Fake News Detection in Social Networks via Crowd Signals

Sebastian Tschiatschek*
Microsoft Research
Cambridge, United Kingdom
setschia@microsoft.com

Adish Singla
MPI-SWS
Saarbrücken, Germany
adishs@mpi-sws.org

Manuel Gomez Rodriguez
MPI-SWS
Kaiserslautern, Germany
manuelgr@mpi-sws.org

Arpit Merchant
IIIT-H
Hyderabad, India
arpitdm@gmail.com

Andreas Krause
ETH Zurich
Zurich, Switzerland
krausea@ethz.ch

ABSTRACT

Our work considers leveraging crowd signals for detecting fake news and is motivated by tools recently introduced by Facebook that enable users to flag fake news. By aggregating users' flags, our goal is to select a small subset of news every day, send them to an expert (e.g., via a third-party fact-checking organization), and stop the spread of news identified as fake by an expert. The main objective of our work is to minimize *the spread of misinformation* by stopping the propagation of fake news in the network. It is especially challenging to achieve this objective as it requires detecting fake news with high-confidence as quickly as possible. We show that in order to leverage users' flags efficiently, it is crucial to learn about users' flagging accuracy. We develop a novel algorithm, DETECTIVE, that performs Bayesian inference for detecting fake news and jointly learns about users' flagging accuracy over time. Our algorithm employs posterior sampling to actively trade off exploitation (selecting news that maximize the objective value at a given epoch) and exploration (selecting news that maximize the value of information towards learning about users' flagging accuracy). We demonstrate the effectiveness of our approach via extensive experiments and show the power of leveraging community signals for fake news detection.

ACM Reference Format:
Sebastian Tschiatschek, Adish Singla, Manuel Gomez Rodriguez, Arpit Merchant, and Andreas Krause. 2018. Fake News Detection in Social Networks via Crowd Signals. In *WWW '18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3184558.3188722

1 INTRODUCTION

Fake news (a.k.a. hoaxes, rumors, etc.) and the spread of misinformation have dominated the news cycle since the US presidential election (2016). Social media sites and online social networks, for example Facebook and Twitter, have faced scrutiny for being unable to curb the spread of fake news. There are various motivations for generating and spreading fake news, for instance, making political gains, harming the reputation of businesses, as clickbait for increasing advertising revenue, and for seeking attention[1]. As a concrete example, Starbucks recently fell victim to fake news with a hoax advertisement claiming that the coffee chain would give free coffee to undocumented immigrants[2]. While Starbucks raced to deny this claim by responding to individual users on social media, the lightening speed of the spread of this hoax news in online social media highlighted the seriousness of the problem and the critical need to develop new techniques to tackle this challenge. To this end, Facebook has recently announced a series of efforts towards tackling this challenge [10, 11].

Detection via expert's verification. Fake news and misinformation have historically been used as tools for making political or business gains [9]. However, traditional approaches based on verification by human editors and expert journalists do not scale to the volume of news content that is generated in online social networks. In fact, it is this volume as well as the lightening speed of spread in these networks that makes this problem challenging and requires us to develop new computational techniques. We note that such computational techniques would typically complement, and not replace, the expert verification process—even if a news is detected as fake, some sort of expert verification is needed before one would actually block it. This has given rise to a number of third-party fact-checking organizations such as Snopes[3] and Factcheck.org[4] as well as a code of principles [25] that should be followed by these organizations.

Detection using computational methods. There has been a recent surge in interest towards developing computational methods for detecting fake news (cf., [7] for a survey)—we provide a more detailed overview of these methods in the Related Work section. These methods are typically based on building predictive models to classify whether a news is fake or not via using a combination of features related to news content, source reliability, and network structure. One of the major challenges in training such predictive models is the limited availability of corpora and the subjectivity of labelling news as fake [27, 33]. Furthermore, it is difficult to design methods based on estimating source reliability

*Work performed while at ETH Zurich.

WWW '18 Companion, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5640-4/18/04.
https://doi.org/10.1145/3184558.3188722

[1] Snopes compiles a list of top 50 fake news stories: http://www.snopes.com/50-hottest-urban-legends/
[2] http://uk.businessinsider.com/fake-news-starbucks-free-coffee-to-undocumented-immigrants-2017-8
[3] http://www.snopes.com/
[4] http://factcheck.org/

and network structure as the number of users who act as sources is diverse and gigantic (e.g., over one billion users on Facebook); and the sources of fake news could be normal users who unintentionally share a news story without realizing that the news is fake. A surge of interest in the problem and in overcoming these technical challenges has led to the establishment of a volunteering based association—FakeNewsChallenge[5]—comprising over 100 volunteers and 70 teams which organizes machine learning competitions related to the problem of detecting fake news.

1.1 Leveraging users' flags.

Given the limitation of the current state-of-the-art computational methods, an alternate approach is to develop hybrid human-AI methods via engaging users of online social networks by enabling them to report fake news. In fact, Facebook has recently taken steps towards this end by launching a fake news reporting tool in Germany [11], as shown in Figure 1. The idea of this tool is that as news propagates through the network, users can flag the news as fake.

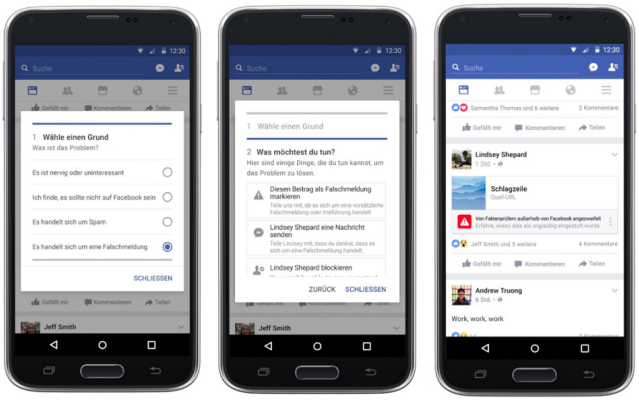


Figure 1: Facebook has launched tools in Germany to report fake news. Image source: [11].

