QuickStop: A Markov Optimal Stopping Approach for Quickest Misinformation Detection

HONGHAO WEI, Arizona State University
XIAOHAN KANG, University of Illinois at Urbana-Champaign
WEINA WANG, Carnegie Mellon University
LEI YING, Arizona State University

This paper combines data-driven and model-driven methods for real-time misinformation detection. Our algorithm, named QuickStop, is an optimal stopping algorithm based on a probabilistic information spreading model obtained from labeled data. The algorithm consists of an offline machine learning algorithm for learning the probabilistic information spreading model and an online optimal stopping algorithm to detect misinformation. The online detection algorithm has both low computational and memory complexities. Our numerical evaluations with a real-world dataset show that QuickStop outperforms existing misinformation detection algorithms in terms of both accuracy and detection time (number of observations needed for detection). Our evaluations with synthetic data further show that QuickStop is robust to (offline) learning errors.

CCS Concepts: • **Human-centered computing → Social network analysis**; • **Computing methodologies** → *Machine learning*.

Additional Key Words and Phrases: fake news; social networks; quickest detection; misinformation detection

ACM Reference Format:
Honghao Wei, Xiaohan Kang, Weina Wang, and Lei Ying. 2019. QuickStop: A Markov Optimal Stopping Approach for Quickest Misinformation Detection. *Proc. ACM Meas. Anal. Comput. Syst.* 3, 2, Article 41 (June 2019), 25 pages. https://doi.org/10.1145/3326156

1 INTRODUCTION

The proliferation of misinformation [20] (colloquially known as "fake news") on online social networks has become one of the greatest threats to our national security, has eroded the public trust in news media, and is an imminent threat to the ecosystem of online social platforms like Facebook, Twitter and Sina Weibo. For example, in 2013, a fake tweet claiming that the then US President Barack Obama was injured by explosives from a hacked Twitter account of the Associated Press caused a 150-point drop of the Dow Jones in just two minutes;[1] and fake news in the 2016 US Presidential Election has led to increased political and social polarization and posed a great threat to democracy. Social media companies, such as Facebook and Twitter, are now taking multiple

[1]https://www.washingtonpost.com/news/worldviews/wp/2013/04/23/syrian-hackers-claim-ap-hack-that-tipped-stock-market-by-136-billion-is-it-terrorism/

Authors' addresses: Honghao Wei, Arizona State University, Tempe, Arizona, 85287, hwei30@asu.edu; Xiaohan Kang, University of Illinois at Urbana-Champaign, Urbana, Illinois, 61820, Veggente@gmail.com; Weina Wang, Carnegie Mellon University, Pittsburgh, Pennsylvania, 15213, weinaw@cs.cmu.edu; Lei Ying, Arizona State University, Tempe, Arizona, 85287, lei.ying.2@asu.edu.

2476-1249/2019/6-ART41 $15.00
https://doi.org/10.1145/3326156

countermeasures to combat misinformation as the proliferation of misinformation is driving users away from these platforms.

Despite the enormous attention it receives and the tremendous efforts from both public and private institutions to counter it, misinformation detection remains a daunting task as of today. Online platforms and news organizations have experimented different methods. Facebook launched its fact-checking project in Spring 2018 to work with third-party publishers to validate facts and accuracy of news articles.[2] The New York Times has recently published a tip form so that its readers can report misinformation and fake news.[3] The third-party fact-checking method is often very effective for detecting whether a specific news article is fake or not, but clearly is not a scalable solution and cannot cover even a tiny fraction of news articles and tweets (there are about 500 million tweets per day on Twitter). The crowdsourcing approach used by New York Times is more scalable, but the reports are not always trustworthy because anyone can send a tip. In light of these challenges, machine-learning and data-mining approaches have emerged to tackle misinformation detection in a systematic way (see [36] for a comprehensive review). It has been shown in [4] that the features extracted from the content of a news article, the features of the users who spread the news, and the connections of these users can be effectively utilized for misinformation detection. These are exciting discoveries and progresses because "machine-based" methods are much more scalable than "human-based" methods, and can handle a vast number of news articles in a short period of time.

While machine-learning approaches address the scalability issue, another important aspect of misinformation detection, *speed* or *sample complexity* (the amount of time or the number of observations needed to detect misinformation), has yet to be tackled. Speed is important because of the disruptive nature of misinformation, which often causes significant damages in a very short period of time. For example, it only took less than two minutes to tip the Dow Jones by 150 points with one single fake tweet. Therefore, it is imperative to detect misinformation at the earliest time so that proper countermeasures can be taken to suppress it. A fact-checking approach may take a few hours because fact-checkers need to gather facts and evidence to validate or invalidate a news article. Therefore, the speed aspect of misinformation detection is equally important as accuracy and scalability in the design of misinformation detection algorithms.

Motivated by the discussions above, this paper focuses on quickest detection of misinformation. The goal is to develop an algorithm that addresses the three important considerations in misinformation detection: scalability, accuracy and *speed*. Note that existing machine-learning-based approaches have demonstrated a strong correlation between user features and the spreading models under different information types (real or fake). We will demonstrate this strong correlation in Section 2 using a Sina Weibo dataset. The signal of a single retweet is often very weak and usually not sufficient for classifying a news article with a reasonable accuracy. But this accuracy can be improved with more and more weak signals. This paper views the problem of misinformation detection as a *sequential* hypothesis testing problem. As the platform receives a *sequence* of weak signals in real time, it determines whether it has collected enough information to declare the type of the news (real or fake). The more signals collected, the more accurate the detection result will be, but then we are at risk of letting the misinformation spread. Enlightened by these observations, we propose QuickStop, a scalable algorithm that performs accurate, quick detection of misinformation. QuickStop combines a data-driven approach with a model-driven approach in the following way.

- **Data-based probabilistic modeling**: Since each retweet is a weak signal for the hypothesis testing (whether the news article is real or fake), extracting the statistics of these weak

[2]https://www.facebook.com/help/1952307158131536?helpref=faq_content
[3]https://www.nytimes.com/2018/09/17/technology/disinformation-tipsheet.html

signals is important for establishing an effective probabilistic model for hypothesis testing. QuickStop first uses an SVM (Support Vector Machine) algorithm to extract an edge-based probabilistic information spreading model. Section 2 explains the rationale behind the edge-based model (compared with a node-based model) and shows the effectiveness using the Sina Weibo dataset.

- **Model-based quickest detection**: After establishing the probabilistic model, we formulate the quickest misinformation detection problem as an *optimal stopping problem*. Specifically, we propose a cost model that includes both the cost due to detection error and the cost due to the propagation of misinformation. Note that the propagation cost occurs only in the case of misinformation. With this formulation, the goal is to discover a *stopping policy*, i.e., a policy that determines when to stop collecting observations and what type to declare after stopping, that minimizes the overall cost. As more observations are collected, the error cost decreases but the propagation cost could increase in the case of misinformation. Therefore, the optimal stopping policy needs to balance the detection accuracy and detection time so that misinformation can be detected confidently at the earliest possible time.

The main contributions of this paper are summarized below.

- **Problem Formulation**: We formulate the quickest misinformation detection problem as a Markov optimal stopping problem based on a probabilistic information spreading model. This probabilistic model can be extracted from training datasets by given classifiers. An interesting feature of our formulation is the asymmetric cost functions between real news and misinformation — spreading misinformation causes far more damage than spreading real news so we need to act quickly only in the case of misinformation. The analytical solution of a Markov optimal stopping problem in general requires computing a function of the state (see, e.g., Chapter 3.4.4 on Page 59 of [30]). Since the state in our formulation includes the current time index (i.e., how many users the article has reached), this function would be *time-dependent*. Effectively, this means that we potentially need a different function of the collected information for each time step. However, utilizing structures in our probabilistic model, we show that the optimal stopping policy has a simple threshold form described by several *time-independent* thresholds. We comment that this structure is similar to that in the solution of the sequential testing problem, but the techniques there do not directly apply to our problem since our cost function has a nonlinear term due to the asymmetry.
- **Algorithm and Analysis**: We propose an algorithm named QuickStop that detects misinformation based on *edge* types, where an edge is a connection between two individuals along which a piece of information spreads from one individual to the other. QuickStop consists of two parts: (i) QuickStop-Training, an offline algorithm that classifies edges into four types and then calculates transition probabilities between different edge types, where the transition probabilities in the case of real news may be different from those for misinformation; and (ii) QuickStop-Detection, an online detection algorithm with low computational and memory complexities. We emphasize that the main computation load is in the offline part. Once the offline training is completed, the online part for detection is very efficient as described below. QuickStop-Detection maintains a scalar variable that describes the current state, and updates the state for each new observation. The update just follows a simple formula and its complexity does not depend on how many observations have been collected. Then the algorithm compares the state with several thresholds calculated offline. Based on the comparison result, it decides whether it will keep collecting observations or declare the type of the information. In the latter case, what type to declare is also determined by the

comparison result. Therefore, QuickStop-Detection has very low computational and memory complexities, and is ideal for real-time large-scale misinformation detection.

- **Evaluations**: We evaluated the performance of QuickStop using both a real-world social network dataset (from Sina Weibo) and synthetic data. The evaluations on the real-world dataset demonstrates the effectiveness of our algorithm in terms of both accuracy and speed compared with state-of-the-art real-time misinformation algorithms. Under QuickStop with a low propagation cost, it took 12 observations on average in the Weibo dataset to detect misinformation, but more to declare real news. This is consistent with the asymmetric cost model. Furthermore, the false negative rate (misinformation classified as real news) is much lower than the false positive rate (real news classified as misinformation), which is also desirable in practice. In contrast, the accuracy of the state-of-the-art early detection algorithms are still lower than ours even with 33× more observations. From the evaluations on synthetic data, we further observed that QuickStop is robust to classification errors.

We finally comment that while several early misinformation detection algorithms have been developed [6, 23–26, 48], these algorithms either use a fixed number of observations as input [6, 26] or observations over a fixed time period as input [23–25, 48]. Therefore, these early detection algorithms do not minimize the detection time (or the number of observations) in real time. Our approach, on the other hand, tackles the problem using the optimal stopping method and optimizes the number of observations needed in real-time for quickest detection. Our numerical evaluations show QuickStop achieves higher accuracy with fewer observations due to the dynamic nature of the algorithm. A detailed review of other related work is presented in Section 7.

