Detection of Content-level Collusive Activities in Online Social Networks

Aditya Chetan
2016217
aditya16217@iiitd.ac.in

Brihi Joshi
2016142
brihi16142@iiitd.ac.in

BTP report submitted in partial fulfillment of the requirements
for the Degree of B.Tech. in Computer Science and Applied Math and B.Tech. in Computer Science & Engineering respectively

on February 11, 2020

BTP Track: Research

BTP Advisor
Dr. Tanmoy Chakraborty

Indraprastha Institute of Information Technology
New Delhi

Students' Declaration

We hereby declare that the work presented in the report entitled **"Detection of Content-level Collusive Activities in Online Social Networks"** submitted by us, Aditya Chetan and Brihi Joshi, for the partial fulfillment of the requirements for the degree of *Bachelor of Technology* in *Computer Science & Applied Mathematics* and *Bachelor of Technology* in *Computer Science & Engineering* respectively, at Indraprastha Institute of Information Technology, Delhi, is an authentic record of our work carried out under the guidance of **Dr. Tanmoy Chakraborty**. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the reward of any other degree.

.... **Place & Date: New Delhi, India February 11, 2020**
Aditya Chetan

....
Brihi Joshi

Certificate

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

.... **Place & Date: New Delhi, India February 11, 2020**
Dr. Tanmoy Chakraborty

Abstract

Twitter, as a micro-blogging service has been increasingly used to express opinions, promote brands and share news. Often, the popularity of a tweet is denoted by how other users of the platform are reacting to it. Retweets are a very important aspect of the endorsement of tweets - a high number of retweets gain a lot of attention. This has led to the creation of unfair methods for gaining a high number of retweets as a natural way of gaining retweets is very time-consuming – one such shortcut is to approach the blackmarket services, and gain retweets for their own tweets by retweeting other customers' tweets. Thus, the users intrinsically become a part of a collusive ecosystem controlled by these services.

In this project, we expand on the idea of collusion by proposing `CoReRank`, an unsupervised algorithm to rank users and tweets based on their participation in the collusive ecosystem. Later, we also leverage the presence of sparse labelled data to formulate its semi-supervised version called `CoReRank+`.

Keywords: Retweets, Collusion, Twitter, Blackmarket, Online Social Networks.

Acknowledgments

We would like to thank our advisor, Dr. Tanmoy Chakraborty for his guidance and encouragement in this project. We would also like to thank Mr. Hridoy Sankar Dutta for his valuable inputs and contributions in this project.

Work Distribution

- Aditya was involved in Data Collection, implementing the baselines, formulating the algorithm and calculating the importance of different components in the experimental results.
- Brihi was involved in Data Collection, formulating the Graph construction method, and formalizing the properties of `CoReRank`. She also was responsible for Comparative evaluation and Scalability analysis of the method.
- The remaining work (for e.g - the ideation and implementation of `CoReRank` and `CoReRank+` providing theoretical guarantees and the overall structure of the project) was done jointly with equal contribution of both.

Contents

Chapter 1

Introduction

1.1 Motivation

Current state-of-the-art algorithms either focus on *Bot Detection* or *Spam Detection* in micro-blogging sites like Twitter. Detection of collusive users is challenging for two reasons -

- Unlike bots, they do not have a fixed activity and purpose. Unlike fake users, their profiles are not fake and thus, not flagged by in-house algorithms deployed by Twitter. What makes them interesting to study is that they demonstrate an amalgamation of *inorganic and organic behavior* - they retweet content associated with the blackmarket services and they also promote content which appeals to their interest.
- Collecting large scale labeled data of collusion users is extremely challenging. This necessitates the design of an unsupervised approach to detect collusive users.

1.2 Research Problem

There are two types of blackmarket services - Premium and Freemium. Premium services only provide services to users after a minimal payment. We primarily focus on the Freemium service model, to be specific, Credit-based services where users retweet others to gain retweets on their own content, a type of a ‘give and take’ relationship. Such users are termed as **Collusive Users** and the activity is called **Collusion**.

The problems that we address are as follows -

- How can we design an efficient system to **simultaneously** detect users (based on their unusual retweeting pattern) and tweets (based on the credibility of the users who retweet them) involved in collusive blackmarket services?
- How can we develop an algorithm that detects collusive users, addressing the fact that there is a scarcity of labelled data? Can some labelled data be leveraged to enhance the algorithms?
- Is collusion detection really different from other fraudulent detection algorithms? How do other state-of-the-art algorithms perform in detecting collusion?

This work has been accepted at The Twelfth International Conference on Web Search and Data Mining 2019 - WSDM 2019 (Acceptance Rate - 16%) for a full paper presentation.

Chapter 2

Literature Survey

2.1 Fraud Detection in OSNs

A series of works investigated fraud detection from various aspects. [19] proposed a honeypot-based approach to detect spammers on Twitter and MySpace. They used profile-based features of the spammers to design supervised classifiers. [2] also used supervised classifiers with features based on tweet content and user social behavior. [35] detected criminal profiles on Twitter. A large number of papers designed methods to detect bots on Twitter. [3] analyzed the behavior of humans, bots, and cyborgs. They reported that humans have complex timing behavior while bots and cyborgs have a periodic behavior. [5] and [32] designed unsupervised and supervised approaches respectively to detect bots. [18, 33] focused on the detection of spam tweets based on tweet-related features. [17] studied fraud favoritism on retweets. Some other studies of fraud detection based on tweet content are [1, 9, 10, 23]. Multiple papers focused on detecting fraud using URLs embedded in tweets [19, 34] and blacklisted URLs [10, 13, 14]. [11] proposed `NDSync` to detect synchronized fraud activities (fake retweets) on Twitter. [12] studied the influence of fraudulent and genuine retweet threads and discovered patterns ('Triangles' and 'Homogeneity') followed by fraudulent users.

2.2 Study of Blackmarket Services

Compared to the study on general fraud detection, the effort so far has been limited to investigate fraudulent activities by blackmarket services. A detailed analysis of blackmarkets with the impact on multiple social networks was studied by [6]. [27] studied the multifaceted behavior of the blackmarket agencies. [29] studied the market size and market price of multiple Twitter follower markets. [30] used supervised approach to detect accounts of blackmarket services. [28] studied the characteristics of Twitter follower merchant markets. [20] proposed a method to detect followers who provide voluntary following services to make the profit. [4] used supervised approach for fake blackmarket follower detection.

Table 2.1: Comparison of `CoReRank` and other baseline methods w.r.t different dimensions of an algorithm.

	[5]	[32]	[9]	[11]	[7]	[15]	Our
Address collusion phenomenon					✓		✓
Consider graph information		✓				✓	✓
Consider topic information							✓
Unsupervised approach	✓			✓		✓	✓
Return ranked list of users				✓			✓
Detect both collusive users and tweets							✓
Theoretical guarantees						✓	✓

Few studies attempted to analyze blackmarkets in other platforms such as live video-streaming, online recruitment, etc. [24] analyzed six different underground forums where users are involved in selling goods and services. [26] proposed an unsupervised approach to detect bot-generated broadcasts and views for famous broadcasting platforms (YouTube and Twitch). [31] used machine learning approach and textual analysis on a dataset containing real-life job ads to detect whether given employment or job advertisement is legitimate or fraudulent. [7] studied blackmarket services which provide fake retweets and designed supervised approach to detect collusive retweeters.

2.3 Major Differences with Existing Approaches

`CoReRank` combines both *graph structure* and *behavioral properties* in an *unsupervised manner* and returns *ranked lists of users and tweets simultaneously* based on *collusive retweeting activities.* `CoReRank` is also *guaranteed to converge* in a finite number of iterations with linear time complexity. Table 2.1 summarizes a comparison of `CoReRank` with other existing approaches. `CoReRank` is the only one which matches all specifications.

Chapter 3

Data Description

We solicited tweets from various freemium blackmarket services. We searched for these services by querying search engines with keywords such as ‘Free Retweets’, ‘Retweet my Tweet’, etc. We collected data from the following services (after taking proper IRB approval) - YouLikeHits[1] and Like4Like[2]. These services provide an ‘earning area’ where tweets submitted by the customers of the service are displayed so that other customers can gain credits by retweeting the tweets. We designed a web-scraper to extract tweets (T_b) as well as the customers (U_c) who submitted these tweets. We created three user sets: ground-truth genuine user set $\mathcal{S}_g$, ground-truth collusive user set $\mathcal{S}_c$, and unknown user set $\mathcal{S}_u$. We collected users (U_b) who retweeted/quoted the extracted tweets. Out of these users, we found (i) 4 verified Twitter users (U_b^v), and (ii) 329 users who were also a part of U_c. We added the former to $\mathcal{S}_g$ and latter to $\mathcal{S}_c$. The remaining 7451 users were added to $\mathcal{S}_u$. We further increased $\mathcal{S}_g$ by adding the followees of the verified users (U_f^v) with the assumption that verified users are more likely to follow genuine users. We also collected the tweets (T_f^v) that were retweeted/quoted by U_f^v. Finally, we collected all retweets and quotes (max. 3200) of users from their timeline. At this stage, the size of the ground-truth sets is as follows: $|\mathcal{S}_c| = 329$, $|\mathcal{S}_g| = 2667$ and $|\mathcal{S}_u| = 7451$. Users present in $\mathcal{S}_u$ were further annotated by human experts as collusive/genuine (see Section 5.2).

