the priors $P(H_s)$ are uniform (so $\frac{\delta P(H_s)}{\delta H_s} = 0$), the gradient simplifies, allowing us to set it to 0 and solve the resulting equation explicitly for the new maximizing value of H_s at the stationary point:

$$H_s = \frac{\sum_m \sum_{y_m} P(y_m|X, \theta^t) w_{s,m} b_{s,y_m}}{\sum_m w_{s,m}}$$

As we would intuitively expect, we thus estimate the honesty of a source, that is, the probability that it provides the true claim, as essentially the expected proportion of true claims made by the source given our current parameters.

This closed form update rule also means that SimpleLCA with uniform honesty priors is as fast as fact-finders in practice, making it extremely scalable. When alternative priors are used, gradient ascent requires about twice as much time per EM iteration, but even on our largest datasets this was a matter of seconds.

3.4 GuessLCA

SimpleLCA is indeed quite simple. But it's also clear that, for sources, identifying the truth in some mutual exclusion sets is much harder than in others; for example, a source who merely guessed randomly would be assigned an honesty of 0.5 by SimpleLCA if it only made claims in mutual exclusion sets of size 2, and 0.25 if size 4.

In GuessLCA, a source has a probability of knowing and telling the truth, H_s. Thus, with probability H_s, it asserts the true claim. However, with probability $1 - H_s$, it guesses claim c with probability $P_g(c|s)$ (where $\sum_{c \in m} P_g(c|s) = 1$). This gives us the joint probability:

$$P(X, Y|\theta) = \prod_m P(y_m) \prod_s (H_s + (1 - H_s)P_g(y_m|s))^{b_{s,y_m} w_{s,m}} \prod_{c \in m \setminus y_m} ((1 - H_s)P_g(c|s))^{b_{s,c} w_{s,m}}$$

This joint can be easily understood by considering the marginal case for each $m \in M$; the probability that the source asserts the true claim ($b_{s,y_m} = 1$) is then just $H_s + (1 - H_s)P_g(y_m|s)$, the probability of knowing the truth plus the chance of not knowing the truth and (fortunately) guessing it; $\sum_{c \in m} b_{s,c} = 1 \Rightarrow \forall_{c \neq y_m} b_{s,c} = 0$, so the product $\prod_{c \in m \setminus y_m} (\ldots)^{b_{s,c}} = 1$ is moot. Conversely, the probability of asserting an untrue claim ($b_{s,c \neq y_m} = 1$) can be similarly found as the probability of not knowing the truth and guessing c, $(1 - H_s)P_g(c|s)$.

Omitting the intermediate steps for brevity, we find that the gradient of the expected log-likelihood with respect to H_s simplifies to

$$\frac{\delta}{\delta H_s} \mathbb{E}_{Y|X,\theta^t}[\log(P(X, Y|\theta)P(\theta))] = \frac{\delta P(H_s)}{\delta H_s} P(H_s)^{-1} + \sum_m \sum_{y_m} P(y_m|X, \theta^t) w_{s,m} \cdot \left(\frac{b_{s,y_m}}{H_s + \frac{P_g(y_m|s)}{1 - P_g(y_m|s)}} + \frac{b_{s,y_m} - 1}{1 - H_s} \right)$$

Like SimpleLCA, the gradient with respect to each H_s is independent of the other parameters $\theta \setminus H_s$, allowing us to maximize the expected log-likelihood in the M-step using gradient ascent parameter-by-parameter, which is very fast in practice. The guess distribution $P_g(c|s)$ is provided to the model as a prior; we could, for example, set $P_g(c|s)$ to the distribution of sources asserting the claims in m under the assumption that a guessing source chooses randomly according to the distribution of "votes" it observes at the time. This mitigates sources becoming trusted by asserting obvious or well-known claims: the assessed probability of guessing these will then be high (because a large majority of sources already assert them, and we assume that guessers tend to go with the crowd), so the model is free to set H_s low as the observation can be effectively explained away by $(1 - H_s)P_g(y_m|s)$; conversely, a source asserting a true claim with a low probability of being guessed will be attributed to a high H_s. GuessLCA thus rewards getting hard claims right and penalizes getting easy claims wrong.

GuessLCA does require that this "difficulty" information be provided a priori rather than learned by the model, and while in most domains the distribution of guesses is easy to approximate (e.g. if the sources tend to guess with the crowd, probably the most prevalent behavior in practice, we can use the distribution of the number of assertions made by other sources for each alternative within the mutual exclusion set, and if the sources are believed to guess randomly we use a uniform prior over the possibilities) this cannot capture the latent difficulty implied by, for example, the disagreement of two highly honest sources (since honesty itself is latent). More significantly, the model assumes that no source will do worse than guessing—even if $H_s = 0$, a source still has a $P_g(c|s)$ probability of guessing the correct claim c. This assumption is violated when sources are systematically wrong. This may be due to intentional deception, or, more commonly, a recurring mistake: for example, there are multiple ways of defining the population of a city (metro area, city limits, etc.) and some Wikipedia editors consistently use definitions that disagree with the "truth" (census data).