2 MODEL AND PROBLEM STATEMENT

We model an online social network as a graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices representing users and $\mathcal{E}$ is the set of directed edges representing the connections between users. Information (real news or misinformation) can spread from one user to another via the edge connecting them, e.g., a Twitter user can retweet a post from one of her/his followees. In this paper, we adopt the terminology of Twitter. Given a directed edge (v, u), user u is called a follower of user v; and user v is called a followee of user u. Information can spread from user v to user u via this directed edge.

We assume two types of information that may spread in the network: *real news articles* (simply called *news* in the remainder of the paper) and *misinformation*. A user (say user u) decides whether to post (retweet) the information based on the following three factors: (i) the type of the information, (ii) the features of user u, and (iii) the set of user u's neighbors who have posted (retweeted) the information before user u.

As information spreads in the network, the platform obtains sequential observations (weak signals) for misinformation detection. In this paper, a retweet is considered to be an observation, which is represented by the edge over which this retweet occurs. Specifically, we define the kth observation to be $(\mathbf{V}_k, \mathbf{U}_k)$, where $\mathbf{U}_k$ is the feature vector of the kth user who retweets the information and $\mathbf{V}_k$ is the feature vector of the followee from whom the kth user retweets the information. We remark that when complete network and information diffusion information is known, the information spreading trace is likely to be a tree or a forest (with multiple information sources). However, in practice, it is often not the case because of missing information and partial observations [15, 18]. Therefore, the observations we have are a sequence of retweets $(\mathbf{V}_k, \mathbf{U}_k)$, which not necessarily form a tree. In particular, $\mathbf{U}_k$ is not necessarily the same as $\mathbf{V}_{k+1}$ in the trace. Now to model these retweets as weak signals, we can consider the following two approaches.

- User-based Model: In the user-based model, given the type of an article, the probability a user retweets the article depends on the features of the user. Intuitively, an honest user has

a lower probability to retweet some misinformation than a malicious user (e.g., a bot). The user-based model is to classify the users based on the user features with a labelled training dataset.

- Edge-based Model: In the edge-based model, we view each edge as a communication channel and classify edges into different groups. For example, misinformation is more likely to spread over an edge between two malicious socialbots than an edge between two honest users. The edge-based model is to classify the edges based on the edge features (the feature vectors of the two end users $(\mathbf{V}, \mathbf{U})$) with a labelled training dataset.

Figure 1 presents the distributions of SVM classification scores of the user-based model and the edge-based model of the Weibo dataset released in [24], where x-axis is the classification score of the SVM classifier, and y-axis is the score distributions (frequencies). A user or an edge with a higher score is considered more likely to spread misinformation. From the figure, we first observe that the scores of users (or edges) involved in spreading news concentrate around zero while the scores of users (or edges) involved in spreading misinformation concentrate around one. This demonstrates a strong correlation between article types and user/edge features. Furthermore, we can see that the score distributions based on edges exhibit a stronger correlation with article types than the score distributions based on users. For example, for misinformation, the score distribution based on edges has a higher frequency around zero than that based on users (60% versus 45%). Because of this observation, in this paper, we use the edge-based model.

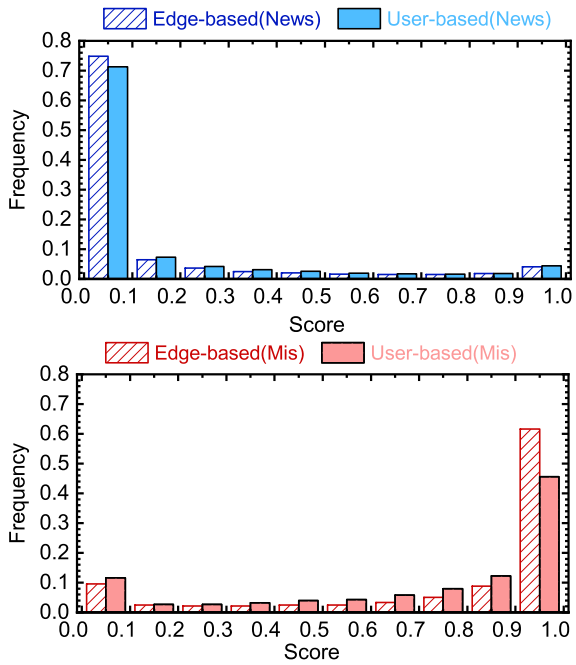

Fig. 1. Classification distribution

We assume that given the article type, the sequential observations form a Markov chain as shown in Figure 2, where we further assume the edge feature vector $(\mathbf{V}, \mathbf{U})$ can be classified into four classes $Z_k = f(\mathbf{V}_k, \mathbf{U}_k) \in \{0, 1, 2, 3\}$ to simplify the model, where 0 is the type of edges that are

most likely to be used for spreading news and 3 is the type of edges that are most likely to be used for spreading misinformation. Under this Markov chain model, besides the edge types, additional parameters to be learned are the transition probabilities, denoted by $\alpha_i(Z_k|Z_{k-1})$, where $i \in \{0, 1\}$, $i = 0$ indicates these are the transition probabilities when spreading news, and $i = 1$ indicates these are the transition probabilities when spreading misinformation. When $\alpha_0(\cdot|\cdot)$ and $\alpha_1(\cdot|\cdot)$ are different, we can detect misinformation using sequential hypothesis testing. We remark that the generalization of the four-class model to a C-class model for a finite C is straightforward. Our main results and the proposed algorithm work for any finite C. The choice of the number of classes, C, however, needs to balance the detection performance, which favors a larger C, and the training complexity and accuracy, which often favor a smaller C. We adopt the four-class model based on experimental evaluations on the Weibo dataset. The evaluations showed that the four-class model performs significantly better than a two-class model, but increasing C from four to eight did not yield any noticeable improvement.

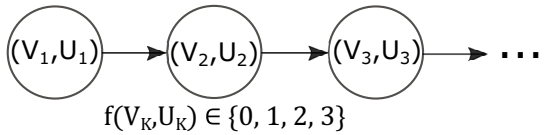

Fig. 2. A Markov chain model for sequential observations

Tables 1 and 2 show the empirical transition probability matrices under news and misinformation obtained from the Weibo dataset (an edge with a classification score below 0.25 is placed in class 0, one with score between 0.25 and 0.5 is placed in class 1, one with score between 0.5 and 0.75 is placed in class 2, and one with score above 0.75 is placed in class 3). We can clearly observe that the observations are not i.i.d., which supports our edge-based Markovian model.

	0	1	2	3
0	0.828	0.120	0.039	0.012
1	0.651	0.224	0.084	0.041
2	0.500	0.193	0.191	0.116
3	0.279	0.181	0.211	0.329

Table 1. Edge Transition Probability Matrix under News from the Weibo Dataset

	0	1	2	3
0	0.163	0.167	0.249	0.421
1	0.105	0.194	0.239	0.461
2	0.080	0.119	0.277	0.524
3	0.052	0.088	0.203	0.657

Table 2. Edge Transition Probability Matrix under Misinformation from the Weibo Dataset

For the edge classifier, we leverage the existing research, in particular, the research in [18], where it shows that SVM performs the best among several popular machine-learning algorithms, including decision tree and random forest for classifying misinformation. We adopt SVM and the user features proposed in [4] to obtain an edge classifier. The details can be found in Section 4. After

classifying the edges in the training data, we further obtain transition probabilities $\alpha_i(Z_k|Z_{k-1})$ from the training data to build a probabilistic information spreading model (details can be found in Section 3).

Our focus is on the quickest detection formulation after training the edge classifier and learning the transition probabilities $\alpha_i(Z_k|Z_{k-1})$. In the next section, we will formulate the quickest misinformation detect problem and prove that the problem is a Markov optimal stopping time problem and its solution is a time-invariant threshold policy. Furthermore, the thresholds can be efficiently calculated offline based on the probabilistic model. The online algorithm is of constant computational and memory complexities, and is very easy to implement.

3 OPTIMAL STOPPING APPROACH FOR QUICKEST MISINFORMATION DETECTION

Consider an online social network platform that is monitoring the spread of some information in the network. We say that an event occurs when a user retweets or posts the information. When the kth event occurs, we obtain an observation $Z_k \in \{0, 1, 2, 3\}$ by using the trained classifier to learn the edge type. Furthermore, we assume that we have learned the transition probabilities $\alpha_i(Z_k|Z_{k-1})$ from training data.

With the model introduced above, the detection of misinformation can be formulated as a hypothesis testing problem with the following two hypotheses:

- H_0: The information is news. In this case, $\{Z_k\}$ is a four-state Markov process with transition probabilities $\alpha_0(Z_k|Z_{k-1})$.
- H_1: The information is misinformation. Then $\{Z_k\}$ is a four-state Markov process with transition probabilities $\alpha_1(Z_k|Z_{k-1})$.

Given observations $\{Z_k\}$, the misinformation detection problem is to determine whether H_0 or H_1 is true. We assume that in terms of the prior distribution, hypothesis H_0 occurs with probability π_0 and H_1 occurs with probability $\pi_1 = 1 - \pi_0$. We assume the first observation Z_1 is uniformly distributed over $\{0, 1, 2, 3\}$ regardless of the hypothesis.

Now define

$$\Pi_k = \Pr\left(H_1|(Z_1, Z_2, \cdots, Z_k)\right), \tag{1}$$

so

$$1 - \Pi_k = \Pr\left(H_0|(Z_1, Z_2, \cdots, Z_k)\right).$$

According to the Bayes rule, we have

$$\begin{aligned}
\Pi_k &= \Pr\left(H_1|(Z_1, Z_2, \cdots, Z_k)\right) \\
&= \frac{\Pr\left((Z_1, Z_2, \cdots, Z_k)|H_1\right)\Pr(H_1)}{\Pr(Z_1, Z_2, \ldots, Z_k)} \\
&= \frac{\pi_1 \prod_{i=1}^{k-1} \alpha_1(Z_{i+1}|Z_i)}{(1-\pi_1)\prod_{i=1}^{k-1} \alpha_0(Z_{i+1}|Z_i) + \pi_1 \prod_{i=1}^{k-1} \alpha_1(Z_{i+1}|Z_i)}.
\end{aligned}$$

From the equation above, we have

$$\frac{1-\Pi_k}{\Pi_k} = \frac{(1-\pi_1)\prod_{i=1}^{k-1} \alpha_0(Z_{i+1}|Z_i)}{\pi_1 \prod_{i=1}^{k-1} \alpha_1(Z_{i+1}|Z_i)},$$

which implies that

$$\frac{1-\Pi_{k+1}}{\Pi_{k+1}} = \frac{1-\Pi_k}{\Pi_k}\frac{\alpha_0(Z_{k+1}|Z_k)}{\alpha_1(Z_{k+1}|Z_k)}.$$

Therefore, we have the following recursive equation

$$\Pi_{k+1} = \frac{\Pi_k \alpha_1(Z_{k+1}|Z_k)}{(1-\Pi_k)\,\alpha_0(Z_{k+1}|Z_k) + \Pi_k \alpha_1(Z_{k+1}|Z_k)}. \tag{2}$$

for updating our belief on H_1.