As proposed by Facebook [11], the aggregated users' flags as well as well as other available signals can be used to identify a set of news which potentially is fake. These news can then be sent to an expert for review via a third-party fact-checking organization. If an expert labels the news as fake, it could be removed from the network or marked as disputed making it appear lower in news-feed ranking. The contemporary work by Kim et al. [16] explored the idea of detecting fake news via leveraging users' flagging activity by using the framework of marked temporal point processes. We highlight the key differences of their approach to ours in the next section.

1.2 Our Contributions

In this paper, we develop algorithmic tools to effectively utilize the power of the crowd (flagging activity of users) to detect fake news. Given a set of news, our goal is to select a small subset of k news, send them to an expert for review, and then block the news which are labeled as fake by the expert. We formalize our objective as to minimize *the spread of misinformation*, i.e., how many users end up seeing a fake news before it is blocked. We design our algorithm DETECTIVE, which implements a Bayesian approach for learning about users' accuracies over time as well as for performing inference to find which news are fake with high confidence. In short, our main contributions include:

- We formalize the problem of leveraging users' flagging activity for detection of fake news. We showcase the need to learn about users' accuracy in order to effectively leverage their flags in a robust way.
- We develop a tractable Bayesian algorithm, DETECTIVE, that actively trades off between exploitation (selecting news that directly maximize the objective value) and exploration (selecting news that helps towards learning about users' flagging accuracy).
- We perform extensive experiments using a publicly available Facebook dataset to demonstrate the effectiveness of our approach. We plan to make the code publicly available so that other researchers can build upon our techniques for this important and timely problem of detecting fake news.

2 RELATED WORK

Contemporary results. Kim et al. [16] explored the idea of detecting fake news via leveraging users' flagging activity. In particular, they introduce a flexible representation of the above problem using the framework of marked temporal point processes. They develop an algorithm, CURB, to select which news to send for fact-checking via solving a novel stochastic optimal control problem. The key technical differences of the approach by Kim et al. [16] to ours are: (1) we learn about the flagging accuracy of individual users in an online setting; in contrast, they consider all users to be equally reliable and estimate the flagging accuracy of the population of users from historical data; (2) our algorithms are agnostic to the actual propagation dynamics of news in the network; they model the actual propagation dynamics as a continuous-time dynamical system with jumps and arrive at an algorithm by casting the problem as an optimal control problem; and (3) we use discrete epochs with a fixed budget per epoch (*i.e.*, the number of news that can be sent to an expert for reviewing); they use continuous time and consider an overall budget for their algorithm.

Computational methods for detecting fake news. There is a large body of related work on rumor detection and information credibility evaluation (with a more recent focus on fake news detection) that are applicable to the problem of detecting fake news. These methods are typically based on building predictive models to classify whether a news is fake. At a high-level level, we can categorize these methods as follows: (i) based on features using news content via natural language processing techniques [13, 31, 34, 38]; (ii) via learning models of source reliability and trustworthiness [20, 22, 28]; (iii) by analyzing the network structure over which a news propagated [6]; and (iv) based on a combination of the above-mentioned features, i.e., linguistic, source, and network structure [1, 17, 18, 35]. As we pointed out in the Introduction, there are several key challenges in building accurate predictive models for identifying fake news including limited availability of corpora, subjectivity in ground truth labels, and huge variability in the sources

[5] http://www.fakenewschallenge.org/

who generate fake news (often constituting users who do it unintentionally). In short, these methods alone have so far proven to be unsuccessful in tackling the challenge of detecting fake news.

Leveraging crowd signals for web applications. Crowdsourcing has been used in both industrial applications and for research studies in the context of different applications related to web security. For instance, [23] and [5] have evaluated the potential of leveraging the wisdom of crowds for assessing phishing websites and web security. Their studies show a high variability among users—(i) the participation rates of users follows a power-law distribution, and (ii) the accuracy of users' reports vary, and users with more experience tend to have higher accuracy. The authors also discuss the potential of voting fraud when using users' reports for security related applications. Wang et al. [32] performed a crowdsourcing study on Amazon's Mechanical Turk for the task of sybil detection in online social networks. Their studies show that there is a huge variability among crowd users in terms of their reporting accuracies that needs to be taken into account for building a practical system. Chen et al. [3], Zheleva et al. [39] present a system similar to that of ours for the task of filtering email spam and SMS spam, respectively. The authors discuss a users' reputation system whereby reliable users (based on history) can be weighted more when aggregating the reports. However, their work assumes that users' reputation/reliability is known to the system, whereas the focus of our paper is on learning users' reputation over time. Freeman [12] discusses the limitations of leveraging user feedback for fake account detection in online social networks—via data-driven studies using Linkedin data, the authors show that there is only a small number of skilled users (who have good accuracy that persists over time) for detecting fake accounts.

Crowdsourcing with expert validation On a technical side, our approach can be seen as that of a semi-supervised crowdsourcing technique where users' answers can be validated via an external expert. Hung et al. [14], Liu et al. [21] present probabilistic models to select specific news instances to be labeled by experts that would maximize the reduction in uncertainty about users' accuracy. With a similar flavor to ours, Zhao et al. [36] presents a Bayesian approach to aggregate information from multiple users, and then jointly infer users' reliability as well as ground truth labels. Similar to our approach, they model users' accuracy via two separate parameters for false positive and false negative rates. However, their approach is studied in an unsupervised setting where no expert validation (ground truth labels) are available.