[1] https://www.youlikehits.com/

[2] https://like4like.com/

Chapter 4

Proposed Methodology

In this section, we explain our efforts in formulating `CoReRank`. (which is motivated by [7, 16]).

4.1 `CoReRank` Preliminaries

We start by constructing a graph comprising users and tweets as nodes. We hypothesize that users and tweets have intrinsic traits that often demonstrate their collusive nature and their credibility. Thus, users and tweets can be allotted scores that define these traits. The reason to operate on the graph and not individual users/tweets (as done in [7]) is that these scores are interdependent on each other and the graph as a whole.

Definition 1 (**User Support).** *A tweet t is considered to be* supported *(by retweeting or quoting) by a user u if u either retweeted or quoted t. This is given by $S(u,t)$ as follows -*

$$S(u,t) = \begin{cases} w_q & \textit{if } u \textit{ quoted } t \\ w_r & \textit{if } u \textit{ retweeted } t \\ 0 & \textit{otherwise} \end{cases}$$

Here, w_r (w_q) denote the weight of the edge when u retweeted (quoted) t. The relation between w_r and w_q can be defined by: $0 < w_r \leq w_q < 1$ as a quote is essentially a retweet augmented with further text or media, allowing the tweet to have more importance or weight than a simple retweet. In our model, we set w_r and w_q to 0.5 and 0.75 respectively.

To select the optimum weights for the edges of the graph, we proceed as follows – the weights of the edges of the graph, w_r and w_q are taken in the range $(0, 1)$ at a step of 0.25. Also, the condition $0 < w_r \leq w_q < 1$ needs to be satisfied. Thus, all the possible combinations of (w_r, w_q) are ordered in the following fashion- $(0.25, 0.25)$, $(0.25, 0.5)$, $(0.25, 0.75)$, $(0.5, 0.5)$, $(0.5, 0.75)$ and $(0.75, 0.75)$.

Figure 4.1 shows how the precision of the algorithm varies with the different ordered combinations of the weights of the edges of the graph. The highest precision is observed at $w_r = 0.50$

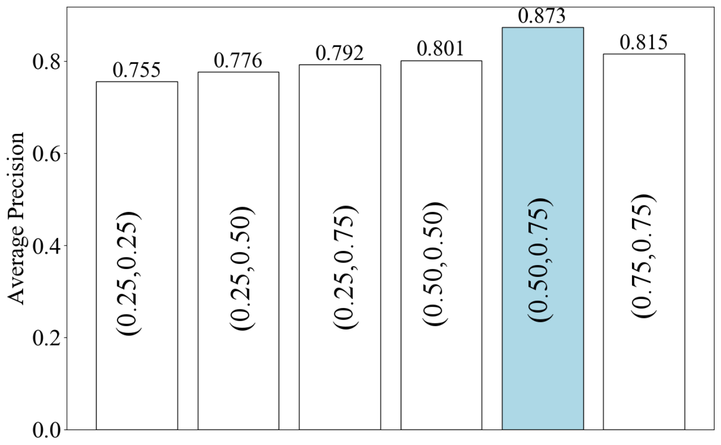

Figure 4.1: **Average precision of CoReRank vs. Index of ordered combinations of edge weights**

and $w_q = 0.75$, which is the optimal parameter combination fixed for our entire experiment.

Definition 2 (**Support Graph**). *A bipartite support graph $G = (U, T, E)$ is a directed bipartite graph where U indicating the set of users forms the left partition, and T indicating the tweets supported by U forms the right partition. Edge $E_{(u,t)}$ connecting user $u \in U$ and tweet $t \in T$ indicates that u supported t with the edge weight $S(u, t)$ denoting the kind of support u extends to t.*

4.1.1 Graph Construction

The graph construction is divided into four distinct steps as follows (we use the same notations mentioned in Section 3).

(i) The users U_b who supported tweets T_b (submitted to Blackmarkets) form the left partition, and the submitted tweets T_b form the right partition of G. U_b and T_b are then connected by directed edges $E_{(U_b,T_b)}$; $S(u, t)$ denotes the edge weight connecting $u \in U_b$ and $t \in T_b$. We call this graph as the upper half G_u of the final graph G.

(ii) The left partition of the graph is further augmented with U_b^v, the verified users in U_b and U_f^v, the followees of U_b^v. The right partition is populated by augmenting T_f^v, the tweets supported by U_f^v. U_f^v and T_f^v are connected by directed edges $E_{(U_f^v,T_f^v)}$ with weights given by $S(.,.)$. We call this as the lower half G_l of the final graph G.

(iii) The next step is to look for possible connections $E_{(U_b,T_f^v)}$ between users in G_u and tweets in G_l. If a user $u \in U_b$ supported a tweet $t \in T_f^v$, a directed edge is added from u and t with weight $S(u, t)$.

(iv) Similarly, we search for possible connections $E_{(U_f^v,T_b)}$ between users in G_l and tweets in G_u. If a user $u \in U_f^v$ supported a tweet $t \in T_b$, a directed edge is added from u and t with weight $S(u, t)$.

After performing the above operation, we obtain the final support graph $G = (U, T, E)$, where

Table 4.1: Statistics of the bipartite support graph G.

	Left partition	**Right partition**	**Edges**
G_u	$\lvert U_b \rvert = 7784$	$\lvert T_b \rvert = 1001$	$\lvert E_{(U_b,T_b)} \rvert = 55382$
G_l	$\lvert U_f^v \rvert = 2667$	$\lvert T_f^v \rvert = 2439319$	$\lvert E_{(U_f^v,T_f^v)} \rvert = 2862793$
G_u to G_l	# nodes in $U_b = 294$	# nodes in $T_f^v = 10512$	$\lvert E_{(U_b,T_f^v)} \rvert = 44466$
G_l to G_u	# nodes in $U_f^v = 10$	# nodes in $T_b = 14$	$\lvert E_{(U_f^v,T_b)} \rvert = 96$
G	$\lvert U \rvert = 10451$	$\lvert T \rvert = 2440320$	$\lvert E \rvert = 2962737$

$U = \{U_b, U_f^v\}$, $T = \{T_b, T_f^v\}$ and $E = \{E_{(U_b,T_b)}, E_{(U_f^v,T_f^v)}, E_{(U_b,T_f^v)}, E_{(U_f^v,T_b)}\}$. G turns out to be a connected graph. Table 4.1 shows the statistics of G.

4.1.2 Intrinsic Traits of Users and Tweets

Users and Tweets have intrinsic traits on the basis of which they can be given a score to determine the 'credibility' (for users) and 'merit' (for tweets). Let $Out(u)$ be the set of tweets user u supported, and $In(t)$ be the set of users that support t. Below we provide a detailed discussion on the interpretation and relevance of these intrinsic properties.

Credibility of Users: The credibility of a user is an indication of how likely they are to support a tweet based on their genuine agreement with the content of the tweet. A highly credible user would only support those tweets which are about the *topics* that they frequently support. In contrast, a user with low credibility, who might be involved in collusive activities would support any tweet that they come across on a blackmarket service. This would highly diversify their timeline w.r.t the topics of supported tweets. Figure 4.2(b) supports this hypothesis. The credibility score of a user u is given by $C(u)$, where $C(u)$ ranges from 0 (very high collusive trait) to 1 (very high credibility trait).

Merit of Tweets: The merit of a tweet is an indication of the genuine organic support of users. A meritorious tweet would be supported by more credible users, indicating that it has a genuine support in the graph and is not earning support because of its submission to a blackmarket service. In contrast, a tweet with lower merit, even though it has a higher user support overall, would still lack the support of users with high credibility. The merit of a tweet t is given by $M(t)$, where $M(t)$ ranges from 0 (highly suspicious) to 1 (highly genuine).

4.2 `CoReRank` Properties

The credibility of user u depends majorly on the merit of the tweets $Out(u)$ that u supported. Similarly, the merit of tweet t depends majorly on the credibility of the users $In(t)$ who supported t. The axioms mentioned below capture this interdependency.