Note that given the observation sequence $\{Z_k\}$, we can calculate $\frac{1-\Pi_k}{\Pi_k}$ in real time. The question is when to declare the type of the information. The more observations we have, the more accurate the decision would be but the more widely the information would have spread. Therefore, we need to balance the accuracy and the potential damage of spreading misinformation. Let $T \geq 1$ denote the random time at which the type of information is declared, which is a function of $Z_1, Z_2, \cdots, Z_T$; i.e., T is a *stopping time* with respect to $\{Z_k\}$. Let δ_T denote the type of information that is declared by a detection algorithm. We consider the following two types of costs in the misinformation detection problem.

Error Cost

The first type of cost comes from mis-detection. Let c_{I} denote the cost of type-I error (also called false positive, where news is declared as misinformation) and c_{II} denote the cost of type-II error (also called false negative, where misinformation is declared as news). The expected cost of mis-detection is

$$c_e(\delta_T) = c_{\mathrm{I}} \Pr(\delta_T = 1|H_0)(1-\pi_1) + c_{\mathrm{II}} \Pr(\delta_T = 0|H_1)\pi_1,$$

where π_1 is the prior probability of H_1.

Propagation Cost

The other type of cost is the propagation cost. Information becomes more influential when more people share it. So we need to detect misinformation as quickly as we can to limit its potential damage, while spreading news does not occur any cost. Consequently, the propagation cost in our model is asymmetric and comes only from misinformation. In particular, we assume that there is a cost of c associated with each time slot of propagation if the information is misinformation. Thus, at the stopping time T, the propagation cost is

$$E\left[cT\mathbb{I}_{H_1}\right],$$

where $\mathbb{I}_{H_1}$ is the indicator function which is equal to 1 when H_1 is true and is equal to 0 when H_0 is true.

A Markov Optimal Stopping Approach

The goal of the misinformation detection algorithm is to minimize the overall cost. Formally, we aim to find a stopping time T and a decision rule δ_T, both depending on $Z_1, \cdots, Z_T$, that solve the following problem

$$\inf_{T,\delta_T} c_e(\delta_T) + E\left[cT\mathbb{I}_{H_1}\right]. \tag{3}$$

An important step for solving this problem is to properly handle the propagation cost term $E\left[cT\mathbb{I}_{H_1}\right]$, which depends on the hypothesis. Note that if this term were $E[cT]$, this problem would be the same as the renowned *sequential testing problem* (see, e.g., [30]). Specifically, the sequential testing problem solves

$$\inf_{T,\delta_T} c_e(\delta_T) + E[cT]. \tag{4}$$

Recall that given $\mathbb{I}_{H_1}$, the observation sequence $\{Z_k\}$ is a Markov chain. Therefore, when we view $\mathbb{I}_{H_1}$ as part of the state, $\{(\mathbb{I}_{H_1}, Z_k)\}$ forms a Markov chain, and thus the formulation (3) is a Markov

optimal stopping problem. However, this Markov chain is only *partially observable* since we cannot observe $\mathbb{I}_{H_1}$. We can transform this optimal stopping problem of a partially observable Markov chain to a fully observable optimal stopping problem. Specifically, consider the conditional distribution of $\mathbb{I}_{H_1}$ given observations $Z_1, Z_2, \ldots, Z_k$. This conditional distribution can be represented by the variable Π_k defined in (1). We can verify that $\{(\Pi_k, Z_k)\}$ is a Markov chain. For the convenience of analysis, we also view the time index k as part of the state and consider the Markov chain $\{(\Pi_k, Z_k, k)\}$. With this, we transform the optimal stopping problem in (3) to a Markov optimal stopping problem in Theorem 3.1.

Theorem 3.1. *The optimal stopping problem* (3) *is equivalent to a Markov optimal stopping problem with respect to the Markov chain* $\{(\Pi_k, Z_k, k)\}$. *Formally,*

$$\begin{aligned} &\inf_{T,\delta_T} c_e(\delta_T) + E\left[cT\mathbb{I}_{H_1}\right] \\ &= \inf_{T\in\mathcal{T}} E\left[\min\{c_{\text{II}}\Pi_T, c_{\text{I}}(1-\Pi_T)\} + cT\Pi_T\right], \end{aligned} \tag{5}$$

where $\mathcal{T}$ *is the set of stopping times with respect to* $\{(\Pi_k, Z_k, k)\}$. □

Note that the variable in the Markov stopping problem (5) is just the stopping time T instead of both T and δ_T. Therefore, we can find the optimal stopping policy in two steps: first find the optimal stopping time T by solving (5), and then find the optimal decision rule δ_T based on $Z_1, \ldots Z_T$. Such a transform from a partially observable Markov chain to a fully observable Markov chain has been widely used in optimal stopping problems and more generally in Markov decision processes (see, e.g., Chapter 4.1 in Vol. I of [3], and [46]). Here we include the proof of Theorem 3.1 in Appendix A.1 for completeness.

The analytical solution of the Markov optimal stopping problem (5) can be obtained using the Snell envelope (see, e.g., Chapter 2.2 on Page 38 of [19], and Chapter 3.4.4 on Page 59 of [30]), which, in general, needs to compute a function of the state and store the function for use in the optimal stopping policy. In our problem, the state includes the time index k. Then to compute the Snell envelope, we potentially need a different function of the collected information Π_k and Z_k for each nonnegative integer k. Interestingly, in Theorem 3.2, we will see that for our problem, the optimal stopping policy is a threshold policy on Π_k described by 8 *time-independent* thresholds. This time-independence property greatly simplifies the computation. The requirement on memory storage is also minimal, so this policy will be very simple to implement. We comment that compared with the sequential testing problem, the cost function in our problem has a non-linear term $cT\Pi_T$. So the proof for the sequential testing problem does not directly apply to our problem. Nevertheless, we utilize an essential observation that the process $\{\Pi_k\}$ is a martingale with respect to $\{Z_k\}$ and still obtain a time-independent threshold policy. The proof of Theorem 3.2 is presented in Appendix A.2.

Theorem 3.2. *The optimal stopping time* T^* *is*

$$T^* = \inf_{k>0}\left\{k : \Pi_k \notin \left(\pi_l^{(Z_k)}, \pi_u^{(Z_k)}\right)\right\}. \tag{6}$$

In other words, there exist positive values $\pi_l^{(z)}$, $\pi_u^{(z)}$, $(z \in \{0,1,2,3\})$, *independent of* T, *such that the algorithm declares the information to be news when* $Z_k = z$ *and* $\Pi_k \le \pi_l^{(z)}$, *and declares the information to be misinformation when* $Z_k = z$ *and* $\Pi_k \ge \pi_u^{(z)}$. *The thresholds* $\pi_l^{(z)}$ *and* $\pi_u^{(z)}$ *for*

$z = 0, 1, 2, 3$ are determined by solving the following equations:

$$\pi_l^{(z)} = \sup_{\pi} \left\{ 0 \le \pi \le \frac{c_{\mathrm{I}}}{c_{\mathrm{I}} + c_{\mathrm{II}}} \middle| s^{(z)}(\pi) = c_{\mathrm{II}}\pi \right\} \tag{7}$$

$$\pi_u^{(z)} = \inf_{\pi} \left\{ \frac{c_{\mathrm{I}}}{c_{\mathrm{I}} + c_{\mathrm{II}}} \le \pi \le 1 \middle| s^{(z)}(\pi) = c_{\mathrm{I}}(1-\pi) \right\} \tag{8}$$

where s is the solution of the Bellman equation below

$$s^{(z)}(\pi) = \min \left\{ g(\pi), E\left[s^{(Z_{k+1})}(\Pi_{k+1}) \middle| \Pi_k = \pi, Z_k = z \right] + c\pi \right\} \tag{9}$$

and

$$g(\pi) = \min\{c_{\mathrm{II}}\pi, c_{\mathrm{I}}(1-\pi)\}.$$

□

Note $E\left[s^{(Z_{k+1})}(\Pi_{k+1}) \middle| \Pi_k = \pi, Z_k = z \right]$ in (9) is understood as the expected cost to go starting from the next time step based on s given the state in the current time step is π and z. In other words,

$$\begin{aligned} & E\left[s^{(Z_{k+1})}(\Pi_{k+1}) \middle| \Pi_k = \pi, Z_k = z \right] \\ = & \sum_{z'=0}^{3} s^{(z')} \left(\frac{\pi\alpha_1(z'|z)}{\pi\alpha_1(z'|z) + (1-\pi)\alpha_0(z'|z)} \right) (\pi\alpha_1(z'|z) + (1-\pi)\alpha_0(z'|z)) . \end{aligned}$$

So it does not depend on k.

4 QUICKSTOP: THE QUICKEST MISINFORMATION DETECTION ALGORITHM

From the results presented in the previous sections, we propose QuickStop, which includes the following components.

- **Training data:** Our algorithm needs labeled training data. The dataset should include a set of information spreading traces which are labeled as news or misinformation. Each user involved in the information trace has a feature vector. The information should also include the followee from whom a user retweeted the information.
- **Learning the information spreading model via the SVM classifier:** Given the labeled data, we first train an SVM classifier with the dataset that classifies information to news or misinformation. The input to the SVM classifier is the average feature vector of edges. Recall that the feature vector of edge (v, u) is $(\mathbf{V}, \mathbf{U})$. After training the SVM classifier, we use the classifier to classify the edges into four groups based on the edge feature vector. Note that SVM outputs an value between 0 to 1. In our experiments, we use the following mapping: $[0, 0.25] \Rightarrow 0$, $(0.25, 0.5] \Rightarrow 1$, $(0.5, 0.75] \Rightarrow 2$, and $(0.75, 1] \Rightarrow 3$. From the transition probabilities learned from the previous step, we calculate $\pi_l^{(z)}$ and $\pi_u^{(z)}$ according to Theorem 3.2.
- **Quickest detection:** When monitoring information spreading, the algorithm updates Π_k according to (2) when an event occurs, where we set $\Pi_1 = \pi_1$ which is the prior distribution of hypothesis H_1 according to the data. The information is declared to be news when $\Pi_k \le \pi_l^{(Z_k)}$ and misinformation when $\Pi_k \ge \pi_u^{(Z_k)}$.