3 THE MODEL

We provide a high-level specification of our model in Protocol 1. There is an underlying social network denoted as $G = (U, E)$ where U is the set of users in the network. We divide the execution into different epochs denoted as $t = 1, 2, \ldots, T$, where each epoch could denote a time window, for instance, one day. Below, we provide details of our model—the process of news generation and spread, users' activity of flagging the news, and selecting news to get expert's labels.

3.1 News Generation and Spread

We assume that new news, denoted by the set X^t, are generated at the beginning of every epoch t (cf., line 4).[6] In this paper, we consider a setting where each news has an underlying label (unknown to the algorithm) of being "fake" (f) or "not fake" ($\bar{f}$). We use random variable $Y^*(x)$ to denote this unknown label for a news x and its realization is given by $y^*(x) \in \{f, \bar{f}\}$. The label $y^*(x)$ can only be acquired if news x is sent to an expert for review who would then provide the true label. We maintain a set of "active" news A^t (cf., line 5) which consists of all news that have been generated by the end of epoch t but for which expert's label have not been acquired yet.

Each news x is associated with a source user who seeded this news, denoted as o_x (cf., line 4). We track the spread of news in the set A^t via a function $\pi^t : A^t \rightarrow 2^U$. For a news $a \in A^t$, the function $\pi^t(a)$ returns the set of users who have seen the news a by the end of epoch t. During epoch t, let $u^t(a) \subseteq U \setminus \pi^{t-1}(a)$ be the set of additional users (possibly the empty set) to whom news $a \in A^t$ propagates in epoch t, hence $\pi^t(a) = \pi^{t-1}(a) \cup u^t(a)$ (cf., line 9).

3.2 Users' Activity of Flagging the News

In epoch t, when a news $a \in A^t$ propagates to a new user $u \in u^t(a)$, this user can flag the news to be fake. We denote the set of users who flag news a as fake in epoch t via a set $l^t(a) \subseteq u^t(a)$ (cf., line 10). Furthermore, the function $\psi^t(a)$ returns the complete set of users who have flagged the news a as fake by the end of epoch t.[7] For any news x and any user $u \in U$, we denote the label user u would assign to x via a random variable $Y_u(x)$. We denote the realization of $Y_u(x)$ as $y_u(x) \in \{f, \bar{f}\}$ where $y_u(x) = f$ signifies that user has flagged the news as fake. In this paper, we consider a simple, yet realistic, probabilistic model of a user's flagging activity as discussed below.

User abstaining from flagging activity. Reflecting the behavior of real-world users, user u might abstain from actively reviewing the news content (and by default, does not flag the news)—we model this happening with a probability $\gamma_u \in [0, 1]$. Intuitively, we can think of $1 - \gamma_u$ as the engagement of user u while participating in this crowdsourcing effort to detect fake news: $\gamma_u = 1$ means that the user is not participating at all.

User's accuracy in flagging the news. With probability $(1 - \gamma_u)$, user u reviews the content of news x and labels the news. We model the accuracy/noise in the user's labels, conditioned on that the user is reviewing the content, as follows:

- $\alpha_u \in [0, 1]$ denotes the probability that user u would not flag the news as fake, conditioned on that *news x is not fake and the user is reviewing the content.*
- $\beta_u \in [0, 1]$ denotes the probability that user u would flag the news as fake, conditioned on that *news x is fake and the user is reviewing the content.*

User's observed activity. Putting this together, we can quantify the observed flagging activity of user u for any news x with the

[6] For simplicity of presentation, we consider every news generated in the network to be unique. In real-world settings, the same news might be posted by multiple users because of externalities, and it is easy to extend our model to consider this scenario.

[7] Note that as per specification of Protocol 1, for any news x, the source user o_x doesn't participate in flagging x.

Protocol 1: High-level specification of our model

1 **Input**: social network graph $G = (U, E)$; labeling budget per epoch k.
2 **Initialize**: active news $A^0 = \{\}$ (i.e., news for which expert's label is not acquired yet).
3 **foreach** $t = 1, 2, \ldots, T$ **do**
/* At the beginning of epoch t */
4 News X^t are generated with $o_x \in U$ as the origin/source of $x \in X^t$.
5 Update the set of active news as $A^t = A^{t-1} \cup X^t$. $\forall x \in X^t$, do the following:
6 Initialize users exposed to the news x as $\pi^{t-1}(x) = \{\}$.
7 Initialize users who flagged the news x as $\psi^{t-1}(x) = \{\}$.
/* During the epoch t */
8 News A^t continue to propagate in the network. $\forall a \in A^t$, do the following:
9 News a propagates to more users $u^t(a) \subseteq U \setminus \pi^{t-1}(a)$; i.e., $\pi^t(a) = \pi^{t-1}(a) \cup u^t(a)$.
10 News a is flagged as fake by users $l^t(a) \subseteq u^t(a)$; i.e., $\psi^t(a) = \psi^{t-1}(a) \cup l^t(a)$.
/* At the end of epoch t */
11 Algorithm Algo selects a subset $S^t \subseteq A^t$ of up to size k to get expert's labels given by $y^*(s) \in \{f, \bar{f}\}\ \forall\ s \in S^t$.
12 Block the fake news, i.e., $\forall s \in S^t$ s.t. $y^*(s) = f$, remove s from the network.
13 Update the set of active news as $A^t = A^t \setminus S^t$
Note that news $s \in S^t$ s.t. $y^*(s) = \bar{f}$ remain in the network, continue to propagate, and being flagged by users

following matrix defined by variables $(\theta_{u,\bar{f}}, \theta_{u,f})$:

$$\begin{bmatrix} \theta_{u,\bar{f}} & 1-\theta_{u,f} \\ 1-\theta_{u,\bar{f}} & \theta_{u,f} \end{bmatrix} = \gamma_u \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} + (1-\gamma_u) \begin{bmatrix} \alpha_u & 1-\beta_u \\ 1-\alpha_u & \beta_u \end{bmatrix}$$

where

$$\begin{cases} \theta_{u,\bar{f}} & \equiv P\big(Y_u(x) = \bar{f} \mid Y^*(x) = \bar{f}\big) \\ 1-\theta_{u,\bar{f}} & \equiv P\big(Y_u(x) = f \mid Y^*(x) = \bar{f}\big) \\ \theta_{u,f} & \equiv P\big(Y_u(x) = f \mid Y^*(x) = f\big) \\ 1-\theta_{u,f} & \equiv P\big(Y_u(x) = \bar{f} \mid Y^*(x) = f\big) \end{cases}$$