3.5 MistakeLCA

To overcome these problems, MistakeLCA models difficulty explicitly, as the probability of an honest source making a mistake. For a source to assert the true claim it must both intend to tell the truth with probability H_s *and* must know what the truth is with probability D. D may be global (in which case all sources have probability D_g of knowing the truth across all mutual exclusion sets) or tied to each mutual exclusion set (in which case sources have probability D_m of knowing the truth in a particular mutual exclusion set); this results in two variants of the model, which we will refer to as MistakeLCA$_g$ and MistakeLCA$_m$. A source thus asserts the true claim $\overline{c}$ with probability $H_s D$, but otherwise, with probability $1 - H_s D$, chooses another claim $c \in m \setminus \overline{c}$ according to $P_e(c|\overline{c}, s)$. Recall that, in GuessLCA, our guessing probability P_g was not conditioned on the true claim, but P_e specifies the distribution of mistakes a source will make given that $\overline{c}$ is true, with $P_e(\overline{c}|\overline{c}, s) = 0$. Like P_g, P_e is provided as a prior, but conditioning on the true claim means that it can also encode very useful information about similar or easily confused claims; for example, if there are three claims about a person's age, 35, 45, and 46, $P_e(45|46, s)$ and $P_e(46|45, s)$ would both be high.

The joint probability is given by:

$$P(X, Y|\theta) = \prod_m P(y_m) \prod_s (H_s D)^{b_{s,y_m} w_{s,m}} \prod_{c \in m \setminus y_m} (P_e(c|y_m, s)(1 - H_s D))^{b_{s,c} w_{s,m}}$$

The gradients of the expected log-likelihood are given by:

$$\frac{\delta(\ldots)}{\delta H_s} = \frac{\delta P(H_s)}{\delta H_s} P(H_s)^{-1} + \sum_m \sum_{y_m} P(y_m|X,\theta^t) w_{s,m} \left(\frac{b_{s,y_m} - D_m H_s}{H_s - D_m H_s^2} \right)$$

$$\frac{\delta(\ldots)}{\delta D_m} = \frac{\delta P(D_m)}{\delta D_m} P(D_m)^{-1} + \sum_s \sum_{y_m} P(y_m|X,\theta^t) w_{s,m} \left(\frac{b_{s,y_m} - D_m H_s}{D_m - D_m^2 H_s} \right)$$

The gradient for D_g is identical, except that we sum over all mutual exclusion sets as well as all sources. Since all H_s are linked by D, we must optimize all parameters jointly in the M-step.

3.6 LieLCA

MistakeLCA makes no distinction between intentional lies caused by a lack of honesty and "honest mistakes" that occur with probability $(1-D)$; we can imagine that the former case is governed by a distribution over possible lies, whereas the latter results in guessing. In LieLCA, a source asserts the true claim $\overline{c}$ if it it is both honest and knows the answer (with probability $H_s D$). A dishonest source who knows the truth, however, chooses a lie c with probability $(1-H)DP_l(c|\overline{c},s)$, where P_l is the distribution over possible lies given the truth ($P_l(\overline{c}|\overline{c},s) = 0$). Finally, any source who does not know the truth guesses a claim c with probability $(1-D)P_g(c|s)$. The D parameters may be per-source, per-mutual exclusion set, or global, resulting in LieLCA$_s$, LieLCA$_m$, and LieLCA$_g$ variants. The joint probability is thus:

$$P(X,Y|\theta) = \prod_m P(y_m) \prod_s (H_s D + (1-D)P_g(y_m|s))^{b_{s,y_m} w_{s,m}} \prod_{c \in m \setminus y_m} ((1-H_s)DP_l(c|y_m,s) + (1-D)P_g(c|s))^{b_{s,c} w_{s,m}}$$

The gradients of the expected log-likelihood with respect to H_s and D can be found as:

$$\frac{\delta(\ldots)}{\delta H_s} = \frac{\delta P(H_s)}{\delta H_s} P(H_s)^{-1} + \sum_m \sum_{y_m} P(y_m|X,\theta^t) w_{s,m} \left(\frac{b_{s,y_m} D}{(1-D)P_g(y_m|s) + DH_s} - \sum_{c \in m \setminus y_m} \frac{b_{s,y_m} DP_l(c|y_m,s)}{(1-D)P_g(c|s) + D(1-H_s)P_l(c|y_m,s)} \right)$$

$$\frac{\delta(\ldots)}{\delta D_g} = \frac{\delta P(D_g)}{\delta D_g} P(D_g)^{-1} + \sum_{m,s} \sum_{y_m} P(y_m|X,\theta^t) w_{s,m} \left(\frac{b_{s,y_m}(H_s - P_g(y_m|s))}{H_s D_g + (1-D_g)P_g(y_m|s)} + \sum_{c \in m \setminus y_m} \frac{b_{s,c}((1-H_s)P_l(c|y_m,s) - P_g(c|s))}{(1-D_g)P_g(c|s) + D_g(1-H_s)P_l(c|y_m,s)} \right)$$