We remark that this algorithm combines a data-driven approach, which learns the underlying probabilistic model of information spreading in networks, and a model-driven approach, which identifies misinformation in a timely manner with the quickest detection formulation.

QuickStop consists of two parts: QuickStop-Training and QuickStop-Detection, whose pseudo-code can be found in Algorithms 1 and 2, respectively.

4.1 Computational and Memory Complexities

In the training part, we use an SVM classifier on n information traces. In SVM, the feature space is obtained by using some mapping functions and the hyperplane is determined by a set of support vectors. Then the dimension of the feature space depends on the mapping function. The minimum computational complexity of training an SVM is $O(n^2)$, and may reach $O(n^3)$.

The thresholds are calculated using the value iteration method. Let ϵ be the quantization step size of the state Π_k. During the value iteration, the terminal time depends on the quantization precision. The computational complexity for each iteration is $O(\frac{1}{\epsilon})$; the memory complexity is also $O(\frac{1}{\epsilon})$. This step is done offline.

For the online misinformation detection part, the computational complexity per iteration and memory complexity are both $O(1)$. The algorithm needs to store 8 threshold values and 32 transition probabilities. Each update of the state Π_k only requires a few elementary operations.

5 PERFORMANCE EVALUATION WITH REAL-WORLD DATASETS

We first evaluate the performance of QuickStop using the following real-world dataset.

The Weibo Dataset: Sina Weibo is a Chinese microblogging website similar to Twitter. The Weibo dataset we use is the one released in [23], which includes 4,664 labeled information traces from Sina's community management center.[4] The dataset also includes user information such as the number of followees, the number of followers, the registration days, etc, which are used as user features in our algorithm. We remove information traces whose sizes are small. In particular, we keep the traces in which the information was retweeted by the followers of at least 50 distinct users. We further balance the dataset by selecting 488 news traces and 488 misinformation traces. The average retweets per trace is 2,031, the largest trace includes 55,155 retweets, and the smallest one has 105 retweets. We used 80% of the traces as training data and the remaining as the testing data.

We compared QuickStop with the following misinformation detection algorithms aiming at early detection: (i) decision-tree-based methods [4]; (ii) SVM-based methods with RBF kernel [45]; (iii) linear SVM-based models for time-series data [24]; (iv) Neural network-based methods with Recurrent Neural Networks (RNNs), or Convolutional Neural Networks (CNNs), or both for sequential data [22]; and (v) a comprehensive approach involving RNNs, Feedforward Neural Networks (FNNs), and singular value decomposition (SVD) for low-dimensional feature representation [32]. Note that all these methods are feature-based classification algorithms. The first three algorithms [4, 24, 45] can take both user features and news content features as input. The algorithm proposed in [22] has three versions, RNN only, CNN only, and both. The algorithms use the sequential user features as the input to the neural networks. The algorithm in [32] uses an RNN to extract article features, an FNN to extract user features, and another FNN to integrate both user and article features for classification. QuickStop, on the other hand, only uses user features. In the evaluations, for the first three algorithms, we implemented two versions: one with only user features (i.e., the same set of user features used in QuickStop), and the other with both user and content features (so more features than QuickStop). The ten different algorithms are summarized below.

- **DTC$_u$:** A Twitter information credibility method [4] based on decision trees, with only user features.
- **DTC$_a$:** A Twitter information credibility method [4] based on decision trees, with both user and content features.

[4]https://service.account.weibo.com

Algorithm 1 QuickStop-Training (Offline)

Input:

A set of information traces: $E = \{e_1, e_2, \cdots, e_n\}$ ▷ e_i is a sequence of users: $\{u_t^{(e_i)}\}$, where t is the posting order of a user, i is the index of the news trace

A set of labels: $\mathbf{l} = \{l_1, l_2, \cdots, l_n\}$ ▷ $l_i \in \{0, 1\}$ is the label of e_i (0: news, 1: misinformation).

For the tth user who post the information i (say user $u_t^{(e_i)}$), obtain feature vector of the edge: $(\mathbf{V}_t^{(e_i)}, \mathbf{U}_t^{(e_i)})$.

Compute $\tilde{\mathbf{U}}^{(e_i)} = \frac{1}{|e_i|-1}\left(\sum_{t=2}^{|e_i|}\mathbf{V}_t^{(e_i)}, \sum_{t=2}^{|e_i|}\mathbf{U}_t^{(e_i)}\right)$ ▷ $|e_i|$ is the cardinality of news trace e_i

Train edge classifier: $f(\cdot)$ using SVM with training dataset $(\tilde{\mathbf{U}}, \mathbf{l})$

Classify edges in the traces, $Z_t^{(e_i)} \leftarrow f(\mathbf{V}_t^{(e_i)}, \mathbf{U}_t^{(e_i)})$

Calculate the transition probabilities

$$\alpha_j(z_1|z_2) = \frac{\sum_{i=1}^{n}\sum_{t=1}^{|e_i|-1}\mathbb{I}_{\{Z_{t+1}^{(e_i)}=z_1, Z_t^{(e_i)}=z_2\}}\mathbb{I}_{\{l_i=j\}}}{\sum_{i=1}^{n}\sum_{t=1}^{|e_i|-1}\mathbb{I}_{\{Z_t^{(e_i)}=z_2\}}\mathbb{I}_{\{l_i=j\}}}, z_1, z_2 \in \{0, 1, 2, 3\}$$

Initialize $\epsilon, \epsilon_0, m \leftarrow \frac{1}{\epsilon}, \pi = \{\pi^1, \cdots, \pi^m\}, c_{\mathrm{I}}, c_{\mathrm{II}}, c$ ▷ ϵ and ϵ_0 specify the quantization step size and the convergence tolerance

for $z = 0, 1, 2, 3$ **do**

  $s_0^{(z)}(\pi^i) \leftarrow \min\{c_{\mathrm{II}}\pi^i, c_{\mathrm{I}}(1-\pi^i)\}, i = 1, \ldots, m$

end for

for $j = 1, 2, \ldots$ **do** ▷ Solve the Bellman equation using value iteration

  $g(\pi^i) = \min\{c_{\mathrm{II}}\pi^i, c_{\mathrm{I}}(1-\pi^i)\}, i = 1, \ldots, m$

  for $z = 0, 1, 2, 3$ **do**

    $s_1^{(z)}(\pi^i) \leftarrow \min\left\{g(\pi^i), E\left[s_0^{(\tilde{z})}(\tilde{\pi})|\pi^i, z\right] + c\pi^i\right\}, i = 1, \ldots, m$ where

$$E\left[s_0^{(\tilde{z})}(\tilde{\pi})|\pi^i, z\right] = \sum_{k=0}^{3} s_0^{(k)}\left(\frac{\pi^i\alpha_1(k|z)}{\pi^i\alpha_1(k|z) + (1-\pi^i)\alpha_0(k|z)}\right) \times (\pi^i\alpha_1(k|z) + (1-\pi^i)\alpha_0(k|z))$$

  end for

  if $\|s_1(\pi) - s_0(\pi)\| \le \epsilon_0$ **then**

    break

  else

    $s_0(\pi) \leftarrow s_1(\pi)$

  end if

end for

$\pi_l^{(z)} \leftarrow \sup_{\pi^i}\left\{0 \le \pi^i \le \frac{c_{\mathrm{I}}}{c_{\mathrm{I}}+c_{\mathrm{II}}} \middle| s_0^{(z)}(\pi^i) = c_{\mathrm{II}}\pi^i\right\}, z \in \{0, 1, 2, 3\}$

$\pi_u^{(z)} \leftarrow \inf_{\pi^i}\left\{\frac{c_{\mathrm{I}}}{c_{\mathrm{I}}+c_{\mathrm{II}}} \le \pi^i \le 1 \middle| s_0^{(z)}(\pi^i) = c_{\mathrm{I}}(1-\pi^i)\right\}, z \in \{0, 1, 2, 3\}$ ▷ Compute thresholds

Output:

Edge classifier: $f(\cdot)$

Transition probabilities: $\alpha_i(\cdot),\ i = 0, 1$

Thresholds $\pi_l^{(z)}$ and $\pi_u^{(z)}, z \in \{0, 1, 2, 3\}$

- **SVM-RBF$_u$:** An SVM-based method with RBF kernel [45], with only user features.

Algorithm 2 QuickStop-Detection (Online)

Input:

Information trace: $y = \{y_1, y_2, \dots\}$ ▷ y_t is the tth user in the information trace y

Edge classifier $f(\cdot)$

Transition probabilities $\alpha_i(\cdot), i = 0, 1$

Thresholds $\pi_l^{(z)}$and $\pi_u^{(z)}, z \in \{0, 1, 2, 3\}$

Initialize $\Pi = \pi_1, k \leftarrow 2, Z_k^{(y)} = f(\mathbf{V}_k^{(y)}, \mathbf{U}_k^{(y)})$ ▷ π_1 is the prior of H_1 (misinformation)

while $\Pi \in \left[\pi_l^{(Z_k^{(y)})}, \pi_u^{(Z_k^{(y)})}\right]$ **do**

  $k \leftarrow k + 1$

  For each user y_k, obtain feature vector of edge: $(\mathbf{V}_k^{(y)}, \mathbf{U}_k^{(y)})$

  $Z_k^{(y)} \leftarrow f(\mathbf{V}_k^{(y)}, \mathbf{U}_k^{(y)})$

  $\Pi \leftarrow \frac{\Pi\alpha_1(Z_k^{(y)}|Z_{k-1}^{(y)})}{(1-\Pi)\alpha_0(Z_k^{(y)}|Z_{k-1}^{(y)})+\Pi\alpha_1(Z_k^{(y)}|Z_{k-1}^{(y)})}$ ▷ Compute Π

end while

$T \leftarrow k$

if $\Pi > \pi_u^{(Z_k^{(y)})}$ **then**

  $\delta_T = 1$

else if $\Pi < \pi_l^{(Z_k^{(y)})}$ **then**

  $\delta_T = 0$

end if

Output:

stopping time: T, type of information: δ_T

- **SVM-RBF$_a$:** An SVM-based method with RBF kernel [45], with both user and content features.
- **SVM-TS$_u$:** A linear SVM-based [24] method for time-series, with only user features.
- **SVM-TS$_a$:** A linear SVM-based [24] method for time-series, with both user and content features.
- **PPC_R:** A variant of RNN [22] called Gated Recurrent Unit (GRU) for time-series data. The neural network has 5,000 parameters.
- **PPC_C:** A CNN based method [22] for time-series data, which has 800 parameters.
- **PPC_R+C:** A method in [22] that combines RNN and CNN, which has 6,000 parameters.
- **CSI:** A method proposed in [32] that uses RNN for content feature extraction and FNN for user feature extraction. The three neural networks have 52,000 parameters in total.