The two parameters (α_u, β_u) allow us to model users of different types that one might encounter in real-world settings. For instance,

- a user with $(\alpha_u \geq 0.5, \beta_u \leq 0.5)$ can be seen as a "news lover" who generally tends to perceive the news as not fake; on the other hand, a user with $(\alpha_u \leq 0.5, \beta_u \geq 0.5)$ can be seen as a "news hater" who generally tends to be skeptical and flags the news (i.e., label it as fake).
- a user with $(\alpha_u = 1, \beta_u = 1)$ can be seen as an "expert" who always labels correctly; a user with $(\alpha_u = 0, \beta_u = 0)$ can be seen as a "spammer" who always labels incorrectly.

3.3 Selecting News to Get Expert's Label

At the end of every epoch t, we apply an algorithm Algo—on behalf of the network provider—which selects news $S^t \subseteq A^t$ to send to an expert for reviewing and acquiring the true labels $y^*(s)\ \forall s \in S^t$ (cf., line 11). If a news is labeled as fake by the expert (i.e., $y^*(s) = f$), this news is then blocked from the network (cf., line 12). At the end of the epoch, the algorithm updates the set of active news as $A^t = A^t \setminus S^t$ (cf., line 13). We will develop our algorithm in the next section; below we introduce the formal objective of minimizing the spread of misinformation via fake news in the network.

3.4 Objective: Minimizing the Spread of Fake News

Let's begin by quantifying the utility of blocking a news $a \in A^t$ at epoch t—it is important to note that, by design, only the fake news are being blocked in the network. Recall that $|\pi^t(a)|$ denotes the number of users who have seen news a by the end of epoch t. We introduce $|\pi^\infty(a)|$ to quantify the number of users who would *eventually* see the news a if we let it spread in the network. Then, if a news a is fake, we define the utility of blocking news a at epoch t as $\text{val}^t(a) = |\pi^\infty(a)| - |\pi^t(a)|$, i.e., the utility corresponds to the number of users saved from being exposed to fake news a. If an algorithm Algo selects set S^t in epoch t, then the total expected utility of the algorithm for $t = 1, \ldots, T$ is given by

$$\text{Util}(T, \text{Algo}) = \sum_{t=1}^{T} \mathbb{E}\Big[\sum_{s \in S^t} \mathbf{1}_{\{y^*(s)=f\}} \text{val}^t(s) \Big] \tag{1}$$

where the expectation is over the randomness of the spread of news and the randomness in selecting $S^t\ \forall t \in \{1, \ldots, T\}$.
In this work, we will assume that the quantity $\text{val}^t(a)$ in Equation 1 can be estimated by the algorithm. For instance, this can be done by fitting parameters of an information cascade model on the spread $\pi^t(a)$ seen so far for news a, and then simulating the future spread by using the learnt parameters [8, 26, 37].
Given the utility values $\text{val}^t(\cdot)$, we can consider an oracle Oracle that has access to the true labels $y^*(\cdot)$ for all the news and maximizes the objective in Equation 1 by simply selecting k fake news with highest utility. In the next section, we develop our algorithm Detective that performs Bayesian inference to compute $y^*(\cdot)$ using the flagging activity of users as well as via learning users' flagging accuracy $\{\theta_{u,\bar{f}}, \theta_{u,f}\}_{u \in U}$ from historic data.

4 OUR METHODOLOGY

In this section we present our methodology and our algorithm Detective. We start by describing how news labels can be inferred

for the case in which users' parameters are fixed. Next, we consider the case in which users' parameters are unknown and employ a Bayesian approach for inferring news labels and learning users' parameters. Given a prior distributions on the users' parameters and a history of observed data (users' flagging activities and experts' labels obtained), one common approach is to compute a point estimate for the users' parameters (such as MAP) and use that. However, this can lead to suboptimal solutions because of limited exploration towards learning users' parameters. In DETECTIVE, we overcome this issue by employing the idea of *posterior sampling* [24, 29].

4.1 Inferring News Labels: Fixed Users' Params

We take a Bayesian approach to deal with unknown labels $y^*(\cdot)$ for maximizing the objective in Equation 1. As a warm-up, we begin with a simpler setting where we fix the users' labeling parameters $(\theta_{u,\bar{f}}, \theta_{u,f})$ for all users $u \in U$. Let's consider epoch t and news $a \in A^t$ for which we want to infer the true label $y^*(a)$. Let ω be the prior that a news is fake; then, we are interested in computing:

$$\begin{aligned} &P(Y^*(a) = f \mid \{\theta_{u,\bar{f}}, \theta_{u,f}\}_{u\in U}, \omega, \psi^t(a), \pi^t(a)) \\ &\propto \omega \cdot \prod_{u\in\psi^t(a)} P\Big(Y_u(a) = f \mid Y^*(a) = f, \theta_{u,f}\Big) \cdot \\ &\qquad \prod_{u\in\pi^t(a)\setminus\psi^t(a)} P\Big(Y_u(a) = \bar{f} \mid Y^*(a) = f, \theta_{u,f}\Big) \\ &= \omega \cdot \prod_{u\in\psi^t(a)} \theta_{u,f} \cdot \prod_{u\in\pi^t(a)\setminus\psi^t(a)} (1-\theta_{u,f}) \end{aligned}$$

where the last two steps follow from applying Bayes rule and assuming that users' labels are generated independently. Note that both users' parameters $\{\theta_{u,\bar{f}}, \theta_{u,f}\}_{u\in U}$ affect the posterior probability of a news being fake as the normalization constant depends on both $P(Y^*(a) = f \mid \cdot)$ and $P(Y^*(a) = \bar{f} \mid \cdot)$.