Definition 3 (Inter-support time). *Given a user u, let t_1 and t_2 be two consecutive tweets that u supported at T_1 and T_2, respectively. The time difference between t_1 and t_2 is thus given by*

$T_1 - T_2$. *We construct a set of such time differences for* all *consecutive tweets that u supported, and form a set of inter-support times for u. This set is denoted by $IST(u)$.*

Axiom 1 (**Collusive users have very less inter-support times compared to genuine users**)**.** *Freemium blackmarket services require users to get 'points' to gain retweets and other privileges. In order to keep scoring, these users keep on retweeting random tweets, which leads to very less inter-support times. Hence, the average inter-support time of a collusive user, $avg(IST(u_1))$ is very less than that of a genuine user. Formally,*

$$\exists u_1, u_2 \in U, C(u_1) < C(u_2) \implies avg(IST(u_1)) < avg(IST(u_2))$$

Figure 4.2(a) validates Axiom 1. Collusive users support tweets in a succession, within a minute; whereas genuine users take days or months to support tweets.

Definition 4 (**Identically collusive support of tweets**)**.** *Two tweets, t_1 and t_2, are said to have identically collusive support if, $|In(t_1)| = |In(t_2)|$ and there exists a bijection f from $|In(t_1)|$ to $|In(t_2)|$ such that, $f(u) = u' \implies C(u) = C(u')\ \forall u \in In(t_1), u' \in In(t_2)$.*

Axiom 2 (**Among tweets with identically collusive support, a highly meritorious tweet receives higher support**)**.** *For two tweets, t_1 and t_2 with identically collusive support, if $S(u, t_1) \geq S(u', t_2)$, where $u \in In(t_1) \wedge u' \in In(t_2)$ such that $C(u) = C(u')$, then $M(t_1) \geq M(t_2)$.*

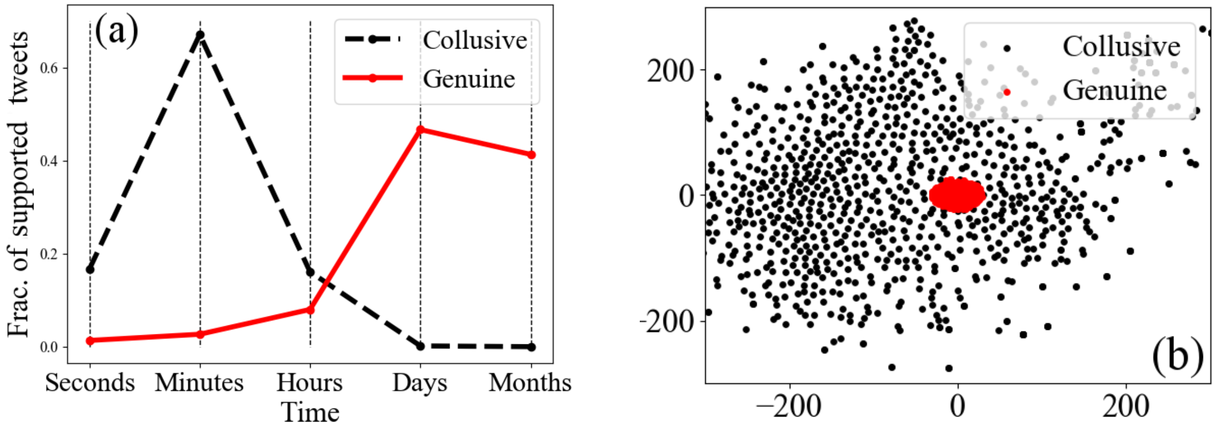

Figure 4.2: (a) Inter-support time for collusive and genuine users. Collusive users are very fast in supporting tweets. (b) Projection of tweets supported by one collusive user and one genuine user. We use t-SNE plot [21] to visualize the tweet space obtained from GloVe embedding (see Section 4.3.2 for details of embedding). Tweets supported by the genuine (collusive) user are on same (different) topic(s).

Axiom 3 (A collusive user associated with blackmarket services demonstrates immense topical diversity). *As per Axiom 1, a collusive user tends to retweet every tweet that can help in receiving more credits. Thus, the tweets supported by that user tend to be very diverse in terms of the topic.*

Figure 4.2(b) supports Axiom 2. We show that the topics of the tweets supported by a collusive user are highly diverse (in the embedding space, points are scattered around the space), whereas the same for a genuine user is uniform (pointers are clustered).

4.3 `CoReRank` Formulation

`CoReRank` is an unsupervised approach which considers a directed bipartite graph of users and tweets. We propose that users have unknown intrinsic scores that quantify how trustworthy they are, and tweets have unknown intrinsic scores that quantify its natural merit of being retweeted organically. Naturally, these scores are interdependent and unknown apriori. Here we start by describing how to obtain seed score which quantifies the intrinsic score of users and tweets. We also define topical diversity score of a user by calculating the inter-support similarity of the tweets s/he retweeted. Finally, we show how one can combine seed score, topical diversity and behavioral activities to jointly obtain the credibility of users and the merit of tweets.

4.3.1 Seed Score

In order to compute the seed scores both for users and tweets that help `CoReRank` propagate further, we use `Birdnest` [15] (as suggested by [16]). `Birdnest` takes the following scores as inputs and uses Bayesian estimates to formulate how much the properties of a user or a tweet deviates from the rest.

- **User-specific Score**, is calculated by providing a vector that contains the inter-support times of the users. This array can be of variable length for each user and is provided as an input to `Birdnest`.
- **Tweet-specific Score**, is calculated with the aid of the length (word count) of the tweet. The length of a retweet (quote) is the length of the original tweet that was retweeted (the sum of the lengths of the original tweet and the quoted text). Each tweet is associated with a vector, whose each entry indicates the length of its retweet/quote. This vector is passed as an input to `Birdnest`. We hypothesize that the text length of a supported tweet is a measure of its behavioural activity [16].

The final outputs of `Birdnest` are the suspicion scores for users $S_U(u)$ and tweets $S_U(T)$. Finally, the seed scores, $\pi_U(u)$ and $\pi_T(t)$ for a user and a tweet, respectively are given by,

$$\pi_U(u) = 1 - S_U(u)\ \forall u \in U$$

$$\pi_T(t) = 1 - S_T(t)\ \forall t \in T$$

In some cases, it might happen that the user (tweet) provided (received) only a single support. In such cases, `Birdnest` does not provide us with a seed score. For these users and tweets, we have assigned the highest possible seed score of 1, giving them the benefit of doubt.

4.3.2 Inter-support Similarity

To calculate the topical diversity score for a user, we first extract information from all the tweets present in our graph G. For each tweet of a user, we split it into words and derive their word representation in embedding space using GloVe embedding trained specifically with Tweets [25]. For this work, we use 100 dimensional pre-trained GloVe vectors. The final embedding of a tweet is obtained by combining the embedding of each word present in the tweet. We do not take into account the words for which the pre-trained word vectors are not found in GloVe. Finally, we measure $\tau_U(u)$, the inter-support similarity for user u by calculating the average cosine similarity of the embeddings corresponding to the tweets supported by u.

4.3.3 Recurrence Formulation

We here propose a systematic approach to combine the following three signals into a recurrence framework – (i) graph-based interdependency between users and tweets, (ii) behavioral activities of users and tweet, and (iii) topical diversity of tweets supported by a user. We also present a way of handling cold start problem.

Graph-based Interdependency of Users and Tweets: As presented in the axioms mentioned previously, the credibility of a user is influenced by the merit of all the tweets that the user supported. Similarly, the merit of a tweet depends on the credibility of the users who supported it. We present below two mathematical formulations that incorporate interdependency among credibility and merit of users and tweets respectively.

For tweets,

$$M(t) = \frac{\sum_{u \in \mathrm{In(t)}} \gamma_{1t} \cdot C(u) \cdot S(u,t)}{\gamma_{1t} + |\mathrm{In(t)}|} \tag{4.1}$$

Similarly for users,

$$C(u) = \frac{\sum_{t \in \mathrm{Out(u)}} \gamma_{1u} \cdot M(t) \cdot S(u,t)}{\gamma_{1u} + |\mathrm{Out(u)}|} \tag{4.2}$$

Here, γ_{1u}, γ_{1t} are constants for users and tweets, respectively. Their values will be learned by parameter sweeping (Section 4.4).

Behavioral Activities: Apart from graph properties, behavioral properties are also included while calculating the credibility and merit of users and tweets. We modify Equations 4.1 and 4.2 to incorporate the seed scores explained in Section 4.3.1 of the users and tweets in the formulation of credibility and merit respectively.

$$M(t) = \frac{\sum_{u \in \mathrm{In(t)}} \gamma_{1t} \cdot C(u) \cdot S(u,t) + \gamma_{2t} \cdot \pi_T(t)}{\gamma_{1t} + \gamma_{2t} + |\mathrm{In(t)}|} \tag{4.3}$$

$$C(u) = \frac{\sum_{t \in \mathrm{Out(u)}} \gamma_{1u} \cdot M(t) \cdot S(u,t) + \gamma_{2u} \cdot \pi_U(u)}{\gamma_{1u} + \gamma_{2u} + |\mathrm{Out(u)}|} \tag{4.4}$$

Here, γ_{2t} and γ_{2u} are also constants for users and tweets respectively and will be learned by parameter sweeping (Section 4.4).