We note that except QuickStop, all other algorithms mentioned require a pre-determined number of observations as input. QuickStop is an optimal stopping algorithm so it decides the number of observations needed in real time.

We remark that all the four neural network based methods (PPC_R, PPC_C, PPC_R+C, and CSI) require a large number of samples for training. Therefore, we used 80% of the entire 4,664 labeled traces for training the neural networks and then tested the performance on the same testing data as the other algorithms. The neural network based algorithms performed poorly when using the smaller training set as that in QuickStop.

Performance Metrics: We considered the following performance metrics.

- Accuracy: the fraction of traces that are correctly identified.
- False positive rate: the fraction of news classified as misinformation.
- False negative rate: the fraction of misinformation classified as news.
- Detection time of news: the average number of events required to declare news.
- Detection time of misinformation: the average number of events required to declare misinformation.

5.1 Numerical Results

Evolution of Π_k under QuickStop: Figure 3 illustrates the evolution of Π_k on two traces chosen from the Weibo dataset: one misinformation trace and one news trace. We can see that the upper threshold becomes smaller and the lower threshold becomes larger when we increase the propagation cost from 0.1 to 0.8, and the algorithm stops earlier when $c = 0.8$ than when $c = 0.1$. Also it takes fewer number of observations to declare misinformation than news. With $c = 0.8$, it takes 15 observations to declare the misinformation and 23 observations to declare the news. Similar trends can be observed on most of the traces.

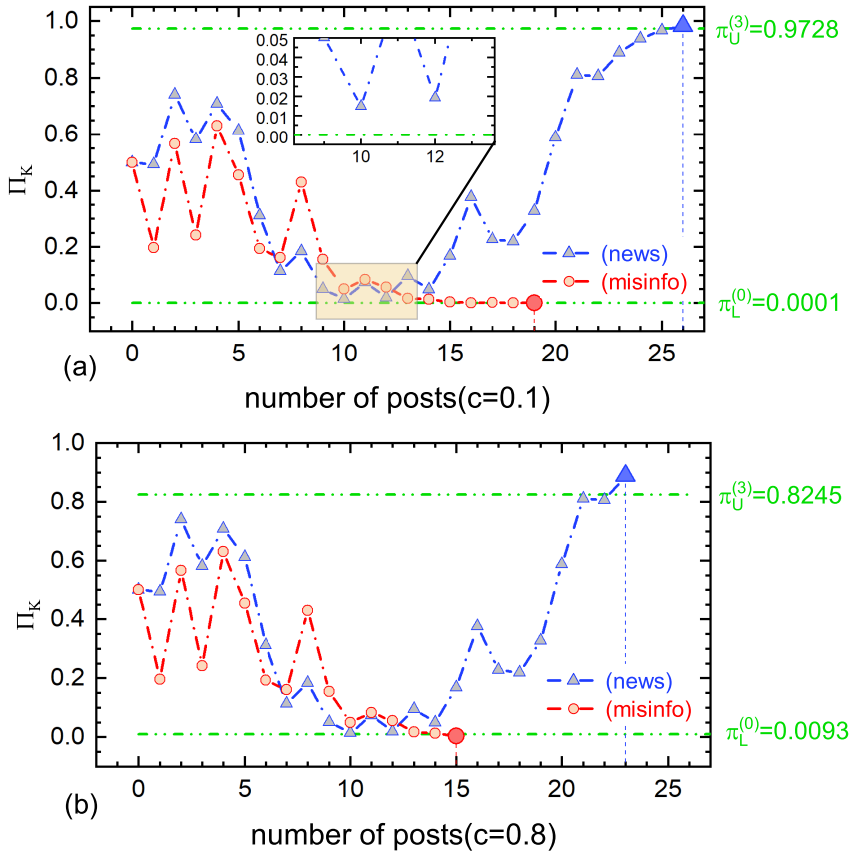

Fig. 3. Examples of Π_k and stopping time T under QuickStop

Figures 4 and 5 summarize the performances of QuickStop and the other ten algorithms. In Figure 4, QuickStop uses parameters $c_{\mathrm{I}} = c_{\mathrm{II}} = 10$ and $c = 0.05$; and the x-axis is the number of tweets used by the other ten algorithms, varying from 10 to 500. Note that when the number of observations in a trace is less than the decision deadline, then the full trace was used as the input. In Figure 5, we varied the parameter c of QuickStop from 0.05 to 1.2 with step size 0.05. In Figure 5, all ten other algorithms used full Weibo traces as input. The key observations are summarized below.

- **High Accuracy:** Figure 5 (a) shows that the accuracy of QuickStop only with user features is substantially higher than other algorithms even when other algorithms use both user features and content features. Specifically, QuickStop with $c = 0.05$ achieves higher accuracy than other algorithms with 500 observations with less than 15 observations on average. Under QuickStop, as c increases, the accuracy decreases but the number of observations used decreases as well, which is the trade-off between accuracy and speed.
- **Quick Detection:** Quickest misinformation detection is the key objective of our algorithm. Figure 4 shows that the accuracy of QuickStop in comparison with the other algorithms. QuickStop with $c = 0.05$ achieves an accuracy of 0.93 with 15 observations on average while the accuracies of all other algorithms are lower than 0.93 even with 500 observations. Note that four of the ten algorithms include content features which are not used in QuickStop.
- **Low False Negative:** In almost all cases, the false negative rate of QuickStop is lower than the false positive rate. This is because with the discriminative propagation cost, QuickStop is more aggressive on declaring misinformation than news in order to minimize the propagation cost. We also remark that CSI, which involves 52,000 parameters, has an accuracy close to QuickStop when using entire traces, but its false negative rate is much higher than QuickStop (0.097 versus 0.031).

The experimental results show that QuickStop detects misinformation faster and more accurately than other algorithms. We believe it is because QuickStop specifically models and utilizes the Markovian structure of the problem, and is based on the optimal stopping rule. The other algorithms were not optimized for the stopping time, nor do they have theoretical guarantees.

6 EVALUATION WITH SYNTHETIC DATA

We further evaluate the algorithm with synthetic network and information spreading data. We construct a network with 500 nodes using the preferential attachment model [38]. Our network includes two types of nodes: gossipers and messengers, where gossipers are more likely to spread misinformation than messengers. When a new node joins the network, it is assigned a type uniform at random, and then connects to three existing nodes in the network, i.e. forming three edges. For each edge, the new node first decides whether to connect to a node of the same type (with probability 0.7) or a node of different type (with probability 0.3). After deciding the type, say it chooses to connected to a gossiper, the new node selects a gossiper among all existing gossipers with probability proportional to their degrees. We define the edge types as follows: 0 - (messenger, messenger), 1 - (gossiper, messenger), 2 - (messenger, gossiper) and 3 - (gossiper, gossiper). We simulated the information spreading using the continuous-time SI model. For each set of parameters, we create 500 traces. Each trace was flagged as news with $\pi_0 = \pi_1 = 0.5$. The probabilities that an article is retweet over a given edge under the SI model are summarized in Table 3. From example, news spreads from a messenger to another messenger with probability 0.9, spreads from a gossiper to messenger with probability 0.7, misinformation spreads from a messenger to another messenger with probability 0.1, and from a gossiper to another gossiper with probability 0.9.

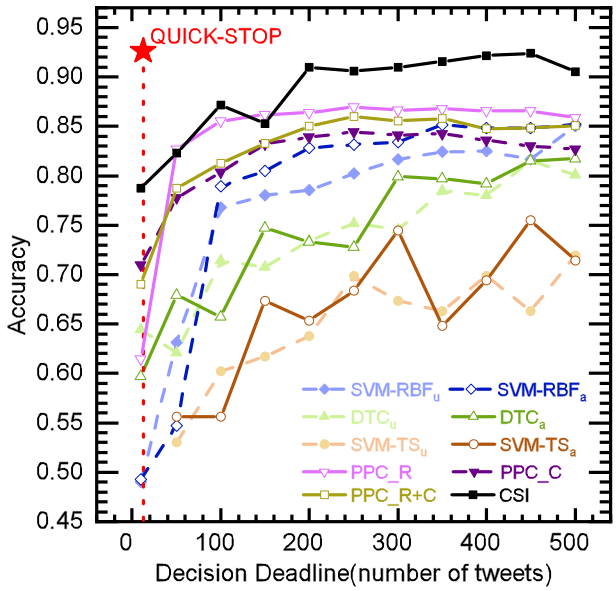

Fig. 4. Performance of Early Misinformation Detection under Different Decision Deadlines (based on the Weibo Data)

	0	1	2	3
News	0.9	0.7	0.3	0.1
Misinformation	0.1	0.2	0.7	0.9

Table 3. Probability of Information Spreading over Different Edge Types

The objective of this evaluation with the synthetic data is to evaluate the robustness of the online QuickStop-Detection with classification errors. With the synthetic data, the edge types are known so we can control the edge classification errors by random flipping the edge types and evaluate the performance of QuickStop-Detection with respect to classification errors.

Figure 6 shows the performance of QuickStop with different classifcation errors. We introduced edge classification errors such that the type of an edge is correctly classified with probability γ and misclassified with probability $1-\gamma$. We varied γ from 0.05 to 0.5. In Figure 6, we used $c_{\mathrm{I}} = c_{\mathrm{II}} = 10$ and $c = 0.3$ for QuickStop.

- **Robust to Learning Errors:** We can observe that even when 50% edges are not correctly classified, QuickStop still has an accuracy close to 91%, which demonstrates the robustness of the detection to modeling errors.

7 RELATED WORK

As we pointed out at the beginning of the introduction, government, industry and academia have made great efforts to combat misinformation. This section focuses on new developments on misinformation detection with machine-learning and data-mining methods in the research community.