At every time $t \in \{1, \ldots, T\}$, we can use the inferred posterior probabilities to greedily select k news $S^t \subseteq A^t, |S^t| = k$ that maximize the total expected utility, i.e.,

$$\sum_{s\in S^t} P(Y^*(s) = f \mid \cdot) \cdot \text{val}^t(s). \tag{2}$$

This greedy selection can be performed optimally by selecting k news with the highest expected utility. This is implemented in our algorithm TOPX, shown in Algorithm 2.

4.2 Inferring News Labels: Learning Users' Params

In our setting, the users' parameters $\{\theta_{u,\bar{f}}, \theta_{u,f}\}_{u\in U}$ are unknown and need to be learnt over time.

Learning about users. We assume a prior distribution over the users' parameters $(\Theta_{\bar{f}}, \Theta_f)$ shared among all users. For each user $u \in U$, we maintain the data history in form of the following matrix:

$$\mathcal{D}_u^t = \begin{bmatrix} d^t_{u,\bar{f}|\bar{f}} & d^t_{u,\bar{f}|f} \\ d^t_{u,f|\bar{f}} & d^t_{u,f|f} \end{bmatrix}.$$

The entries of this matrix are computed from the news for which experts' labels were acquired. For instance, $d^t_{u,\bar{f}|\bar{f}}$ represents the count of how often the user u labeled a news as not fake and the acquired expert's label was not fake.

Given $\mathcal{D}_u^t$, we can compute the posterior distribution over the users' parameters using Bayes rules as follows:

$$\begin{aligned} P(\theta_{u,\bar{f}} \mid \Theta_{\bar{f}}, \mathcal{D}_u^t) &\propto P(\mathcal{D}_u^t \mid \theta_{u,\bar{f}}) \cdot P(\theta_{u,\bar{f}} \mid \Theta_{\bar{f}}) \\ &= (\theta_{u,\bar{f}})^{d^t_{u,\bar{f}|\bar{f}}} \cdot (1-\theta_{u,\bar{f}})^{d^t_{u,f|\bar{f}}} \cdot P(\theta_{u,\bar{f}} \mid \Theta_{\bar{f}}) \end{aligned}$$

Similarly, one can compute $P(\theta_{u,f} \mid \cdot)$.

Inferring labels. We can now use the users' parameters posteriors distributions to infer the labels, for instance, by first computing the MAP parameters

$$\theta^{\text{MAP}}_{u,\bar{f}} = \arg\max_{\theta_{u,\bar{f}}} P(\theta_{u,\bar{f}} \mid \Theta_{\bar{f}}, \mathcal{D}_u^t)$$

(and $\theta^{\text{MAP}}_{u,f}$ similarly) and invoking the results from the previous section.[8] Then, at every epoch t we can invoke TOPX with a point estimate for the users' parameters to select a set S^t of news. However this approach can perform arbitrarily bad compared to an algorithm that knows the true users' parameters (we refer to this algorithm as OPT) as we show in our analysis. The key challenge here is that of actively trading off exploration (selecting news that maximize the value of information towards learning users' parameters) and exploration (selecting news that directly expected utility at a given epoch). This is a fundamental challenge that arises in sequential decision making problems, e.g., in multi-armed bandits [2], active search [4, 30] and reinforcement learning.

Algorithm 2: Algorithm TOPX

Input:
- Active news A^t; information $\text{val}^t(\cdot), l^t(\cdot), \pi^t(\cdot)$
- budget k; news prior ω
- users' parameters $\{\theta_{u,\bar{f}}, \theta_{u,f}\}_{u\in U}$.

Compute $p(a)$ for all $a \in A^t$ as
$P(Y^*(a) = f \mid \{\theta_{u,\bar{f}}, \theta_{u,f}\}_{u\in U}, \omega, l^t(a), \pi^t(a))$

Select
$S^t = \arg\max_{S\subseteq A^t, |S|\le k} \sum_{a\in S} p(a)\text{val}^t(a)$

Return: S^t

4.3 Our Algorithm DETECTIVE

In this section, we present our algorithm DETECTIVE, shown in Algorithm 3, that actively trades off between exploration and exploitation by the use of posterior sampling aka Thompson sampling [24, 29]. On every invocation, the algorithm samples the users' parameters from the current users' posterior distributions and invokes TOPX with these parameters. Intuitively, we can think of this approach as sampling users' parameters according to the probability they are optimal.

Analysis. We analyze our algorithms in a simplified variant of Protocol 1, in particular we make the following simplifications:

[8] Note that a fully Bayesian approach for integrating out uncertainty about users' parameters in this case is equivalent to using the mean point estimate of the posterior distribution.

Algorithm 3: Algorithm DETECTIVE

Input:
- user priors $\Theta_f, \Theta_{\bar{f}}$; users' histories $\{\mathcal{D}_u^t\}_{u \in U}$.

Sample
$\theta_{u,\bar{f}} \sim P(\theta_{u,\bar{f}} \mid \Theta_{\bar{f}}, \mathcal{D}_u^t)$, $\theta_{u,f} \sim P(\theta_{u,f} \mid \Theta_f, \mathcal{D}_u^t)$

$S^t \leftarrow$ Invoke TOPX with parameters $\{\theta_{u,\bar{f}}, \theta_{u,f}\}_{u \in U}$

Return: S^t

(1) There are M sources $o_1, \ldots, o_M$, each generating news every epoch t.
(2) For any news x seeded at epoch t, $\text{val}^\tau(x) > 0$ only for $\tau = t$. This means that news x reaches it maximum spread at the next timestep $t+1$, hence the utility of detecting that news drops to 0.