Again, the gradients for D_m and D_s are identical, except $\sum_{m,s}$ is replaced by $\sum_s$ and $\sum_m$, respectively. It is interesting to note that LieLCA$_s$ is a special case since each pair of (H_s, D_s) parameters may be optimized independently of the others, with the same linearly scaling complexity as SimpleLCA and GuessLCA; otherwise, like MistakeLCA, the parameters must be optimized jointly.

It is important to note that we are abusing language somewhat here; in LiarLCA, a "lie" is an intentional, incorrect assertion by a source who knows the truth, but it need not imply malice or an intent to deceive. A Wikipedia editor who (perhaps out of ignorance) accurately lists the population of cities by their greater metro area rather than by their city limits when the latter is held to be the true measure would not normally be considered a liar, even though the model considers their assertions to be "lies" (and in this particular case those "lies" may be quite informative since we know they will be drawn from values strictly greater than the true population such that $P_l(c|\overline{c},s) > 0$ iff $c \geq \overline{c}$).

3.7 Discussion

3.7.1 Model Complexity and Semantics

Given that we have presented a series of increasingly complex models it might be tempting to think of these hierarchically along the lines of SimpleLCA $\subset$ GuessLCA $\subset$ MistakeLCA $\subset$ LieLCA. However, this is incorrect: it is easy to see that there are some worlds that SimpleLCA can model (a source with an honesty of 0 who always asserts the wrong claim) that, for example, GuessLCA cannot (at worst a source will still sometimes guess the truth). We can similarly observe that the H_s parameters have subtly different meanings in each model: in SimpleLCA, it is simply the probability that a source asserts the correct claim; in GuessLCA, it is the probability that it both knows and asserts it; and in MistakeLCA and LieLCA, it is the probability the source *intends* to tell the truth. Such distinctions are of practical importance: because each model tells a different story with different semantics, we should not expect, for instance, that the more sophisticated LieLCA will necessarily outperform the SimpleLCA model given sufficient data (as we might if SimpleLCA were indeed subsumed by LieLCA); rather, we expect that relative performance will depend on which model more closely reflects the actual behavior of sources within a particular domain. That said, our experiments showed that, indeed, some models appear to be more plausible than others, and the more complex models are vulnerable to overfitting: in particular, GuessLCA performs substantially better than SimpleLCA overall and is competitive with MistakeLCA and LieLCA, especially where these models overfit (e.g. on the stocks dataset).

3.7.2 Extensions

A key benefit of LCA is its flexibility and transparency relative to fact-finders. Bayesian priors over the parameters, claims, and other phenomena (such as the mistake distribution, P_e) provide a straightforward way of encoding domain knowledge, but many extensions are also possible.

The modularity of LCA can be illustrated by an example: consider a case where we have features X_f (such as the quality of a source's website, his academic degrees, years of experience, etc.) associated with the credibility of our sources. By assuming that these features are independent from the

sources' assertions given their credibility, we can create a new model by simply concatenating two joint distributions: $P(X, Y|\theta) = P_{\text{LCA}}(X_b, Y|\theta)P_f(X_f|\theta)$, where $P_{\text{LCA}}(X_b, Y|\theta)$ is an LCA model over observed assertions X_b and $P_f(X_f|\theta)$ is the probability of observing features X_f given the credibility of the sources (captured by parameters θ).

Additionally, LCA models (and fact-finders) will normally only give credibility to claims that are known to exist and asserted by at least one source (an unknown alternative obviously cannot be explicitly considered in the set of possibilities m, and the models infer a distribution over the possible values of $y_m \in m$). However, we can easily create a new "none of the above" claim u and assign it a prior probability $P(u)$; believing one of the known, asserted claims will then depend on the evidence outweighing our prior inclination towards doubt.

4. EXPERIMENTS

We evaluate our models on two unsupervised datasets, book authorship [19] and city populations [13], and two semi-supervised datasets, stock predictions and U.S. Supreme Court decision predictions[1]. Our evaluation compares our four basic LCA models with several top-performing fact-finders found in the literature: TruthFinder [19], Investment, PooledInvestment, and Average-Log [13], Sums [9], 3-Estimates [4], as well as simple voting (choose the claim with the most sources asserting it). For Investment and PooledInvestment we used the same values for g as [13], 1.2 and 1.4, respectively. We run both the fact-finders and EM (for LCA) until convergence (within 50 iterations in our experiments). Additionally, we supplement our real-world experiments with synthetic data from sampled from SimpleLCA joint distributions to more carefully analyze the relative performance of the LCA models in a controlled context.