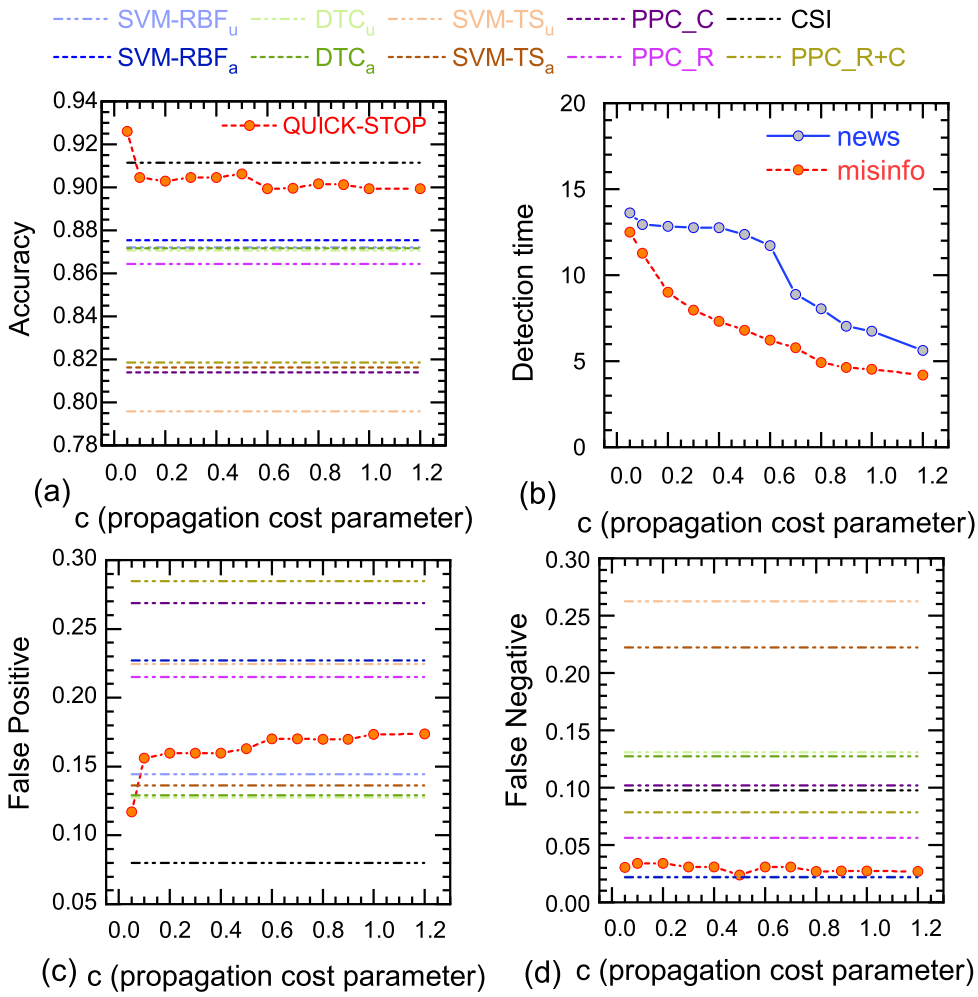

Fig. 5. Performance of QuickStop under Different Choices of Parameter c (based on the Weibo Data)

We have discussed several early detection algorithms and compared their performance with QuickStop. We now focus on other related work. The algorithm developed in [31] detects whether a post is similar to one of the posts (topics) that are known to be misinformation; and declares it as misinformation if so. A line of work [13, 14, 40] analyzes similar models and knowledge/content-based detection algorithms. These approaches are effective for detecting whether a post is associated with misinformation already identified, but not suitable for detecting new misinformation. [2, 9, 27, 35] exploit open fact-checking sources (such as DPpedia, Wikipedia, etc) to validate the truthfulness of news articles. Viewpoints of users towards news articles such as "like" and "dislike" have also been used in the literature to infer the veracity of a news article. For example, [39] classifies Facebook posts as hoax or non-hoaxes based on the set of users who "liked" them. The work [16] uses a topic

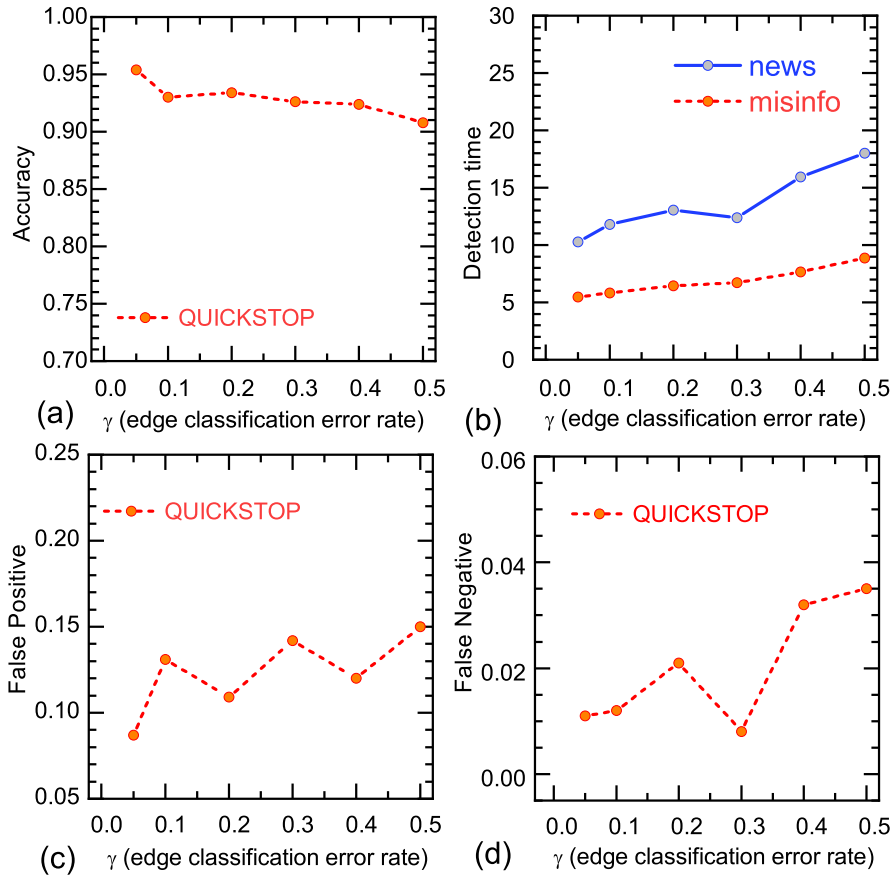

Fig. 6. Performance of QuickStop with the synthetic Data with classification errors

model to discover viewpoint values from tweets and evaluated the credibility of relevant posts based on these viewpoints.

In [4], a comprehensive data-mining approach has been proposed for determining the veracity of social media contents. They considered four categories of features: message-based, user-based, topic-based, and propagation-based features to study information credibility, and proposed a PageRank-like credibility analysis method to verify the credibility of twitter events. The features used in [4] have later been used in other papers [12, 21, 45]. In [17], the authors argued that features vary over time. They reported that linguistic features are effective for detecting rumor even at the early stage of information spreading. A model for time-varying features has been proposed in [24]. [44] explores the use of the features of the message propagation trees for detecting misinformation. [8] analyzes six categories of features: comprehensibility, sentiment, time-orientation, quantitative details, writing style, and topic. [10] analyzes users' stance in their tweets to evaluate the credibility of information. [5] studies the characteristics of users who often post misinformation, and proposes that after identifying these users, a news article is likely to be misinformation if it spreads among these users. [42] proposes a misinformation detection algorithm with dynamic time wrapping and hidden Markov models based on three categories of features (linguistic, user identities and temporal

propagation related features). Recently, deep neural network (CNN, RNN, and FNN) based methods have also been used for misinformation detection [22, 32].

Users play the central role in information diffusion in social networks. Their social engagements such as sharing, forwarding, commenting are considered to be auxiliary information for improving fake news detection. [41] uses users' flags of fake news as signals and leverages community for misinformation detection by learning the users' flagging accuracy. Online social network users who intentionally spread misinformation can be divided into three categories: (1) bots, software apps that run automated scripts[5] (2) trolls, persons who like to provoke others, and (3) cyborgs[6], accounts registered to run automated programs that mimic human behaviors [36]. [6, 34] analyze the behavior patterns of bots and trolls in misinformation propagation. In [7], an automated method is proposed for classifying the users into the three categories mentioned above. In [29], bot detection is studied. [37] analyzes the users' role in spreading information and concludes that (1) some specific users are more likely to believe in misinformation than real news; (2) these users have different features form other users. These two key observations motivated the edge-based model considered in this paper. [1] proposes a method for measuring user credibility in information spreading for misinformation detection. The spread of rumors and misinformation has also been studied in [11, 15, 43], where it has been shown that misinformation and news have different spreading patterns and structures. In this paper, we consider both edge profiles (the edge classification) and spreading patterns (the Markovian spreading model) in QuickStop to design a highly efficient misinformation detection algorithm. Different from existing work, QuickStop is an optimal stopping algorithm that optimizes the number of observations in realtime and makes the quickest decision on misinformation detection. Finally, recent algorithms for distinguishing epidemics from random infection (e.g., [28]) and for locating information sources (e.g., [33]) can also help detect misinformation. A comprehensive review of diffusion source localization can be found in [47].

8 CONCLUSIONS

In this paper, we proposed a quickest misinformation detection algorithm, named QuickStop. We formulated the problem as an optimal stopping problem with a asymmetric cost function towards misinformation. We proved that the problem is a Markov optimal stopping problem and showed that the solution is a threshold-based stopping rule based on the martingale theory. Our numerical results with a real-world data demonstrated that QuickStop outperforms existing algorithms even though the latter use 10 times (sometimes 50 times) more observations and use more features. Our numerical evaluation with the synthetic data showed that the algorithm is robust to edge classification errors.

ACKNOWLEDGEMENT

The authors thank R. Srikant for his invaluable comments and feedback. This work was supported in part by NSF Grant IIS-1715385.

REFERENCES

[1] Mohammad-Ali Abbasi and Huan Liu. 2013. Measuring user credibility in social media. In *Proc. Int. Conf. Social Comput. Behavioral-Cultural Model. and Prediction (SBP)*. Springer-Verlag, Washington, DC, 441–448.

[2] Michele Banko, Michael J Caella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web.. In *Proc. Int. Joint Conf. Artif. Intell. Org (IJCAI)*, Vol. 7. Morgan Kaufmann Publishers Inc., Hyderabad, India, 2670–2676.

[5] https://en.wikipedia.org/wiki/Internet_bot

[6] https://en.wikipedia.org/wiki/Internet_troll

[3] Dimitri P. Bertsekas. 2017. *Dynamic Programming and Optimal Control* (4th ed.). Athena Scientific, Belmont, MA.