Topic-based Inter-support Similarity: Finally, we add $\tau_U(u)$, the inter-support similarity of user u into the credibility formulation. The lower the inter-support similarity, the higher the probability that the user is collusive. The modified version of Equation 4.3 is as follows:

$$C(u) = \frac{\sum_{t \in \mathrm{Out(u)}} \gamma_{1u} \cdot M(t) \cdot S(u,t) + \gamma_{2u} \cdot \pi_U(u) + \gamma_{3u} \cdot \tau_U(u)}{\gamma_{1u} + \gamma_{2u} + \gamma_{3u} + |\mathrm{Out(u)}|} \tag{4.5}$$

Here, γ_{3u} is a parameter for users. We learn this parameter by performing parameter sweeping as well (Section 4.4).

Handling Cold Start: For some users as well as tweets, multiple supports may not be available (as discussed in Section 4.3.1). Hence, getting seed scores for them is a problem. Also, such users and tweets can lead to biased rankings with high credibility or merit scores. We solve this problem using Laplace smoothing by assigning a default score μ_T to tweets and μ_U to users, weighed by parameters γ_{3t} and γ_{4u} respectively. The value of γ_{3t} and γ_{4u} decides how much importance is given to the default scores – high value implies less influence of the graph structure in the scores of users and tweets. Thus, the modified formulation of merit and credibility is given below.

$$M(t) = \frac{\sum_{u \in \mathrm{In(t)}} \gamma_{1t} \cdot C(u) \cdot S(u,t) + \gamma_{2t} \cdot \pi_T(t) + \gamma_{3t} \cdot \mu_T}{\gamma_{1t} + \gamma_{2t} + \gamma_{3t} + |\mathrm{In(t)}|} \tag{4.6}$$

$$C(u) = \frac{\sum_{t \in \mathrm{Out(u)}} \gamma_{1u} \cdot M(t) \cdot S(u,t) + \gamma_{2u} \cdot \pi_U(u) + +\gamma_{3u} \cdot \tau_U(u) + \gamma_{4u} \cdot \mu_U}{\gamma_{1u} + \gamma_{2u} + \gamma_{3u} + \gamma_{4u} + |\mathrm{Out(u)}|} \tag{4.7}$$

Here, γ_{3t} and γ_{4u} are parameters for tweets and users respectively, and will be learned by parameter sweeping (Section 4.4). The default scores are set as the average of the initial scores of all users and tweets.

Algorithm 1: `CoReRank` Algorithm

```
Input  : $G(U,T,E), \gamma_{1t}, \gamma_{2t}, \gamma_{3t}, \gamma_{1u}, \gamma_{2u}, \gamma_{3u}, \gamma_{4u}$
Output: Credibility and Merit scores for all users and tweets
Calculate $\pi_U(u)\ \forall u \in U$ and $\pi_T(t)\ \forall t \in T$
Initialize $C(u)^0 = \pi_U(u)$ and $M(t)^0 = \pi_T(t)\ \forall u \in U, \forall t \in T$
Initialize $\mu_U = \frac{\sum_{u \in \mathrm{U}} C(u)^0}{|U|}$ and $\mu_T = \frac{\sum_{t \in \mathrm{T}} M(t)^0}{|T|}$
$k = 0$
error = maximum possible integer value
while $error > \epsilon$ do
    $k = k + 1$
    $\tilde{C}^{k-1}(u) = \mathtt{norm}(C^{k-1}(u)) \forall u \in U$ such that $\tilde{C}^{k-1}(u) \in [0,1]$
    Update the merit of tweets using Equation 6: $\forall t \in T$,
    $M^k(t) = \frac{\sum_{u \in \mathrm{In(t)}} \gamma_{1t} \cdot \tilde{C}^{k-1}(u) \cdot S(u,t) + \gamma_{2t} \cdot \pi_T(t) + \gamma_{3t} \cdot \mu_T}{\gamma_{1t} + \gamma_{2t} + \gamma_{3t} + |\mathrm{In(t)}|}$
    Update the credibility of users using Equation 7: $\forall u \in U$,
    $C^k(u) = \frac{\sum_{t \in \mathrm{Out(u)}} \gamma_{1u} \cdot M^k(t) \cdot S(u,t) + \gamma_{2u} \cdot \pi_U(u) + \gamma_{3u} \cdot \tau_U(u) + \gamma_{4u} \cdot \mu_U}{\gamma_{1u} + \gamma_{2u} + \gamma_{3u} + \gamma_{4u} + |\mathrm{Out(u)}|}$
    error = $\max(\max_{u \in U} |C^k(u) - C^{k-1}(u)|, \max_{t \in T} |M^k(t) - M^{k-1}(t)|)$
end
return $C^k(u)\ \forall u \in U$ and $M^k(t)\ \forall t \in T$
```

4.4 The `CoReRank` Algorithm

We now briefly describe the `CoReRank` algorithm (see Algorithm 1 for the pseudo-code). Given $G(U,T,E)$, the support graph of users and tweets, `CoReRank` takes all constants – γ_{1t}, γ_{2t}, γ_{3t}, γ_{1u}, γ_{2u}, γ_{3u}, γ_{4u} as parameters. At first we calculate the seed scores for all the users and tweets, using the `Birdnest` algorithm. For users and tweets for which the seed score could not be found, it is initialized to the highest value, i.e., 1. In the first iteration, the scores for all the users and tweets are initialized to their seed scores. Next, we calculate the cold start constants for users and tweets, i.e., μ_U and μ_T respectively. At the beginning of each iteration, we normalise the credibility scores for users using *Min-Max Normalization*.

Following this, we keep computing scores using Equations 4.6 and 4.7. Convergence is achieved when the maximum of the maximum of differences of scores between $(t+1)^{th}$ and t^{th} iterations, i.e., *error* is less than a very small value, ϵ (set as 10^{-6}).

The convergence factor ϵ is chosen using a grid search in $\{10^{-6}, 10^{-4}, 0.01, 0.1, 0.5, 1\}$. We use *parameter sweep* to tune other parameters – γ_{1t}, γ_{2t}, γ_{3t}, γ_{1u}, γ_{2u}, γ_{3u} and γ_{4u}. In this method, possible combinations of all the parameters from a given list of values are considered. The possible values for γ_{1t}, γ_{2t}, γ_{3t}, γ_{1u}, γ_{2u}, γ_{4u} are taken in the range $[0, 1]$ with a step of 0.3, while γ_{3u} is chosen from $\{0, 1, 2, 3\}$. Figure 4.3 demonstrates how the precision changes due to the affect of different combinations of the parameters. The `x-axis` denotes the combinations of the parameters. The combinations are ordered, in order to facilitate the plotting of the curve. The `y-axis` denotes the precision of the algorithm. As it can be noticed, the precision of the algorithm reaches a global maxima at index 10663. The combination that is represented by this index 10663 is selected as our optimal parameter combination throughout all the experiments. This combination is as follows - $\gamma_{1t} = 0.6$, $\gamma_{2t} = 0.6$, $\gamma_{3t} = 0.3$, $\gamma_{1u} = 0.6$, $\gamma_{2u} = 0.6$, $\gamma_{3u} = 3$ and $\gamma_{4u} = 0.3$.

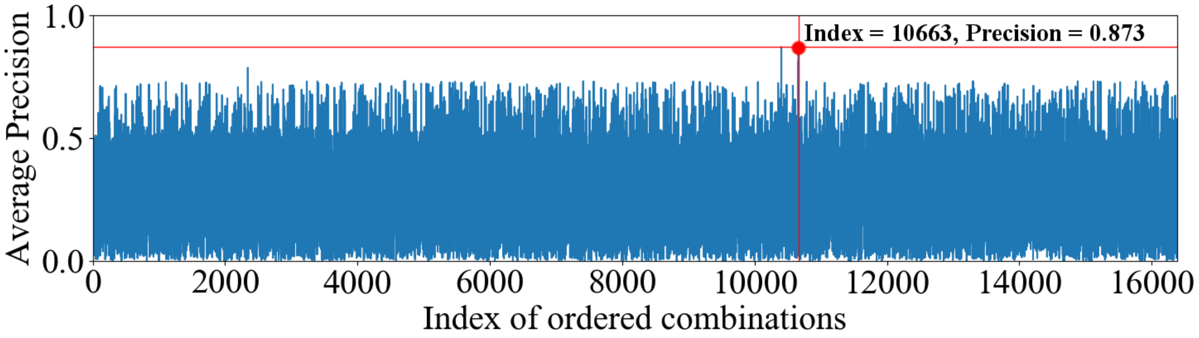

Figure 4.3: **Average precision of CoReRank vs. Index of ordered combinations of parameters**

Output: The outputs of `CoReRank` are ranked lists of users and tweets based on $C(u)$ and $M(t)$ respectively. Ranking in descending (ascending) order of $C(u)$ will place the genuine (collusive) users at the top of the ranking (same for the tweets based on $M(t)$).