4.1 Books

The books dataset [19] is a collection of 14,287 claims of the authorship of various books by 894 websites, with an evaluation set of 605 true claims collected by examining the books' covers. We used uniform priors for the parameters $P(\theta)$. For for the claim priors $P(c)$ and guess priors $P_g(c|s)$ we used "voted" priors corresponding to the distribution of sources asserting each claim relative to the number of sources asserting any claim within the mutual exclusion set: $\frac{|\{t:w_{t,m}=b_{t,c}=1\}|}{\sum_{v\in m}|\{t:w_{t,m}=b_{t,v}=1\}|}$. Finally, the mistake and lie priors $P_e(c|\overline{c}, s)$ were also "voted", computed as $P_l(c|\overline{c}, s) = \frac{|\{t:w_{t,m}=b_{t,c}=1\}|}{\sum_{v\in m\setminus\overline{c}}|\{t:w_{t,m}=b_{t,v}=1\}|}$ [for $c \neq \overline{c}$]; this is the proportion of sources asserting c relative to the total number of sources asserting any claim in m other than $\overline{c}$. For simplicity, the distributions are the same for all sources s. For LieLCA$_s$, LieLCA$_m$, and MistakeLCA$_m$, the D_s or D_m parameters in the model are much more variable than a single global D_g (which tends to be high), resulting in greater emphasis on the voted P_e priors and making voted claim priors $P(c)$ effectively redundant; to correct this, we instead use uniform claim priors on these models.

The results are shown in Table 2; we calculate confidence intervals with the simplifying assumption that the prediction over each mutual exclusion set is independent from the others. The only fact-finder to do better than *any* of the LCA models is PooledInvestment, still more than 3% below LieLCA$_s$. The LieLCA$_s$ generative story fits especially well with what we know about online booksellers a priori: some sources will consistently corrupt, abbreviate or omit authors names (in other words, they consistently "lie" with a low H_s), while others "guess" by copying prevailing sources since they tend not to research the information themselves (low D_s).

[1]The Supreme Court, city population, and book authorship datasets are available at http://lotho.cs.illinois.edu/data/ Unfortunately, we are unable to release the stock predictions data due to licensing restrictions.

4.2 Population

The population dataset [13] contains 44,761 claims about the population of a city in a specific year made by 171,171 Wikipedia editors in infoboxes, with an evaluation set of 274 true claims identified from U.S. census data. Our evaluation set is marginally smaller than [13] because when an editor made multiple claims about the population of a city in the same year, we kept only the most recent edit and discarded the rest; this resulted in some true claims becoming uncontested and thus eliminated from the evaluation set. Our priors remained the same as before, except that the claim priors followed the distribution of the number of revisions a claim was present in, rather than the number of sources asserting it, as per [13]. Additionally, we noticed that some models could achieve better results if we knew exactly when to stop them prior to convergence (which is not possible given the unsupervised setting); Investment is the most extreme example of this, as at 20 iterations its accuracy is 86.86%, but it ultimately converges to 75.55%.

There is a wide variance in the the cities in this dataset; some, like Ventura, California are relatively contentious (49 edits asserted a population of 105,000 in 2006, while 68 asserted 106,744), while in others things are more lopsided (in Springfield the split was 202 edits vs. 10). As a consequence, some cities can be considered much "harder" than others, since an overwhelming majority for one option over the others means both that the answer is well-known and that an editor needs only follow the crowd to identify it. Given this, we would expect those models that are capable of capturing this variable difficulty to perform the best, and this matches our experiments exactly: GuessLCA (which attributes greater honesty [H_s] to sources that assert true but hard-to-guess claims and less to those that assert false, easy-to-guess claims) and LieLCA$_m$ and MistakeLCA$_m$ (which model the variable difficulty of each city directly with D_m parameters) are the best performing among the LCA models. TruthFinder also does quite well, but the opaque nature of fact-finders precludes an explanation why, or a prediction of the domains where it might similarly perform well in the future. LieLCA$_m$'s top performance, however, is a result of having both D_m parameters to model latent difficulty (e.g. as demonstrated by incorrect assertions by highly honest sources) and guessing priors to incorporate the more obvious situations of lopsided and even votes where the difficulty is apparent even without having an estimate of the honesty of the sources involved.