[4] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information Credibility on Twitter. In *Proc. Int. Conf. World Wide Web (WWW)*. ACM, New York, NY, 675–684. https://doi.org/10.1145/1963405.1963500

[5] Cheng Chang, Yihong Zhang, Claudia Szabo, and Quan Z Sheng. 2016. Extreme user and political rumor detection on Twitter. In *Int. Conf. on Adv. Data Mining and Appl(ADMA)*. Springer, Gold Coast, Australia, 751–763. https://doi.org/10.1007/978-3-319-49586-6_54

[6] Tong Chen, Xue Li, Hongzhi Yin, and Jun Zhang. 2018. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. In *Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD)*. Springer, Melbourne, Australia, 40–52.

[7] Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. 2012. Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *IEEE Trans. Dependable Secur. Comput.* 9, 6 (Nov. 2012), 811–824. https://doi.org/10.1109/TDSC.2012.75

[8] Alton Yeow Kuan Chua and Snehasish Banerjee. 2016. Linguistic predictors of rumor veracity on the Internet. In *Proc. Int. MultiConf. Emerg. and Comput. Scientists (IMECS)*. Newswood Limited, Hongkong, China, 387–391.

[9] 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.

[10] Bontcheva Kalina Liakata Maria Procter Rob Wong Sak Hoi Geraldine Derczynski, Leon and Arkaitz Zubiaga. 2017. SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proc. Int. Workshop Semantic Eval.(SemEval)*. Association for Computational Linguistics, Vancouver, Canada, 69–76. https://doi.org/10.18653/v1/S17-2006

[11] Adrien Friggeri, Lada A Adamic, Dean Eckles, and Justin Cheng. 2014. Rumor Cascades.. In *Proc. Int. Conf. Weblogs and Social Media (ICWSM)*. AAAI Press, Ann Arbor, MI, 101–110.

[12] Manish Gupta, Peixiang Zhao, and Jiawei Han. 2012. Evaluating event credibility on Twitter. In *IEEE Int. Conf. Data Mining (ICDM)*. SIAM, Brussels, Belgium, 153–164. https://doi.org/10.1137/1.9781611972825.14

[13] Sardar Hamidian and Mona Diab. 2016. Rumor Identification and Belief Investigation on Twitter. In *Proc. Workshop Comput. Approaches to Subjectivity, Sentiment and Social Media Anal (WASSA)*. Association for Computational Linguistics, San Diego, CF, 3–8. https://doi.org/10.18653/v1/W16-0403"

[14] Sardar Hamidian and Mona T Diab. 2015. Rumor detection and classification for Twitter data. In *Proc. Int. Conf. Social Media Technol (SOTICS)*. IARIA XPS Press, Barcelona, Spain, 71–77. https://doi.org/10.1109/SCIS-ISIS.2012.6505254

[15] Fang Jin, Edward Dougherty, Parang Saraf, Yang Cao, and Naren Ramakrishnan. 2013. Epidemiological modeling of news and rumors on twitter. In *Proc. Workshop Social Netw Mining and Anal (SNAKDD)*. ACM, Chicago, IL, 8:1–8:9. https://doi.org/10.1145/2501025.2501027

[16] Zhiwei Jin, Juan Cao, Yongdong Zhang, and Jiebo Luo. 2016. News Verification by Exploiting Conflicting Social Viewpoints in Microblogs.. In *AAAI Conf. Artificial Intelligence*. AAAI Press, Phoenix, AZ, 2972–2978.

[17] Sejeong Kwon, Meeyoung Cha, and Kyomin Jung. 2017. Rumor detection over varying time windows. *PloS ONE* 12, 1 (jan 2017), e0168344. https://doi.org/10.1371/journal.pone.0168344

[18] Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. 2013. Prominent features of rumor propagation in online social media. In *IEEE Int. Conf. Data Mining (ICDM)*. IEEE, Dallas, TX, 1103–1108. https://doi.org/10.1109/ICDM.2013.61

[19] Damien Lamberton and Bernard Lapeyre. 2008. *Introduction to Stochastic Calculus Applied to Finance* (2nd ed.). Chapman & Hall/CRC, Boca Raton, FL, 38.

[20] David MJ Lazer, Matthew A Baum, Yochai Benkler, Adam J Berinsky, Kelly M Greenhill, Filippo Menczer, Miriam J Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, et al. 2018. The science of fake news. *Science* 359, 6380 (2018), 1094–1096.

[21] Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. 2015. Real-time Rumor Debunking on Twitter. In *Proc. ACM Int. Conf. Information and Knowledge Management (CIKM)*. ACM, Melbourne, Australia, 1867–1870. https://doi.org/10.1145/2806416.2806651

[22] Yang Liu and Yi-Fang Brook Wu. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *AAAI Conf. Artificial Intelligence*. AAAI Press, New Orleans, Louisiana, 354–361.

[23] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting Rumors from Microblogs with Recurrent Neural Networks. In *Proc. Int. Joint Conf. Artif. Intell. Org (IJCAI)*. AAAI Press, New York, NY, 3818–3824.

[24] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. 2015. Detect Rumors Using Time Series of Social Context Information on Microblogging Websites. In *Proc. ACM Int. Conf. Information and Knowledge Management (CIKM)*. ACM, Melbourne, Australia, 1751–1754. https://doi.org/10.1145/2806416.2806607

[25] Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. In *Proc. Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Vol. 1. Association for Computational Linguistics, Vancouver, Canada, 708–717.

[26] Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor detection on twitter with tree-structured recursive neural networks. In *Proc. Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Vol. 1. Association for Computational Linguistics, Melbourne, Australia, 1980–1989.

[27] Amr Magdy and Nayer Wanas. 2010. Web-based statistical fact checking of textual documents. In *Proc. Int. Workshop Search and Mining User-generated Contents (SMUC)*. ACM, ACM, Toronto, ON, Canada, 103–110. https://doi.org/10.1145/1871985.1872002

[28] Chris Milling, Constantine Caramanis, Shie Mannor, and Sanjay Shakkottai. 2013. Detecting epidemics using highly noisy data. In *Proc. ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*. ACM, Bangalore, India, 177–186. https://doi.org/10.1145/2491288.2491294

[29] Fred Morstatter, Liang Wu, Tahora H Nazer, Kathleen M Carley, and Huan Liu. 2016. A new approach to bot detection: striking the balance between precision and recall. In *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. and Mining (ASONAM)*. IEEE Press, Davis, CF, 533–540.

[30] H. Vincent Poor and Olympia Hadjiliadis. 2008. *Quickest Detection.* Cambridge University Press, Cambridge, United Kingdom, Chapter Markov optimal stopping theory, 40–64. https://doi.org/10.1017/CBO9780511754678.005

[31] Vahed Qazvinian, Emily Rosengren, Dragomir R Radev, and Qiaozhu Mei. 2011. Rumor Has It: Identifying Misinformation in Microblogs. In *Proc. the Conf. on Empirical Methods in Natural Language Process.* Association for Computational Linguistics, Edinburgh, United Kingdom, 1589–1599.

[32] Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. Csi: A hybrid deep model for fake news detection. In *Proc. ACM Int. Conf. Information and Knowledge Management (CIKM)*. ACM, Singapore, Singapore, 797–806. https://doi.org/10.1145/3132847.3132877

[33] Devavrat Shah and Tauhid Zaman. 2011. Rumors in a Network: Who's the Culprit? *IEEE Trans. Inf. Theory* 57, 8 (Aug. 2011), 5163–5181. https://doi.org/10.1109/TIT.2011.2158885

[34] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Alessandro Flammini, and Filippo Menczer. 2018. The spread of fake news by social bots. *Nature Communications* 9 (2018), 96–104. https://doi.org/10.1038/s41467-018-06930-7

[35] Baoxu Shi and Tim Weninger. 2016. Fact checking in heterogeneous information networks. In *Proc. Int. Conf. World Wide Web (WWW)*. International World Wide Web Conferences Steering Committee, Montréal, Québec, Canada, 101–102. https://doi.org/10.1145/2872518.2889354

[36] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter* 19, 1 (Sept. 2017), 22–36. https://doi.org/10.1145/3137597.3137600

[37] Kai Shu, Suhang Wang, and Huan Liu. 2018. Understanding user profiles on social media for fake news detection. In *Conf. Multimedia Inf. Process. and Retrieval (MIPR)*. IEEE, Miami, FL, 430–435. https://doi.org/10.1109/MIPR.2018.00092

[38] Herbert A Simon. 1955. On a class of skew distribution functions. *Biometrika* 42, 3-4 (Dec. 1955), 425–440. https://doi.org/10.1093/biomet/42.3-4.425

[39] Eugenio Tacchini, Gabriele Ballarin, Marco L Della Vedova, Stefano Moret, and Luca de Alfaro. 2017. Some like it hoax: Automated fake news detection in social networks. In *Workshop on Data Science for Social Good*, Vol. 1960. CEUR-WS, Skopje, Macedonia, 1–15.

[40] Tetsuro Takahashi and Nobuyuki Igata. 2012. Rumor detection on twitter. In *Proc. Int. Conf. Soft Comput. and Intell. Syst. and Proc. Int. Symp. Adv. Intell. Syst (SCIS & ISIS)*. IEEE, Kobe, Japan, 452–457. https://doi.org/10.1109/SCIS-ISIS.2012.6505254

[41] Sebastian Tschiatschek, Adish Singla, Manuel Gomez Rodriguez, Arpit Merchant, and Andreas Krause. 2018. Fake News Detection in Social Networks via Crowd Signals. In *Proc. Int. Conf. World Wide Web (WWW)*. International World Wide Web Conferences Steering Committee, Lyon, France, 517–524. https://doi.org/10.1145/3184558.3188722

[42] Soroush Vosoughi. 2015. *Automatic detection and verification of rumors on Twitter.* Ph.D. Dissertation. Massachusetts Institute of Technology.

[43] Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science* 359, 6380 (2018), 1146–1151. https://doi.org/10.1126/science.aap9559

[44] Ke Wu, Song Yang, and Kenny Q Zhu. 2015. False rumors detection on Sina Weibo by propagation structures. In *Proc. Int. Conf. Data Engineering (ICDE)*. IEEE, Seoul, South Korea, 651–662. https://doi.org/10.1109/ICDE.2015.7113322

[45] Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. 2012. Automatic Detection of Rumor on Sina Weibo. In *Proc. Ann. ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD)*. ACM, Beijing, China, Article 13, 7 pages. https://doi.org/10.1145/2350190.2350203

[46] Fan Ye and Enlu Zhou. 2013. Optimal Stopping of Partially Observable Markov Processes: A Filtering-Based Duality Approach. *IEEE Trans. Autom. Control* 58, 10 (Oct 2013), 2698–2704.