4.5 `CoReRank+`: A Semi-supervised Version

Oftentimes, we have partial knowledge about the labels of some users (verified, blackmarket customers, etc.) and tweets. We can leverage such prior information and incorporate them to our formulation in a semi-supervised fashion. We provide each user u with a label score, $\alpha_U(u)$

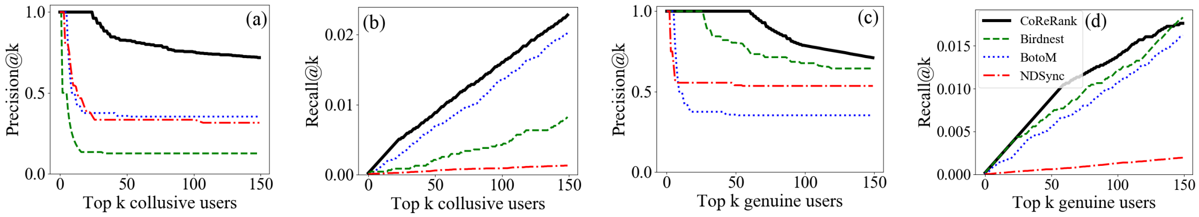

Figure 4.4: Change in performance of the competing unsupervised methods with the increase of k (the number of results returned) for detecting both (a-b) collusive and (c-d) genuine users.

which can be defined as follows -

$$\alpha_U(u) = \begin{cases} \alpha_U^c & \text{if } u \text{ is a collusive user} \\ \alpha_U^g & \text{if } u \text{ is a genuine user} \\ 0 & \text{if } u \text{ has no pre-defined label} \end{cases}$$

Similarly, the label score for a tweet t, $\alpha_T(t)$ can be defined as follows -

$$\alpha_T(t) = \begin{cases} \alpha_T^c & \text{if } t \text{ is a labeled as suspicious} \\ 0 & \text{if } t \text{ has no pre-defined label} \end{cases}$$

We set the values of these constants in such a manner that the genuine users are awarded with a high positive label score and the customers of blackmarket services are given a high negative label score. In our experiment, we set these constants as follows: $\alpha_U^c = -100$, $\alpha_U^g = 100$ and $\alpha_T^c = -100$.

The final equations of credibility and merit after incorporating the labels are given below -

$$M(t) = \frac{\sum_{u \in \text{In(t)}} \gamma_{1t} \cdot C(u) \cdot S(u,t) + \gamma_{2t} \cdot \pi_T(t) + \gamma_{3t} \cdot \mu_T + \alpha_T(t)}{\gamma_{1t} + \gamma_{2t} + \gamma_{3t} + |\text{In(t)}|} \tag{4.8}$$

$$C(u) = \frac{\sum_{t \in \text{Out(u)}} \gamma_{1u} \cdot M(t) \cdot S(u,t) + \gamma_{2u} \cdot \pi_U(u) + +\gamma_{3u} \cdot \tau_U(u) + \gamma_{4u} \cdot \mu_U + \alpha_U(u)}{\gamma_{1u} + \gamma_{2u} + \gamma_{3u} + \gamma_{4u} + |\text{Out(u)}|} \tag{4.9}$$

We design a modified algorithm, called `CoReRank+` by replacing the equations mentioned in Lines 10 and 12 of Algorithm 1 by Equations 4.8 and 4.9.

4.6 Theoretical Guarantee

In this section, we provide the theoretical guarantee to show that `CoReRank` will converge for a given set of inputs. Before we begin, let $C^\infty(u)$ and $M^\infty(t)$ be the final scores of user u and tweet t respectively.

Before we begin, here are the formulas for the credibility scores and merit scores used in the `CoReRank`algorithm to calculate scores for the k^{th} iteration:

$$\tilde{C}^{k-1}(u) = \texttt{norm}(C^{k-1}(u)) \forall u \in U \textit{ such that } \tilde{C}^{k-1}(u) \in [0, 1] \tag{4.10}$$

$$M^k(t) = \frac{\sum_{u \in \text{In(t)}} \gamma_{1t} \cdot \tilde{C}^{k-1}(u) \cdot S(u,t) + \gamma_{2t} \cdot \pi_T(t) + \gamma_{3t} \cdot \mu_T}{\gamma_{1t} + \gamma_{2t} + \gamma_{3t} + |\text{In(t)}|} \tag{4.11}$$

$$C^k(u) = \frac{\sum_{t \in \text{Out(u)}} \gamma_{1u} \cdot M^k(t) \cdot S(u,t) + \gamma_{2u} \cdot \pi_U(u) + +\gamma_{3u} \cdot \tau_u + \gamma_{4u} \cdot \mu_U}{\gamma_{1u} + \gamma_{2u} + \gamma_{3u} + \gamma_{4u} + |\text{Out(u)}|} \tag{4.12}$$

where $\gamma_{1t}, \gamma_{2t}, \gamma_{3t}, \gamma_{1u}, \gamma_{2u}, \gamma_{3u}, \gamma_{4u}$ are constants provided as inputs to the algorithm. Also, it is important to note that $S(u,t) \leq \frac{3}{4}$, as maximum weight of an edge can be $w_q = \frac{3}{4}$.

Then we have the following results:

[**Lemma 1**] For any given tweet, t, the difference between their final score and their score after the first iteration of `CoReRank` cannot exceed $\frac{3}{4}$. Formally, it means that $|M^\infty(t) - M^1(t)| \leq \frac{3}{4}$. Similarly, for users the upper bound is $\frac{3}{4}$, i.e., $|C^\infty(u) - C^1(u)| \leq \frac{3}{4}$

Proof.

Let us first prove that $|M^\infty(t) - M^1(t)| \leq \frac{3}{4}$.

From 4.11, we know that when the algorithm converges,

$$M^\infty(t) = \frac{\sum_{u \in \text{In(t)}} \gamma_{1t} \cdot \tilde{C}^\infty(u) \cdot S(u,t) + \gamma_{2t} \cdot \pi_T(t) + \gamma_{3t} \cdot \mu_T}{\gamma_{1t} + \gamma_{2t} + \gamma_{3t} + |\text{In(t)}|}$$

Similarly after the first iteration,

$$M^1(t) = \frac{\sum_{u \in \text{In(t)}} \gamma_{1t} \cdot \tilde{C}^0(u) \cdot S(u,t) + \gamma_{2t} \cdot \pi_T(t) + \gamma_{3t} \cdot \mu_T}{\gamma_{1t} + \gamma_{2t} + \gamma_{3t} + |\text{In(t)}|}$$

Hence, substituting values in $|M^\infty(t) - M^1(t)|$, we get,

$$|M^\infty(t) - M^1(t)| = |\frac{\sum_{u \in \text{In(t)}} \gamma_{1t} \cdot (\tilde{C}^\infty(u) - \tilde{C}^0(u)) \cdot S(u,t)}{\gamma_{1t} + \gamma_{2t} + \gamma_{3t} + |\text{In(t)}|}|$$

Since $|x + y| \leq |x| + |y|$,

$$|M^\infty(t) - M^1(t)| \leq \frac{\sum_{u \in \text{In(t)}} |\gamma_{1t} \cdot (\tilde{C}^\infty(u) - \tilde{C}^0(u)) \cdot S(u,t)|}{\gamma_{1t} + \gamma_{2t} + \gamma_{3t} + |\text{In(t)}|}$$

As $|x \cdot y| = |x| \cdot |y|$,

$$|M^{\infty}(t) - M^{1}(t)| \leq \frac{\sum_{u\in\mathrm{In(t)}} |\gamma_{1t}| \cdot |(\tilde{C}^{\infty}(u) - \tilde{C}^{0}(u))| \cdot |S(u,t)|}{\gamma_{1t} + \gamma_{2t} + \gamma_{3t} + |\mathrm{In(t)}|}$$

Since $\tilde{C}^{k}(u) \in [0,1] \forall k \in [0,\infty), \forall u \in U$, $|(\tilde{C}^{\infty}(u) - \tilde{C}^{0}(u))| \leq 1$.