To state our theoretical results, let us introduce the regret of an algorithm ALGO as

$$\text{Regret}(T, \text{ALGO}) = \text{Util}(T, \text{OPT}) - \text{Util}(T, \text{ALGO}).$$

We can now immediately state our first theoretical result, highlighting the necessity of exploration.

PROPOSITION 1. *Any algorithm ALGO using deterministic point estimates for the users' parameters suffers linear regret, i.e.,*

$$\text{Regret}(T, \text{ALGO}) = \Theta(T).$$

PROOF SKETCH. The proof follows by considering a simple problem involving two users, where we have perfect knowledge about one user with parameters $(0.5+\epsilon, 0.5+\epsilon)$ and the other user either has parameters $(1,1)$ or $(0,0)$ (*expert* or *spamer*). The key idea here is that any algorithm using point estimates can be tricked into always making decisions based on the first user's flagging activities and is never able to learn about the perfect second user. □

The above result is a consequence of insufficient exploration which is overcome by our algorithm DETECTIVE, as formalized by the following theorem.

THEOREM 1. *The expected regret of our algorithm DETECTIVE is* $\mathbb{E}[\text{Regret}(T, \text{DETECTIVE})] = O(C\sqrt{M'T\log(CM'T)})$, *where* $M' = \binom{M}{k}$ *and* C *is a problem dependent parameter. C quantifies the total number of realizations of how M news can spread to U users and how these users label the news.*

PROOF SKETCH. The proof of this theorem follows via interpreting the simplified setting as a reinforcement learning problem. Then, we can apply the generic results for reinforcement learning via posterior sampling of Osband et al. [24]. In particular, we map our problem to an MDP with horizon 1 as follows. The actions in the MDP correspond to selecting k news from the M sources, the reward for selecting a set of news S is given by Equation 2 (evaluated using the true users' parameters). □

Given that the regret only grows as $O(\sqrt{T})$ (i.e., sublinear in T), this theorem implies that DETECTIVE converges to OPT as $T \to \infty$. However, as a conservative bound on C could be exponential in $|U|$ and M, convergence may be slow. Nevertheless, in practice we observe competitive performance of DETECTIVE compared to OPT as indicated in our experiments. Hence, DETECTIVE overcomes the issues in Proposition 1, and actively trades off exploration and exploitation.

5 EXPERIMENTAL SETUP

Social network graph and news generation. We consider the *social circles Facebook* graph [19], consisting of 4,039 users (nodes) U and 88,234 edges, computed from survey data collected by using a Facebook app for identifying social circles. Every user can be the seed of news as described shortly and to every user a probability is assigned with which it (hypothetically) generates fake news in case it seeds news. In particular, 20% of the users generate fake news with probability 0.6, 40% of the users generate fake news with probability 0.2 and the remaining 40% of the users generate fake news with probability 0.01 (the class of a user is assigned randomly). For determining the seeds of news, we partition the users into users U_n which commonly spread news and users $U_r = U \setminus U_n$ which only occasionally spread news. That is, in every iteration of Protocol 1, we select $M = 25$ users for generating news, where users in U_n are selected with probability $\frac{0.5}{|U_n|}$ and users in U_r are selected with probability $\frac{0.5}{|U_r|}$. Hence, in our experimental setup this corresponds to a prior for seeding fake news of about 20%, i.e., $\omega \approx 0.2$.

News spreading. In our experiments, news spread according to an independent cascade model [15], i.e., the diffusion process of every news is a separate independent cascade with infection probability $0.1 + \mathcal{U}[0, 0.1]$ (fixed when the news is seeded). In every epoch of Protocol 1, we perform two iterations of the independent cascade models to determine the news spread at the next epoch. We estimate the number of users who would eventually see news a, i.e., $|\pi^\infty(a)|$, by executing the independent cascade models for each news for 600 iterations.

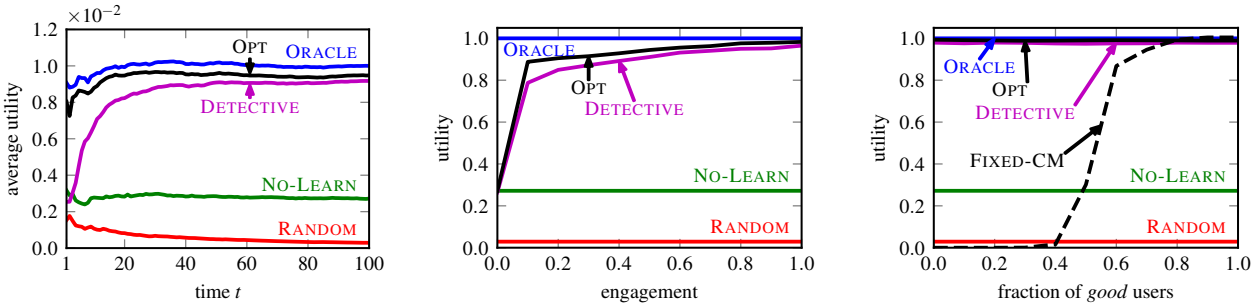
Users' parameters. In our experiments we consider three types of users, i.e., *good users* ($\alpha_u = \beta_u = 0.9$), *spammers* ($\alpha_u = \beta_u = 0.1$) and *indifferent users* ($\alpha_u = \beta_u = 0.5$). Unless specified otherwise, each user is randomly assigned to one of these three types. Also, we set $\gamma_u = 0$ unless specified otherwise (note that $1 - \gamma_u$ quantifies the engagement of a user).