[47] Lei Ying and Kai Zhu. 2018. Diffusion source localization in large networks. *Synthesis Lectures on Communication Networks* 11, 1 (2018), 1–95.

[48] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. 2015. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proc. Int. Conf. World Wide Web (WWW)*. (IW3C2, Florence, Italy, 1395–1405. https://doi.org/10.1145/2736277.2741637

A APPENDICES

A.1 Proof of Theorem 3.1

We first show that $E[cT\mathbb{I}_{H_1}] = E[cT\Pi_T]$ when T is a stopping time.

$$\begin{aligned} E[cT\mathbb{I}_{H_1}] &= E[E[cT\mathbb{I}_{H_1}|T]] \\ &= \sum_{k=1}^{\infty} ckE\left[\mathbb{I}_{H_1}|T=k\right]\Pr(T=k). \end{aligned}$$

Since T is a stopping time based on $Z_1, \cdots, Z_T$, we further have

$$\begin{aligned} E\left[\mathbb{I}_{H_1}|T=k\right] &= E\left[E\left[\mathbb{I}_{H_1}|Z_1,\ldots,Z_k\right]|T=k\right] \\ &= E\left[\Pi_k|T=k\right]. \end{aligned}$$

Therefore, we have

$$E[cT\mathbb{I}_{H_1}] = \sum_{k=1}^{\infty} ckE\left[\Pi_k|T=k\right]\Pr(T=k) = E\left[cT\Pi_T\right].$$

For any $T \in \mathcal{T}$, it is well known (see for example [30]) that

$$\inf_{\delta_T} c_e(\delta_T) = E\left[\min\{c_{\mathrm{II}}\Pi_T, c_{\mathrm{I}}(1-\Pi_T)\}\right].$$

We next present the proof tailored for our problem for the completeness of the paper.

Note that the equation is obvious when $\pi_1 = 0$ or $\pi_1 = 1$, so we only consider the case $\pi_1 \in (0,1)$. Recall that

$$
\begin{aligned}
& c_e(\delta_T) \\
=&(1-\pi_1)c_{\mathrm{I}} \Pr(\delta_T = 1|H_0) + \pi_1 c_{\mathrm{II}} \Pr(\delta_T = 0|H_1) \\
=&c_{\mathrm{I}} \Pr(\delta_T = 1, H_0) + c_{\mathrm{II}} \Pr(\delta_T = 0, H_1) \\
=&\sum_{k=1}^{\infty} (c_{\mathrm{I}} \Pr(\delta_T = 1, H_0|T=k) + c_{\mathrm{II}} \Pr(\delta_T = 0, H_1|T=k)) \Pr(T=k) \\
=&\sum_{k=1}^{\infty} (c_{\mathrm{I}} \Pr(\delta_k = 1, H_0|T=k) + c_{\mathrm{II}} \Pr(\delta_k = 0, H_1|T=k)) \Pr(T=k) \\
=&\sum_{k=1}^{\infty} \big(c_{\mathrm{I}} E\left[E\left[\mathbb{I}_{\delta_k=1}\mathbb{I}_{H_0}|Z_1,\cdots,Z_k\right]|T=k\right] + \\
& \quad c_{\mathrm{II}} E\left[E\left[\mathbb{I}_{\delta_k=0}\mathbb{I}_{H_1}|Z_1,\cdots,Z_k\right]|T=k\right]\big) \Pr(T=k) \\
=&\sum_{k=1}^{\infty} E\left[\left(c_{\mathrm{I}}\mathbb{I}_{\delta_k(Z_1,\cdots,Z_k)=1}(1-\Pi_k) + c_{\mathrm{II}}\mathbb{I}_{\delta_k(Z_1,\cdots,Z_k)=0}\Pi_k\right)\Big| T=k\right] \\
& \times \Pr(T=k) \\
& \geq_{(a)} \sum_{k=1}^{\infty} E\left[\min\{c_{\mathrm{I}}(1-\Pi_k), c_{\mathrm{II}}\Pi_k\}|\, T=k\right] \Pr(T=k) \\
& = E\left[\min\{c_{\mathrm{I}}(1-\Pi_T), c_{\mathrm{II}}\Pi_T\}\right],
\end{aligned}
$$

where the inequality (a) becomes equality when the algorithm declares H_1 when $c_{\mathrm{I}}(1-\Pi_T) \leq c_{\mathrm{II}}\Pi_T$ and declares H_0 otherwise.

A.2 Proof of Theorem 3.2

We define the following value function for $n \geq 1$

$$
s_n(\pi, z) = \inf_{T\in\mathcal{T}, T\geq n} E\left[g(\Pi_T) + cT\Pi_T|\Pi_n = \pi, Z_n = z\right].
$$

Then $s_n(\pi, z)$ is the minimum expected total cost if one is only allowed to stop at or after time step n given the state at n. Note $\mathcal{T} = \{T \in \mathcal{T} : T \geq 1\}$. Then the minimum expected total cost over the prior π_0 is

$$
s_1^* \triangleq \inf_{T\in\mathcal{T}} E\left[g(\Pi_T) + cT\Pi_T\right] = \frac{1}{4}\sum_{z=0}^{3} s_1(\pi_0, z),
$$

where we use the fact that $\Pi_1 = \Pi_0 = \pi_0$ as the first observation Z_1 does not provide any information about the type of the information.

Now according to the optimality principle of dynamic programming,

$$
s_k(\pi, z) = \min\left\{g(\pi) + ck\pi, E\left[s_{k+1}(\Pi_{k+1}, Z_{k+1})|\,\Pi_k = \pi, Z_k = z\right]\right\},
$$

where $\{\Pi_k\}$ is a random process defined by $\{Z_k\}$ as in equation (2).

We next show that $\{\Pi_k\}$ is a martingale with respect to $\{Z_k\}$. Define $\mathcal{F}_k = \sigma(Z_1, \cdots, Z_k)$, which is the σ-algebra generated by $Z_1, \ldots, Z_k$. We have

$$
E\left[\Pi_{k+1}|\mathcal{F}_k\right] = \sum_{z=0}^{3} E\left[\Pi_{k+1}|\mathcal{F}_k, Z_{k+1} = z\right] \Pr(Z_{k+1} = z|\mathcal{F}_k).
$$

Since

$$\begin{aligned}&\Pr(Z_{k+1}=z|\mathcal{F}_k)\\=&\Pr(Z_{k+1}=z|\mathcal{F}_k,H_1)\Pr(H_1|\mathcal{F}_k)+\Pr(Z_{k+1}=z|\mathcal{F}_k,H_0)\Pr(H_0|\mathcal{F}_k)\\=&\alpha_1(Z_{k+1}=z|Z_k)\Pi_k+\alpha_0(Z_{k+1}=z|Z_k)(1-\Pi_k),\end{aligned}$$

we have

$$\begin{aligned}&E\left[\Pi_{k+1}|\mathcal{F}_k\right]\\=&\sum_z\frac{\Pi_k\alpha_1(Z_{k+1}=z|Z_k)}{\Pi_k\alpha_1(Z_{k+1}=z|Z_k)+(1-\Pi_k)\alpha_0(Z_{k+1}=z|Z_k)}\\&\times(\alpha_1(Z_{k+1}=z|Z_k)\Pi_k+\alpha_0(Z_{k+1}=z|Z_k)(1-\Pi_k))\\=&\sum_z\Pi_k\alpha_1(Z_{k+1}=z|Z_k)\\=&\Pi_k.\end{aligned}$$

For $n\geq 1$ let $\Pi'_k=\Pi_{k+n-1}$ and $Z'_k=Z_{k+n-1}$ for all $k\geq 1$. Then

$$\begin{aligned}&s_n(\pi,z)\\=&\inf_{\substack{T\in\mathcal{T}\\T-n+1\geq 1}}E\left[g(\Pi_T)+c(T-n+1)\Pi_T+c(n-1)\Pi_T|\Pi_n=\pi,Z_n=z\right]\\=&\inf_{\substack{T'\in\mathcal{T}\\T'\geq 1}}E[g(\Pi'_{T'})+cT'\Pi'_{T'}|\Pi'_1=\pi,Z'_1=z]+c(n-1)\pi\\=&s_1(\pi,z)+c(n-1)\pi.\end{aligned}$$

In other words, because the posterior probability $\{\Pi_k\}$ is a martingale with respect to the observations $\{Z_k\}$, every time step passed before time n (when one is allowed to stop and make a decision) incurs a constant additive cost of $c\pi$ to the minimum expected total cost.

Now define

$$s^{(z)}(\pi)=s_1(\pi,z)-c\pi. \tag{10}$$

Then for any $k\geq 1$,

$$\begin{aligned}&s^{(z)}(\pi)\\=&s_k(\pi,z)-ck\pi\\=&\min\left\{g(\pi),E\left[s^{(Z_{k+1})}(\Pi_{k+1})+c(k+1)\Pi_{k+1}\middle|\Pi_k=\pi,Z_k=z\right]-ck\pi\right\}\\=&\min\left\{g(\pi),E\left[s^{(Z_{k+1})}(\Pi_{k+1})\middle|\Pi_k=\pi,Z_k=z\right]+c\pi\right\}.\end{aligned}$$

Hence s as defined in (10) satisfies the Bellman equation (9).

Note that $g(\pi)+ck\pi$ is the cost when the information type is declared at iteration k given $\Pi_k=\pi$, and $E\left[s^{(z)}(\Pi_{k+1})\middle|\Pi_k=\pi,Z_k=z\right]+c(k+1)\pi$ is the minimum cost the information type is declared after iteration k given $\Pi_k=\pi$. Therefore, at optimal stopping time T, we have

$$s^{(z)}(\pi)+ck\pi=g(\pi)+ck\pi$$

i.e.,

$$s^{(z)}(\pi)=g(\pi).$$

Furthermore, if $s^{(z)}(\pi) = g(\pi)$ and $c_{\text{II}}\pi < c_{\text{I}}(1-\pi)$, then $s^{(z)}(\pi) = c_{\text{II}}\pi$, so the information is declared to be news; otherwise, it is declared to be misinformation. Therefore, after solving $s^{(z)})(\pi)$, we have

$$\pi_l^{(z)} = \sup_{\pi} \left\{ \pi : s^{(z)}(\pi) = c_{\text{II}}\pi \right\},$$

and

$$\pi_u^{(z)} = \inf_{\pi} \left\{ \pi : s^{(z)}(\pi) = c_{\text{I}}(1-\pi) \right\}.$$

Received February 2019; revised April 2019; accepted May 2019