4.3 Predicting Stock Returns

We took the set of stocks that were in the S&P 500 Index on January 1st, 2000 (the index changes composition over time) and followed them through February 1st, 2012. Our results average predictive accuracy across 10 dates, at July

Model	Books Unsupervised	Populations Unsupervised	Stocks Semi-Supervised	Supreme Court Semi-Supervised
Voting	84.95 ± 2.85	79.93 ± 4.74	47.14 ± 4.13	54.72 ± 13.40
Sums	82.87 ± 3.00	82.12 ± 4.54	48.93 ± 4.14	56.60 ± 13.34
3-Estimates	85.12 ± 2.84	74.45 ± 5.16	47.14 ± 4.13	52.83 ± 13.44
TruthFinder	86.16 ± 2.75	85.04 ± 4.22	47.14 ± 4.13	58.49 ± 13.27
Average-Log	85.47 ± 2.81	81.02 ± 4.64	46.61 ± 4.13	52.83 ± 13.44
Investment	80.10 ± 3.18	75.55 ± 5.09	51.61 ± 4.14	75.47 ± 11.58
PooledInvestment	87.72 ± 2.62	79.93 ± 4.74	48.93 ± 4.14	77.36 ± 11.27
SimpleLCA	86.51 ± 2.72	82.48 ± 4.50	56.96 ± 4.10	79.25 ± 10.92
GuessLCA	89.10 ± 2.48	83.58 ± 4.39	56.25 ± 4.11	**88.68** ± 8.53
MistakeLCA$_g$	86.33 ± 2.74	82.12 ± 4.54	55.54 ± 4.12	N/A
MistakeLCA$_m$	88.58 ± 2.53	**86.13** ± 4.09	50.89 ± 4.14	N/A
LieLCA$_g$	89.62 ± 2.43	81.39 ± 4.61	**57.86** ± 4.09	N/A
LieLCA$_m$	87.89 ± 2.60	83.94 ± 4.35	51.61 ± 4.14	N/A
LieLCA$_s$	**90.83** ± 2.30	82.85 ± 4.46	53.39 ± 4.13	N/A

Table 2: Experimental Results (N/A: Not Available).
Values are percent accuracy (proportion of true claims correctly identified) and 95% confidence interval. The best LCA models outperform the best fact-finders with statistical significance in the Books, Stocks and Supreme Court datasets.

1st, 2011 and every two weeks thereafter. We pretend that each of these dates is the present time and interpret stock analysts' buy or sell predictions as claims about whether each stock will yield a return higher or lower than the baseline S&P 500 return over the next 60 days. For example, when we pretend that the date is July 1st, 2011 and are considering Microsoft stock we know the buy or sell recommendations analysts have made over the previous two weeks (in late June), and the latent truth we seek to identify is, of course, whether or not the stock will actually outperform the S&P 500 over the next 60 days. As a technical detail, stocks are assumed to be bought piecemeal over a week, starting on the subsequent day, and then sold piecemeal over a week, starting 60 days later (this reduces the day-to-day price variance). At each of these dates, we also know which recommendations analysts made more 60 days ago were proven true, and this observed truth of whether each stock went up or down is our labeled data. Similarly, the remainder of the predictions (those recommendations made in the last 60 days) are effectively unlabeled data, since we do not know if they will be proven true yet. In total, there are approximately 4K distinct analysts and 80K distinct stock predictions, and our evaluation set consists of 560 true claims about stocks where analysts disagreed.

One thing we can quickly observe is that analysts are, in fact, usually wrong, as reflected by the 47.14% accuracy of voting. We therefore used uniform claim priors, which are a better alternative to the voted priors of our previous experiments; all other priors remain the same. Given the difficulty of the problem (as the oft-cited efficient market hypothesis that consistent risk-adjusted returns relative to the market are impossible would suggest [11]) we would expect no analyst to be especially good (otherwise they would presumably be running a hedge fund) nor any stock to be especially easy to predict; modeling these features, then, would offer little benefit but substantial risk of overfitting, as we observe in LieLCA$_m$, MistakeLCA$_m$, and LieLCA$_s$, the three lowest-performing LCA models. Conversely, LieLCA$_g$, balancing the overall difficulty of stock prediction with each source's ability (captured by H_s), does the best (D_g essentially serves as a latent, universal cap on how accurate any analyst can be at the task). Amusingly, the (aptly-named) Investment is the only fact-finder to do better than 50%, although it surpasses only one LCA model (MistakeLCA$_m$).

Given the practical importance of this domain, a natural question to ask is if these models would work in practice as an investment strategy, given the $\sim$ 58% accuracy of LieLCA$_g$. It is important to observe, however, that we considered only binary outperform and underperform labels and, critically, not how much would have been gained (or lost) on each stock; overall excess return relative to the market as a whole is likely to be minor. Furthermore, since the market changes over time, there is no guarantee that a strategy that works on historical data would continue to work in the future, nor can we easily quantify this risk (and unexpected, unlikely events can collectively pose a major hazard to any strategy, e.g. the collapse of Long-Term Capital Management [6]).