Algorithms. We execute Protocol 1 for $T = 100$ epochs. In every epoch of Protocol 1, the evaluated algorithms select $k = 5$ news to be reviewed by an expert. In our experiments we compare the performance of DETECTIVE, OPT (unrealistic: TOPX invoked with the true users' parameters), ORACLE (unrealistic: knows the true news labels). In addition, we consider the following baselines:

- FIXED-CM. This algorithm leverages users' flags without learning about or distinguishing between users. It uses fixed users parameters $\theta_{u,\bar{f}} = \theta_{u,f} = 0.6$ for invoking TOPX.
- NO-LEARN. This algorithm does not learn about users and does not consider any user flags. It greedily selects those news with highest $\text{val}^t(\cdot)$, i.e.,

$$S^t = \arg \max_{S \subseteq A^t, |S| = k} \sum_{s \in S} \text{val}^t(s).$$

- RANDOM. This algorithm selects a random set $S^t \subseteq A^t, |S^t| = k$ of active news for labeling by experts.



(a) Learning about users **(b) Users' engagement in flagging.** **(c) Robustness against spammers**

Figure 2: Experimental results. *(a)* Learning about users: Detective achieves average utility competitive compared to that of Oracle (which knows the true news labels). The average utility of Detective converges to that of Opt as Detective progressively learns the users' parameters. *(b)* Users' engagement in flagging: even with low engagement Detective can effectively leverage crowd signals to detect fake news. *(c)* Robustness against spammers: Detective is effective even if the majority of users is adversarial, highlighting the importance of learning about users' flagging accuracy for robustly leveraging crowd signals.

6 EXPERIMENTAL RESULTS

In this section we demonstrate the efficiency of our proposed algorithm for fake news detection in a social network. All reported utilities are normalized by Util(T, Oracle) and all results are averaged over 5 runs.

Learning about users and and exploiting user's flags. In this experiment we compare the average utility, i.e., $\frac{1}{t}$Util(t, Algo) (cf., Equation 1), achieved by the different algorithms at epoch t for $t = 1, \ldots, T$. The results are shown in Figure 2a. We observe that Detective and Opt achieve performance close to that of Oracle. This is impressive, as these algorithms can only use the users' flags and the users' parameters $\{\theta_{u,\bar{f}}, \theta_{u,f}\}_{u \in U}$ (or their beliefs about the users' parameters in case of Detective) to make their predictions. We also observe that the performance of Detective converges to that of Opt as Detective progressively learns the users' parameters. The algorithms No-Learn and Random achieve clearly inferior performance compare to Detective.

Users' engagement in flagging. In this experiment, we vary the engagement $1-\gamma_u$ of the users. We report the utilities Util(T, Algo) in Figure 2b. We observe that with increasing engagement the performance of Detective and Opt improves while the performance of the other shown algorithms is not affected by the increased engagement. Importantly, note that also with a low engagement Detective can effectively leverage crowd signals to detect fake news.

Robustness against spammers. In this experiment we consider only two types of users, i.e., good users and spammers. We vary the fraction of good users relative to the total number of users. We report the utilities Util(T, Algo) achieved by the different algorithms in Figure 2c. We also plot the additional baseline Fixed-CM. Observe that the performance of Fixed-CM degrades with a decreasing fraction of good users. Detective (thanks to learning about users) is effective even if the majority of users is adversarial. This highlights the fact that it is crucial to learn about users' flagging accuracy in order to robustly leverage crowd signals.

7 CONCLUSIONS

In our paper we considered the important problem of leveraging crowd signals for detecting fake news. We demonstrated that any approach that is not learning about users' flagging behaviour is prone to failure in the presence of adversarial/spam users (who want to "promote" fake news). We proposed the algorithm Detective that performs Bayesian inference for detecting fake news and jointly learns about users over time. Our experiments demonstrate that Detective is competitive with the fictitious algorithm Opt, which knows the true users' flagging behaviour. Importantly, Detective (thanks to learning about users) is robust in leveraging flags even if a majority of the users is adversarial. There are some natural extensions for future work. For instance, it would be useful to extend our approach to model and infer the trustworthiness of sources. It would also be important to conduct user studies by deploying our algorithm in a real-world social system.

ACKNOWLEDGMENTS

This work was supported in part by the Swiss National Science Foundation, and Nano-Tera.ch program as part of the Opensense II project, ERC StG 307036, and a Microsoft Research Faculty Fellowship. Adish Singla acknowledges support by a Facebook Graduate Fellowship.

REFERENCES

[1] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *WWW*. 675–684.
[2] Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of thompson sampling. In *NIPS*. 2249–2257.
[3] Liang Chen, Zheng Yan, Weidong Zhang, and Raimo Kantola. 2015. TruSMS: a trustworthy SMS spam control system based on trust management. *Future Generation Computer Systems* 49 (2015), 77–93.
[4] Yuxin Chen, Jean-Michel Renders, Morteza Haghir Chehreghani, and Andreas Krause. 2017. Efficient Online Learning for Optimizing Value of Information: Theory and Application to Interactive Troubleshooting. In *UAI*.
[5] Pern Hui Chia and Svein Johan Knapskog. 2011. Re-evaluating the wisdom of crowds in assessing web security. In *International Conference on Financial Cryptography and Data Security*. 299–314.
[6] Giovanni Luca Ciampaglia, Prashant Shiralkar, Luis M Rocha, Johan Bollen, Filippo Menczer, and Alessandro Flammini. 2015. Computational fact checking

from knowledge networks. *PloS one* 10, 6 (2015), e0128193.

[7] Niall J Conroy, Victoria L Rubin, and Yimin Chen. 2015. Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology* 52, 1 (2015), 1–4.