Also, $S(u,t) \leq \frac{3}{4}$ and $\gamma_{1t} \in [0,1)$

$$\therefore |M^{\infty}(t) - M^{1}(t)| \leq \frac{\sum_{u\in\mathrm{In(t)}} \gamma_{1t} \cdot 1 \cdot \frac{3}{4}}{\gamma_{1t} + \gamma_{2t} + \gamma_{3t} + |\mathrm{In(t)}|}$$

$$\therefore |M^{\infty}(t) - M^{1}(t)| \leq \frac{\gamma_{1t} \cdot |In(t)| \cdot \frac{3}{4}}{\gamma_{1t} + \gamma_{2t} + \gamma_{3t} + |\mathrm{In(t)}|} \leq \frac{3}{4}$$

Similarly, it is possible to show that $|C^{\infty}(u) - C^{1}(u)| \leq \frac{3}{4}$

Theorem 1 (**Theorem of convergence**)**.** *Between successive iterations, the difference in the scores of the users and tweets is bounded. For any user* $u \in U$*,* $|C^{\infty}(u) - C^{k}(u)| \leq \left(\frac{3}{4}\right)^{k}$*. Thus, as the algorithm proceeds through more and more iterations, the value of* k *keeps on increasing and the difference in score from the final score keeps on decreasing. Similarly for a tweet* $t \in T$*,* $|M^{\infty}(t) - M^{k}(t)| \leq \left(\frac{3}{4}\right)^{k-1}$*.*

Proof.
We will prove these using induction on k.
Base Cases (**$k = 1$**)**:** We know from Lemma 4.6, $\forall u \in U, \forall t \in T$

$$|C^{\infty}(u) - C^{1}(u)| \leq \frac{3}{4}$$

and

$$|M^{\infty}(t) - M^{1}(t)| \leq \frac{3}{4}$$

Hence, the base cases are satisfied.
Induction Hypothesis: Assume that for any $n \leq k$, we have ,

$$|C^{\infty}(u) - C^{k}(u)| \leq \big(\frac{3}{4}\big)^{k}$$

and

$$|M^{\infty}(t) - M^{k}(t)| \leq \big(\frac{3}{4}\big)^{k-1}$$

$\forall u \in U, \forall t \in T$.

Induction Step (**$k = n + 1$**)**:** For $k = n + 1$, we have,

$$|M^{\infty}(t) - M^{n+1}(t)| = |\frac{\sum_{u \in \mathrm{In(t)}} \gamma_{1t} \cdot (\tilde{C}^{\infty}(u) - \tilde{C}^{n}(u)) \cdot S(u,t)}{\gamma_{1t} + \gamma_{2t} + \gamma_{3t} + |\mathrm{In(t)}|}|$$

$$\implies |M^{\infty}(t) - M^{n+1}(t)| \leq \gamma_{1t} \frac{\sum_{u \in \mathrm{In(t)}} |(\tilde{C}^{\infty}(u) - \tilde{C}^{n}(u))| \cdot |S(u,t)|}{\gamma_{1t} + \gamma_{2t} + \gamma_{3t} + |\mathrm{In(t)}|}$$

As $|(\tilde{C}^{\infty}(u) - \tilde{C}^{n}(u))| \leq |C^{\infty}(u) - C^{n}(u)| \leq \left(\frac{3}{4}\right)^{n}$,

$$\implies |M^{\infty}(t) - M^{n+1}(t)| \leq \frac{\gamma_{1t} \cdot \left(\frac{3}{4}\right)^{n} \cdot |\mathrm{In(t)}| \cdot |S(u,t)|}{\gamma_{1t} + \gamma_{2t} + \gamma_{3t} + |\mathrm{In(t)}|} \leq \big(\frac{3}{4}\big)^{n}$$

Thus, $\forall t \in T$, $|M^{\infty}(t) - M^{k}(t)| \leq \left(\frac{3}{4}\right)^{k-1}$ by proof of induction.

Similarly,

$$|C^{\infty}(u) - C^{n+1}(u)| = |\frac{\sum_{t \in \mathrm{Out(u)}} \gamma_{1u} \cdot (M^{\infty}(t) - M^{n+1}(t)) \cdot S(u,t)}{\gamma_{1u} + \gamma_{2u} + \gamma_{3u} + \gamma_{4u} + |\mathrm{Out(u)}|}|$$

$$\implies |C^{\infty}(u) - C^{n+1}(u)| \leq \frac{\sum_{t \in \text{Out(u)}} |\gamma_{1u}| \cdot |(M^{\infty}(t) - M^{n+1}(t))| \cdot |S(u,t)|}{\gamma_{1u} + \gamma_{2u} + \gamma_{3u} + \gamma_{4u} + |\text{Out(u)}|}$$

$$\implies |C^{\infty}(u) - C^{n+1}(u)| \leq \frac{|\gamma_{1u}| \cdot \left(\frac{3}{4}\right)^{n} \cdot |S(u,t)|}{\gamma_{1u} + \gamma_{2u} + \gamma_{3u} + \gamma_{4u} + |\text{Out(u)}|}$$

Since $S(u,t) \leq \frac{3}{4}$,

$$\implies |C^{\infty}(u) - C^{n+1}(u)| \leq \frac{|\gamma_{1u}| \cdot \left(\frac{3}{4}\right)^{n+1}}{\gamma_{1u} + \gamma_{2u} + \gamma_{3u} + \gamma_{4u} + |\text{Out(u)}|} \leq \left(\frac{3}{4}\right)^{n+1}$$

Thus, $\forall u \in U$, $|C^{\infty}(u) - C^{k}(u)| \leq \left(\frac{3}{4}\right)^{k}$ by proof of induction.

This completes our proof.

Theorem 2 (Bound on iterations). *There exists a bound on the number of iterations until* ***CoReRank*** *converges. This bound is governed by the precision to which the score is calculated before convergence is declared, i.e.,* ϵ. *The number of iterations till convergence is at most* $2 + \lceil \frac{\log\left(\frac{\epsilon}{2}\right)}{\log\left(\frac{3}{4}\right)} \rceil$

Proof: Let $k = 2 + \lceil \frac{\log\left(\frac{\epsilon}{2}\right)}{\log\left(\frac{3}{4}\right)} \rceil$. By Theorem 1, after $k+1$ iterations, $\forall t \in \mathcal{T}$, $|M^{\infty}(t) - M^{k+1}(t)| \leq \frac{3}{4}^{k} \leq \frac{3}{4}^{\log_{\frac{3}{4}}(\frac{\epsilon}{2})} = \frac{\epsilon}{2}$. Similarly, $M^{\infty}(t) - M^{k+2}(t)| \leq \frac{\epsilon}{2} \cdot \frac{3}{4} \leq \frac{\epsilon}{2}$. Thus,

$$|M^{k+1}(t) - M^{k+2}(t)| = |M^{k+1}(t) - M^{\infty}(t) + M^{\infty}(t) - M^{k+2}(t)|$$

As $|x + y| \leq |x| + |y|$,

$\implies |M^{k+1}(t) - M^{k+2}(t)| \leq |M^{k+1}(t) - M^{\infty}(t)| + |M^{\infty}(t) - M^{k+2}(t)| \leq 2 \cdot \frac{\epsilon}{2} = \epsilon$

Similarly for credibility, we have, $|C^{k+1}(u) - C^{k+2}(u)| \leq \epsilon \forall u \in \mathcal{U}$.

Thus, by line 6 of Algorithm 1 it will take $k + 2$ iterations to converge.

Hence, a low value of ϵ would require a larger number of iterations for the algorithm to converge.

Time Complexity of `CoReRank`: In each iteration, `CoReRank` updates the scores of users and tweets in constant time. Thus, the complexity of each iteration is O(|E| + |V|). Since $|E| >> |V|$ in $G(U, T)$, O(|E| + |V|) ≈ O(|E|). As explained in Theorem 2, `CoReRank` converges in a constant number of iterations. Let n be the product of the number of iterations till convergence and the number of runs of `CoReRank`. Thus, the time complexity of algorithm is O(n|E|), which is linear in the number of edges of $G(U, T)$ (as supported empirically in Figure 5.1(d)).

Chapter 5

Experimental Results

This section starts by briefly explaining the baselines and human annotation process, followed by the detailed performance analysis of the competing methods.

5.1 Baseline Methods

As mentioned before, `CoReRank` is the second algorithm after `SCoRe` which addresses the problem of detecting collusive users involved in fake retweeting activities. We choose six state-of-the-art (supervised and unsupervised) methods as baselines, which are similar to the problem we address here. (i) **Baseline I** (`BotoM`): [5] measured bot score for each user and produced a ranked list of users. (ii) **Baseline II** (`NDSync`): [11] detected synchronous retweet fraudulent activities by calculating a user-level suspiciousness score which combines the suspiciousness score for each retweet thread by projecting them into a multi-dimensional feature space. (iii) **Baseline III** (`Birdnest`): [15] detected fraudulent reviewers by combining the rating and temporal information and generating a likelihood-based suspiciousness metric for users and reviews. We adopted `Birdnest` to rank users and tweets.