4.4 Predicting Supreme Court Decisions

Finally, we considered the FantasySCOTUS project; here, 1138 people (largely law students) have made predictions about the outcome of 53 U.S. Supreme Court cases that have already been decided, and 24 that have not been. Using the same priors as the Books experiment (based on voting), we evaluated with 10-fold cross-validation. Within each fold, Investment, PooledInvestment, SimpleLCA and GuessLCA were tuned by nested 4-fold cross-validation. For Investment and PooledInvestment, the growth parameter (from 1 to 2 in increments of .1), was tuned, whereas for SimpleLCA and GuessLCA the parameter priors $P(H_s)$ were tuned over sets of 10 possible Beta distributions. Since the votes for most cases are nearly tied, we concluded that most sources did little better than guessing, and selected Beta distributions biased toward 0 for GuessLCA (such that the prior on the probability of doing better than guessing is low), and biased towards 1/2 for SimpleLCA (such that the prior probability of asserting the truth is near random). The other fact-finders were not tuned because they lacked tunable parameters; LieLCA and MistakeLCA results are omitted because

the experiments were not feasible; 10-fold cross-validation with 4-fold nested cross-validated tuning across 10 possible distributions of the priors of $P(H_s)$ and $P(D)$ is 4000 times as expensive as a normal run (and running a greatly reduced cross-validation regimen with just a few alternative priors for each parameter would underestimate performance relative to our other LCA results). This is a tradeoff for the greater sophistication of the LieLCA and MistakeLCA models: not only are there an additional set of parameters (the D's) to select priors for, the M-step requires a substantially more expensive optimization (up to about 200 times as expensive as that for SimpleLCA or GuessLCA as previously discussed; a single, normal run of LieLCA on this dataset takes 20-30 minutes). However, we note that this cross-validated tuning is parallelizable, and a real-world implementation could handle the task by splitting it over a cluster of machines.

4.5 Synthetic Results and Analysis

In our experimental results, our understanding of the domains allowed us to regularly anticipate which models would be most appropriate: in the books domain, the propensity of different booksellers to copy each others' claims ("guessing") or systematically disagree with the truth ("lying", e.g. an idiosyncratic way of abbreviating author names) suggested that LieLCA$_s$ was the best fit. For Wikipedia population claims, LieLCA$_m$ and MistakeLCA$_m$ captured the widely varying difficultly of identifying the true population among the cities. In predicting stocks we could expect LieLCA$_m$ and MistakeLCA$_m$ to *not* work because predicting stocks is more-or-less uniformly challenging across companies and per-company difficulty parameters merely worsens the chance of overfitting. Finally, in the Supreme Court domain, we know that historically some sources have been much more accurate than others, but given the even split of votes in most cases it's clear that other sources (a majority) are more-or-less guessing; here we would expect LieLCA$_m$ (which models both guessing and varying difficulty amongst mutual exclusion sets) to perform best, although it's similarly clear why GuessLCA outperforms SimpleLCA.

However, these are qualitative judgements, and while they certainly help us narrow down the set of potential models, it is not always clear precisely which should be used, particularly when partial supervision is not available to empirically estimate performance; e.g. in city populations it is not obvious why MistakeLCA$_m$ outperforms LieLCA$_m$. Arguably, since both of these models do well (and are presumably both reasonably good approximations to the collection of highly varied processes that sources really do follow in generating claims) we could acknowledge that either would be a satisfactory choice. Still, we also wanted briefly investigate the idea of model fit quantitatively, empirically observing how well these models perform given varying quantities of data and a precise knowledge of how the data were really generated (as opposed to real word datasets, where we are left to speculate using our knowledge of the domain). To do so, we generated data using the SimpleLCA joint distribution with the intent of obtaining a simple underlying process that would allow us to focus on the models' behavior.

4.5.1 SimpleLCA Generation

We ran two sets of experiments using a SimpleLCA model to generate data; SimpleLCA does not incorporate guessing, mistake or lie prior probabilities, so in the first set we give GuessLCA, MistakeLCA and LieLCA uniform probabilities. In the second set, however, we generate these priors randomly[2], with the idea that this will give some insight into the effect of a poor model choice when mixed with a bad (random and independent of reality) priors. In each experiment we had 100 sources and 100 mutual exclusion sets, each containing between 2 and 5 claims (selected uniformly at random). The number of claims made by each source was fixed at 3, 5, 10, or 20, and increasing this effectively increased the amount of data provided to the models. To mitigate statistical noise, every experiment was repeated 100 times with 100 different generated datasets, and the reported accuracies are an average of those runs (and, within each experiment, the same 100 randomly-generated datasets were used to test each model).