[8] Nan Du, Le Song, Manuel Gomez-Rodriguez, and Hongyuan Zha. 2013. Scalable Influence Estimation in Continuous-Time Diffusion Networks. In *NIPS*. 3147–3155.

[9] Stuart Ewen. 1998. *PR!: a social history of spin*. Basic Books.

[10] Facebook. 2016. News Feed FYI: Addressing Hoaxes and Fake News. https://newsroom.fb.com/news/2016/12/news-feed-fyi-addressing-hoaxes-and-fake-news/. (December 2016).

[11] Facebook. 2017. Umgang mit Falschmeldungen (Handling of false alarms). https://de.newsroom.fb.com/news/2017/01/umgang-mit-falschmeldungen/. (January 2017).

[12] David Mandell Freeman. 2017. Can You Spot the Fakes?: On the Limitations of User Feedback in Online Social Networks. In *WWW*. 1093–1102.

[13] Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. 2014. Tweetcred: Real-time credibility assessment of content on twitter. In *International Conference on Social Informatics*. Springer, 228–243.

[14] Nguyen Quoc Viet Hung, Duong Chi Thang, Matthias Weidlich, and Karl Aberer. 2015. Minimizing efforts in validating crowd answers. In *SIGMOD*. 999–1014.

[15] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *KDD*. 137–146.

[16] J. Kim, B. Tabibian, A. Oh, B. Schoelkopf, and M. Gomez-Rodriguez. 2018. Leveraging the Crowd to Detect and Reduce the Spread of Fake News and Misinformation. In *WSDM '18: Proceedings of the 11th ACM International Conference on Web Search and Data Mining*.

[17] Srijan Kumar, Robert West, and Jure Leskovec. 2016. Disinformation on the web: Impact, characteristics, and detection of wikipedia hoaxes. In *WWW*. 591–602.

[18] Sejeong Kwon, Meeyoung Cha, and Kyomin Jung. 2017. Rumor detection over varying time windows. *PloS one* 12, 1 (2017), e0168344.

[19] Jure Leskovec and Julian J Mcauley. 2012. Learning to discover social circles in ego networks. In *NIPS*. 539–547.

[20] Yaliang Li, Qi Li, Jing Gao, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2015. On the discovery of evolving truth. In *KDD*. 675–684.

[21] Mengchen Liu, Liu Jiang, Junlin Liu, Xiting Wang, Jun Zhu, and Shixia Liu. 2017. Improving Learning-from-Crowds through Expert Validation. In *IJCAI*. 2329–2336.

[22] Cristian Lumezanu, Nick Feamster, and Hans Klein. 2012. # bias: Measuring the tweeting behavior of propagandists. In *AAAI Conference on Weblogs and Social Media*.

[23] Tyler Moore and Richard Clayton. 2008. Evaluating the wisdom of crowds in assessing phishing websites. *Lecture Notes in Computer Science* 5143 (2008), 16–30.

[24] Ian Osband, Dan Russo, and Benjamin Van Roy. 2013. (More) efficient reinforcement learning via posterior sampling. In *NIPS*. 3003–3011.

[25] Poynter. 2016. International Fact-Checking Network: Fact-Checkers Code Principles. https://www.poynter.org/international-fact-checking-network-fact-checkers-code-principles. (September 2016).

[26] Marian-Andrei Rizoiu, Lexing Xie, Scott Sanner, Manuel Cebrián, Honglin Yu, and Pascal Van Hentenryck. 2017. Expecting to be HIP: Hawkes Intensity Processes for Social Media Popularity. In *WWW*. 735–744.

[27] Victoria L Rubin, Yimin Chen, and Niall J Conroy. 2015. Deception detection for news: three types of fakes. *Proceedings of the Association for Information Science and Technology* 52, 1 (2015), 1–4.

[28] Behzad Tabibian, Isabel Valera, Mehrdad Farajtabar, Le Song, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. 2017. Distilling information reliability and source trustworthiness from digital traces. In *WWW*. 847–855.

[29] William R Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25, 3/4 (1933), 285–294.

[30] Hastagiri P Vanchinathan, Andreas Marfurt, Charles-Antoine Robelin, Donald Kossmann, and Andreas Krause. 2015. Discovering valuable items from massive data. In *KDD*. 1195–1204.

[31] Svitlana Volkova, Kyle Shaffer, Jin Yea Jang, and Nathan Hodas. 2017. Separating Facts from Fiction: Linguistic Models to Classify Suspicious and Trusted News Posts on Twitter. In *ACL*, Vol. 2. 647–653.

[32] Gang Wang, Manish Mohanlal, Christo Wilson, Xiao Wang, Miriam J. Metzger, Haitao Zheng, and Ben Y. Zhao. 2013. Social Turing Tests: Crowdsourcing Sybil Detection. In *NDSS*.

[33] William Yang Wang. 2017. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In *ACL*. 422–426.

[34] Wei Wei and Xiaojun Wan. 2017. Learning to Identify Ambiguous and Misleading News Headlines. In *IJCAI*. 4172–4178.

[35] Shu Wu, Qiang Liu, Yong Liu, Liang Wang, and Tieniu Tan. 2016. Information Credibility Evaluation on Social Media.. In *AAAI*. 4403–4404.

[36] Bo Zhao, Benjamin IP Rubinstein, Jim Gemmell, and Jiawei Han. 2012. A bayesian approach to discovering truth from conflicting sources for data integration. *Proceedings of the VLDB Endowment* 5, 6 (2012), 550–561.

[37] Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. 2015. SEISMIC: A Self-Exciting Point Process Model for Predicting Tweet Popularity. In *KDD*. 1513–1522.

[38] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. 2015. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *WWW*. 1395–1405.

[39] Elena Zheleva, Aleksander Kolcz, and Lise Getoor. 2008. Trusting spam reporters: A reporter-based reputation system for email filtering. *TOIS* 27, 1 (2008), 3.