The **supervised approaches** are as follows. (i) **Baseline IV** (`SpamBot`): [32] detected spam bots using a set of content-based and graph-based features. The proposed set of features is used in Naive Bayes classifier to classify a user into collusive or genuine. (ii) **Baseline V** (`FakeAcc`): [9] used a classification method to detect fake accounts on Twitter based on minimum weighted feature set calculated over 22 features. The feature set is then applied on several classifiers among which SVM performed the best. (iii) **Baseline VI** (`SCoRe`): [7] is the closest baseline of our method which identified blackmarket customers by running SVM using a set of 64 features.

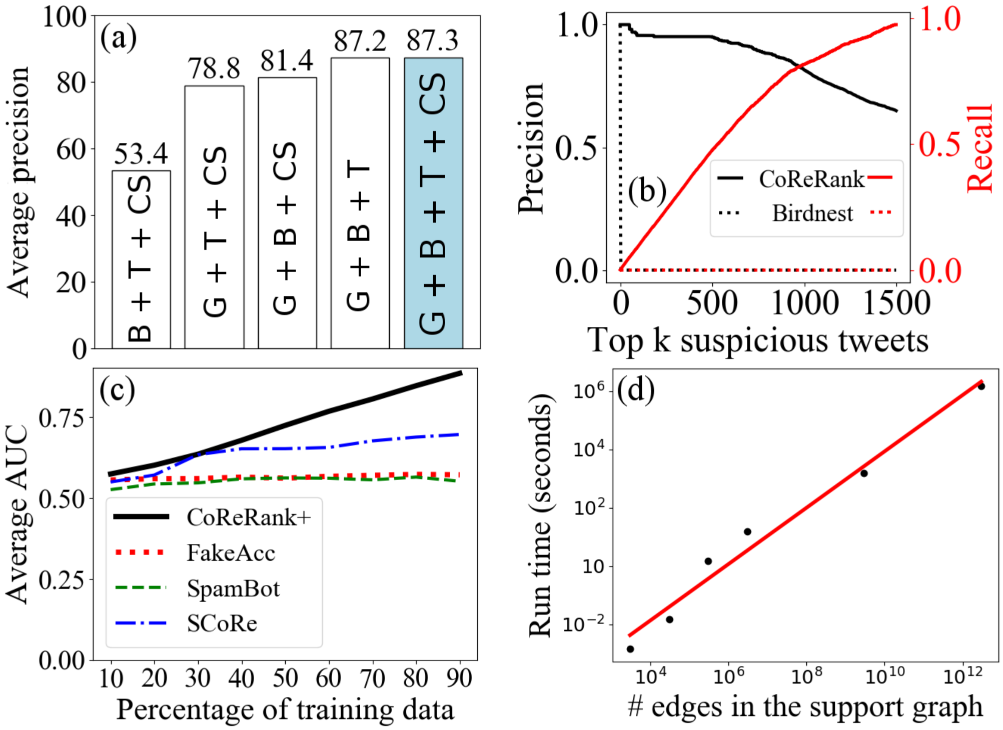

Figure 5.1: (a) Importance of different components of the recurrence formulation – graph (G), behavior (B), topic (T) and cold start (CS). (b) Precision and recall of `CoReRank` and `Birdnest` in detecting suspicious tweets. (c) Average AUC with different percentage of training data. (d) Scalability analysis of `CoReRank`.

5.2 Annotating Unknown Users

We asked three human annotators[1] to label 7,451 unknown users $\mathcal{S}_u$ (mentioned in Section 3) into *collusive* or *genuine*. Annotators were given the definition of collusive users and Twitter Terms of Service. They were also given complete freedom to search for more information related to the (collusive) users on the web and apply their intuitions. The following guidelines were also given to them:

(1) Collusive users are members of blackmarket services who retweet/quote the tweets submitted to blackmarket services to earn credits in return.

(2) To earn credits from the blackmarket services, collusive users tend to show more aggressive behavior in terms of retweeting activity. Moreover, collusive users are less likely to retweet content of their friends.

(3) Generally, tweets submitted to the blackmarket services are related to promotions of tweets, accounts or services [8].

(4) Users created by the blackmarket services will only be involved in retweeting other tweets rather than publishing their own tweets. The annotators were asked to check the retweet statistics such as number of retweets, inter-retweet times, etc.

[1] Annotators were experts in social media, and their age ranged between 25-35.

We found high inter-annotator agreement (Fleiss' kappa coefficient of 0.75). We finally considered 4732 users as collusive and 2719 users as genuine (for which at least two annotators agreed). The newly labeled sets were augmented with the sets of ground-truth users $\mathcal{S}_c$ and $\mathcal{S}_g$ (as described in Section 3).

5.3 Comparative Evaluation

We observe the performance of the competing unsupervised methods with the increase of k, number of results returned. Figure 4.4 shows that till $k = 30$ ($k = 60$), precision of `CoReRank` is almost 1 for collusive (genuine) user detection; even after this, the decrease in performance is significantly less for `CoReRank` compared to other methods. The recall curve corresponding to `CoReRank` also increases steadily with k compared to other methods.

Table 5.1 shows the performance of unsupervised and (semi-) supervised methods separately. We observe that `CoReRank` beats the best baseline (`Birdnest`) with 269% high average precision and 300% high average recall for collusive user detection. For genuine user detection, its improvement is 20% and 22.22% higher than `Birdnest` based on average precision and average recall respectively. The performance of (semi-) supervised methods is reported separately after averaging over 10-fold cross validation. For `CoReRank+`, we choose 0.202 as the threshold of $C(u)$; users whose corresponding $C(u)$ values are higher than (lower than or equal to) the threshold are considered as genuine (collusive). We observe that `CoReRank+` beats the best baseline (`SCoRe`) by 33.18% in terms of AUC (similar pattern is observed for other evaluation metrics). The thresholding in the following manner – in Figure 5.2, we plotted the non-cumulative distribution of credibility scores for all users ($C(u) \forall u \in U$) against their allotted ranks. At $x = 4248$, we encounter the **Sharpest Turning Point** of the curve. Corresponding to $x = 4248$, $C(u) = 0.202$. A threshold margin is then created at $x = 4248$. All the users whose rank lies before 4248 are termed as `collusive` and all those whose rank is after 4248 are termed as `genuine`.

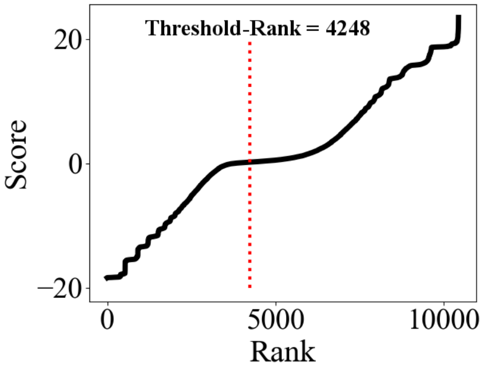

Figure 5.2: **Non-cumulative distribution of credibility scores vs. rank of users after algorithm**

Robustness Analysis: We also report the performance of the competing (semi-)supervised methods with the increase of training size to show the robustness of the methods w.r.t the size of the training set. We vary the training data from 10% to 90%. Figure 5.1(c) shows the AUC on test sets, averaged over 50 random iterations of training data. We observe that `CoReRank+` always outperforms others for all training percentage. This shows the efficiency of our framework even when a small amount of training data is available.

Table 5.1: Performance of the competing methods. We show the accuracy separately for unsupervised (in terms of Average Precision (AP) and Average Recall (AR)) and (semi-) supervised (in terms of ROC-AUC, Precision (P) and Recall (R), averaged over 10-fold cross validation) methods.

Metric	`BotoM`	`NDSync`	`Birdnest`	`CoReRank`
AP (Collusive)	0.212	0.218	0.221	**0.817**
AR (Collusive)	0.006	0.010	0.003	**0.012**
AP (Genuine)	0.394	0.532	0.727	**0.875**
AR (Genuine)	0.005	0.009	0.009	**0.011**
Metric	`SpamBot`	`FakeAcc`	`SCoRe`	`CoReRank+`
AUC	0.573	0.578	0.696	**0.927**
P (Collusive)	0.611	0.589	0.724	**0.933**
R (Collusive)	0.336	0.362	0.727	**0.887**
P (Genuine)	0.547	0.559	0.603	**0.910**
R (Genuine)	0.790	0.762	0.599	**0.947**

5.4 Importance of Different Components

In order to understand the importance of different components in the recurrence formulation of `CoReRank`, we drop each component in isolation (set its corresponding coefficient as 0) and measure the accuracy. Figure 5.1(a) shows maximum decrease in accuracy (38%) after removing graph (G) component, followed by behavior (9%), topic (6.75%) and cold start (0.11%). However, removing none of the components increases the accuracy, indicating that all the components are important.