The distribution of H_s was $Beta(7, 3)$; this prior over H_s was used in all models in both experiment sets, despite H_s having somewhat different semantics in each model (the intent is to observe performance when the models do not fit the data in a well-understood way). The results of our synthetic experiments may be found in Table 3.

There are a number of interesting phenomena that we may observe in these results:

- Surprisingly, with uniform priors, two of the models (GuessLCA and LieLCA$_g$) consistently outperform SimpleLCA on data generated by a SimpleLCA process. In SimpleLCA, the model tends to conclude that, given a disagreement between sources, one is perfectly honest ($H_s = 1$) and the other is constantly wrong ($H_s = 0$). Other models avoid this with guessing, such that even the worst source can always make a lucky guess, which prevents the model from disregarding their claims entirely.

- With sufficient data this overfitting is avoided entirely.

- MistakeLCA$_g$ versus MistakeLCA$_m$: the latter fares quite poorly in all experiments, while the former does quite well, reflecting a substantial difference in the models in practice despite a similar joint distribution. MistakeLCA$_g$'s global D_g parameter controls the frequency sources make mistakes, again creating an alternative explanation for a source's error other than complete dishonesty (since some of their inaccuracy will be attributed to "honest mistakes" rather than dishonesty).

- MistakeLCA$_m$, by contrast, has far more freedom to set its 100 D_m parameters to extreme values (overfitting).

- With randomized priors handicapping the other models, SimpleLCA leads the pack, as expected.

[2]We generated these distributions by drawing a [0,1] value uniformly for each claim and then normalizing over the mutual exclusion set for $P_g(c|s)$ and normalizing over the claims in the mutual set excluding y_m for $P_l(c|y_m, s)$ and $P_e(c|y_m, s)$. This results in a rather complex distribution: for example, given two claims A and B, the probability of guessing A is taken as $\frac{a}{a+b}$, where a is the value drawn for claim A, and b is the value drawn for claim B. Marginalizing out b gives $P_g(A) = a(\log(a+1) - \log(a)))$.

P_g, P_e, P_l	Uniform				Randomized			
Claims per Source	3	5	10	20	3	5	10	20
SimpleLCA	79.92	87.80	95.83	99.54	79.92	87.80	95.83	99.54
GuessLCA	80.10	88.14	95.96	99.54	77.67	84.73	92.51	96.27
MistakeLCA$_g$	79.90	88.08	96.00	99.52	78.03	86.38	94.52	99.10
MistakeLCA$_m$	75.48	78.08	78.87	80.45	70.53	68.99	60.33	56.60
LieLCA$_g$	80.10	88.06	96.01	99.54	78.83	86.96	95.20	99.28
LieLCA$_m$	79.90	87.92	95.85	99.53	76.14	82.24	89.59	94.51
LieLCA$_s$	78.35	86.89	95.58	99.52	75.23	84.54	94.94	99.29

Table 3: Performance of LCA Models with Synthetic Data from a SimpleLCA Process. Each experiment was run over 100 random datasets and the results averaged.

- With randomized priors, MistakeLCA$_m$ suffers from worsening performance as more assertions are made in each mutual exclusion set, increasing the D_m gradients relative to those of H_s and pushing D_m to lower values (it is easier to "explain away" bad assertions by decreasing the D_m for the mutual exclusion set than decreasing the H_s for many sources). This then places greater weight on the (random) mistake priors.
- The other models prove remarkably robust given their completely incorrect priors, although its clear that this does cap the possible performance of GuessLCA and LieLCA$_m$ a bit, whereas MistakeLCA$_g$ and LieLCA$_g$ can simply set a high D_g, eliminating or reducing their influence, respectively.

In our real-world data, SimpleLCA was often among the least accurate LCA models; the synthetic results here suggest that, indeed, even in an artificial best-case scenario other models are able to perform almost as well. However, SimpleLCA remains easy to implement, easy to understand, and very tractable, and so should not be discounted entirely. It is also apparent that MistakeLCA$_m$ may face severe difficulty in some cases; whereas LieLCA$_m$ can believe a source will assert the correct claim by guessing even if the D_m parameter for the relevant mutual exclusion set is 0, MistakeLCA$_m$ has no "safety valve" of this sort: if D_m is 0, the source must always get the claim wrong (this creates a sort of perverse "anti-vote", whereby the claim with the fewest assertions is likely to be believed). This danger manifests itself in the high variance we see in the model's real-world performance; while the top performer in the population domain, it is also the lowest performer in the stocks domain. Care must therefore be taken to ensure that MistakeLCA$_m$ is a reasonably good fit to the domain, whereas the other models are much more forgiving.