5.5 Suspicious Tweet Detection

One of the advantages of `CoReRank` is that apart from ranking users based on their collusive activities, it can also rank tweets based on the merit score, which none of the baselines (except `Birdnest`) can. We take 1001 tweets which we collected from the blackmarket services as ground-truth suspicious tweets[2]. Figure 5.1(b) shows the precision and recall of `CoReRank` and `Birdnest` with the number of tweets returned (k). `CoReRank` achieves 0.85 average precision and 0.60 average recall, whereas both values for `Birdnest` is 0. In the ranked list that `Birdnest` returns, the first suspicious tweet appears at 3485^{th} position.

[2] We did not show the results for genuine tweet detection as we were not sure about the ground-truth genuine tweets.

5.6 Scalability Analysis

Theorem 2 already showed that the running time of CoReRank scales linearly in the number of edges in the bipartite support graph. To show this empirically, we take the complete bipartite graph and keep adding edges (within the range of $10^4 - 10^{12}$) randomly without changing the number of nodes. Figure 5.1(d) shows that the running time increases linearly with the number of edges. On the dataset we collected, the average running time over 50 iterations is 170 seconds, which is much faster than BotoM (14400 seconds), NDSync (3000 seconds) and Birdnest (1200 seconds)[3]. We implemented CoReRank in Python using the Pandas library [22] for efficiency. All experiments were executed on a 1.8 GHz Intel Core i5 Macbook Air, 8 GB DDR3 RAM, running macOS High Sierra v10.13.3.

[3] We did not report the running time of supervised methods as it may not be appropriate to compare two different classes of methods.

Chapter 6

Conclusion

We presented CoReRank, the first unsupervised framework to simultaneously detect collusive users and suspicious tweets, controlled by blackmarket retweeting services. We showed the effectiveness of our framework over six other baselines on our own (manually curated and annotated) dataset. We also provided theoretical guarantees to show the convergence of CoReRank. Its runtime scales linearly with the number of edges, and it is faster than other competing methods. The dataset we collected is also the first dataset of this kind. We also made the code and dataset available for the purpose of reproducibility.

Bibliography

[1] Ahmed, F., and Abulaish, M. A generic statistical approach for spam detection in online social networks. *Computer Communications 36*, 10-11 (2013), 1120–1129.

[2] Benevenuto, F., Magno, G., Rodrigues, T., and Almeida, V. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)* (2010), vol. 6, p. 12.

[3] Chu, Z., Gianvecchio, S., Wang, H., and Jajodia, S. Who is tweeting on twitter: human, bot, or cyborg? In *Proceedings of the 26th annual computer security applications conference* (2010), ACM, pp. 21–30.

[4] Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., and Tesconi, M. Fame for sale: efficient detection of fake twitter followers. *Decision Support Systems 80* (2015), 56–71.

[5] Davis, C. A., Varol, O., Ferrara, E., Flammini, A., and Menczer, F. Botornot: A system to evaluate social bots. In *Proceedings of the 25th International Conference Companion on World Wide Web* (2016), International World Wide Web Conferences Steering Committee, pp. 273–274.

[6] De Micheli, C., and Stroppa, A. Twitter and the underground market. In *11th Nexa Lunch Seminar* (2013), vol. 22.

[7] Dutta, H., Chetan, A., Joshi, B., and Chakraborty, T. Retweet us, we will retweet you: Spotting collusive retweeters involved in blackmarket services. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (2018), pp. 242–249.

[8] Eftekhar, M., and Koudas, N. Some research opportunities on twitter advertising. *IEEE Data Eng. Bull. 36*, 3 (2013), 77–82.

[9] ElAzab, A. Fake accounts detection in twitter based on minimum weighted feature. *World* (2016).

[10] Gao, H., Chen, Y., Lee, K., Palsetia, D., and Choudhary, A. N. Towards online spam filtering in social networks. In *NDSS* (2012), vol. 12, pp. 1–16.

[11] Giatsoglou, M., Chatzakou, D., Shah, N., Beutel, A., Faloutsos, C., and Vakali, A. Nd-sync: Detecting synchronized fraud activities. In *PAKDD* (2015), pp. 201–214.

[12] Giatsoglou, M., Chatzakou, D., Shah, N., Faloutsos, C., and Vakali, A. Retweeting activity on twitter: Signs of deception. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (2015), Springer, pp. 122–134.

[13] Grier, C., Thomas, K., Paxson, V., and Zhang, M. @ spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security* (2010), ACM, pp. 27–37.

[14] Gupta, S., Khattar, A., Gogia, A., Kumaraguru, P., and Chakraborty, T. Collective classification of spam campaigners on twitter: A hierarchical meta-path based approach. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018* (2018), pp. 529–538.

[15] Hooi, B., Shah, N., Beutel, A., Günnemann, S., Akoglu, L., Kumar, M., Makhija, D., and Faloutsos, C. Birdnest: Bayesian inference for ratings-fraud detection. In *Proceedings of the 2016 SIAM International Conference on Data Mining* (2016), SIAM, pp. 495–503.

[16] Kumar, S., Hooi, B., Makhija, D., Kumar, M., Faloutsos, C., and Subrahmanian, V. Rev2: Fraudulent user prediction in rating platforms. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (New York, NY, USA, 2018), WSDM '18, ACM, pp. 333–341.

[17] Kwak, H., Lee, C., Park, H., and Moon, S. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web* (2010), ACM, pp. 591–600.

[18] Lancaster, E., Chakraborty, T., and Subrahmanian, V. Maltp: Parallel prediction of malicious tweets. *IEEE Transactions on Computational Social Systems* (2018).

[19] Lee, K., Caverlee, J., and Webb, S. Uncovering social spammers: social honeypots+ machine learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (2010), ACM, pp. 435–442.

[20] Liu, Y., Liu, Y., Zhang, M., and Ma, S. Pay me and i'll follow you: Detection of crowdturfing following activities in microblog environment. In *IJCAI* (2016), pp. 3789–3796.

[21] Maaten, L. v. d., and Hinton, G. Visualizing data using t-sne. *JMLR 9*, Nov (2008), 2579–2605.

[22] McKinney, W. pandas: a foundational python library for data analysis and statistics. *Python for High Performance and Scientific Computing* (2011), 1–9.

[23] Miller, Z., Dickinson, B., Deitrick, W., Hu, W., and Wang, A. H. Twitter spammer detection using data stream clustering. *Information Sciences 260* (2014), 64–73.

[24] Motoyama, M., McCoy, D., Levchenko, K., Savage, S., and Voelker, G. M. An analysis of underground forums. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference* (2011), ACM, pp. 71–80.

[25] Pennington, J., Socher, R., and Manning, C. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), pp. 1532–1543.

[26] Shah, N. Flock: Combating astroturfing on livestreaming platforms. In *Proceedings of the 26th International Conference on World Wide Web* (2017), International World Wide Web Conferences Steering Committee, pp. 1083–1091.

[27] Shah, N., Lamba, H., Beutel, A., and Faloutsos, C. The many faces of link fraud. In *Data Mining (ICDM), 2017 IEEE International Conference on* (2017), IEEE, pp. 1069–1074.

[28] Singh, M., Bansal, D., and Sofat, S. Followers or fradulents? an analysis and classification of twitter followers market merchants. *Cybernetics and Systems 47*, 8 (2016), 674–689.

[29] Stringhini, G., Wang, G., Egele, M., Kruegel, C., Vigna, G., Zheng, H., and Zhao, B. Y. Follow the green: growth and dynamics in twitter follower markets. In *Proceedings of the 2013 conference on Internet measurement conference* (2013), ACM, pp. 163–176.

[30] Thomas, K., McCoy, D., Grier, C., Kolcz, A., and Paxson, V. Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse. In *USENIX Security Symposium* (2013), pp. 195–210.

[31] Vidros, S., Kolias, C., Kambourakis, G., and Akoglu, L. Automatic detection of online recruitment frauds: Characteristics, methods, and a public dataset. *Future Internet 9*, 1 (2017), 6.

[32] Wang, A. H. Detecting spam bots in online social networking sites: a machine learning approach. In *IFIP Annual Conference on Data and Applications Security and Privacy* (2010), Springer, pp. 335–342.

[33] Wang, B., Zubiaga, A., Liakata, M., and Procter, R. Making the most of tweet-inherent features for social spam detection on twitter. *arXiv preprint arXiv:1503.07405* (2015).

[34] Wang, D., Navathe, S. B., Liu, L., Irani, D., Tamersoy, A., and Pu, C. Click traffic analysis of short url spam on twitter. In *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on* (2013), IEEE, pp. 250–259.

[35] Yang, C., Harkreader, R., Zhang, J., Shin, S., and Gu, G. Analyzing spammers' social networks for fun and profit: a case study of cyber criminal ecosystem on twitter. In *Proceedings of the 21st international conference on World Wide Web* (2012), ACM, pp. 71–80.