4.5.2 Discussion

Our synthetic experiments are limited in scope, but they do inform our approach to real-world problems. MistakeLCA$_m$ can sometimes yield the best results, but LieLCA$_m$ has a similar generative story and is a less variable choice that can do well in the same domains without MistakeLCA$_m$'s risk of overfitting. A second lesson is that these models can be remarkably resistant to bad priors (when the underlying process generating the data is simple), and uniform priors are a good choice even if the generating process is quite different from the model being applied. GuessLCA in particular does quite well with uniform priors in our synthetic experiments, and, moreover, performs consistently well in the real-world experiments, too. This consistency is partly due to its simplicity (little danger of overfitting) and partly because it manages to at least approximately model the important "difficulty" aspect of claims; not as precisely as the more sophisticated LieLCA or MistakeLCA models, of course, but also without their computational cost. LieLCA and MistakeLCA are, on the other hand, more appropriate where the behavior of sources is well understood (e.g. the books domain) and where partial supervision can be used to avoid overfitting (e.g. the stocks domain).

5. CONCLUSION

Latent Credibility Analysis is a flexible and powerful approach to modeling the information credibility problem; although we have really only begun to explore its potential in our experiments so far, we have nonetheless seen that the performance of LCA models surpasses that of fact-finders on both semi-supervised and unsupervised real-world datasets, often substantially. GuessLCA in particular is promising due to its consistently strong performance and tractability, scaling linearly with the size of the problem as fact-finders do, although other, more expressive (and expensive) LCA models can achieve better results when used judiciously. Future work should extend the LCA framework, capturing phenomena such as source dependency and real-valued claims that will allow it to model an even wider range of domains; for now, however, LCAs are a new approach to credibility that is already both semantically appealing and of substantial practical utility.

This research was sponsored in part by the Army Research Laboratory (ARL) under agreement W911NF-09-2-0053. Any opinions, findings, conclusions or recommendations are those of the authors and do not necessarily reflect the view of the ARL.

6. REFERENCES

[1] M. Chang, L. Ratinov, and D. Roth. Structured Learning with Constrained Conditional Models. *Machine Learning*, 88(3):399–431, 2012.

[2] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[3] X. Dong, L. Berti-Equille, and D. Srivastava. Truth discovery and copying detection in a dynamic world. *VLDB*, 2009.

[4] A. Galland, S. Abiteboul, A. Marian, and P. Senellart. Corroborating information from disagreeing views. In *WSDM*, 2010.

[5] K. Ganchev, J. Graca, J. Gillenwater, and B. Taskar. Posterior Regularization for Structured Latent Variable Models. *Journal of Machine Learning Research*, 2010.

[6] P. Jorion. Risk management lessons from Long-Term Capital Management. *European financial management*, 6(3):277–300, 2000.

[7] A. Josang. Artificial reasoning with subjective logic. *2nd Australian Workshop on Commonsense Reasoning*, 1997.

[8] A. Josang, S. Marsh, and S. Pope. Exploring different types of trust propagation. *Lecture Notes in Computer Science*, 3986:179, 2006.

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

[10] H. K. Le, J. Pasternack, H. Ahmadi, M. Gupta, Y. Sun, T. Abdelzaher, J. Han, D. Roth, B. Szymanski, and S. Adali. Apollo : Towards Factfinding in Participatory Sensing. *IPSN*, 2011.

[11] B. G. Malkiel. The efficient market hypothesis and its critics. *Journal of Economic Perspectives*, pages 59–82, 2003.

[12] Y. Nesterov and I. U. E. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer, 2004.

[13] J. Pasternack and D. Roth. Knowing What to Believe (when you already know something). In *COLING*, 2010.

[14] J. Pasternack and D. Roth. Making Better Informed Trust Decisions with Generalized Fact-Finding. In *IJCAI*, 2011.

[15] G. Shafer. *A mathematical theory of evidence*. Princeton University Press Princeton, NJ, 1976.

[16] V. G. Vydiswaran, C. X. Zhai, and D. Roth. Content-driven trust propagation framework. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 974–982. ACM, 2011.

[17] D. Wang, T. Abdelzaher, H. Ahmadi, J. Pasternack, D. Roth, M. Gupta, J. Han, O. Fatemieh, H. Le, and C. Aggarwal. On bayesian interpretation of fact-finding in information networks. *Information Fusion*, 2011.

[18] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. *In Proc. of SIGKDD*, 2007.

[19] X. Yin, P. S. Yu, and J. Han. Truth Discovery with Multiple Conflicting Information Providers on the Web. *IEEE Transactions on Knowledge and Data Engineering*, 20(6):796–808, 2008.

[20] B. Yu and M. P. Singh. Detecting deception in reputation management. *Proceedings of the second international joint conference on Autonomous agents and multiagent systems - AAMAS '03*, page 73, 2003.

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