JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 37, 935-958 (2021)
DOI: 10.6688/JISE.202107_37(4).0012

Mining Influential Who-to-Post and When-to-Post Curators on Social Networks

CHIEH-CHENG HSIA AND CHENG-TE LI
Department of Statistics
Institute of Data Science
National Cheng Kung University
Tainan, 701 Taiwan
E-mail: cchsia@gs.ncku.edu.tw; chengte@mail.ncku.edu.tw

Curators on the social networking sites become prominent and indispensable nowadays. Gradually, they come to be the voice in the business's online marketing field. The problem that how to find the most future-influential curators and plan the best posting time for them, notwithstanding, has been hidden and under-explored as yet. In this study, we initiate to analyze this problem with those two primary concerns from four distinct dimensions. To find the most future-influential curators, we consider this problem from the following two dimensions, Future Influence Ranking Prediction and Future Influential Leader Prediction. To plan the best posting time for the curators, similarly, we consider this part with two dimensions below, Accumulated Influence Post-time Scheduling and Limited Influence Post-time Scheduling. We aim at predicting the future influential curator with a series of basic and advanced self-defined features. Based on network embedding, we add learned features to capture the connection between users. To deal with the problem, we implement Learning-to-Rank algorithm and two newly devised ones, self-training algorithm and mutual-training algorithm, which are served to become the solution to the imbalanced data. With the experiments on large-scale Facebook data, we find the proposed methods significantly outperform the conventional prediction settings. The F1 score in predicting the most future influential curators can be up to 0.875. Also, in the part of planning the best posting time, the result shows in comparison with the overall performance of curators, the limited influence of the curators in our planned time can be boosted up to three times.

Keywords: social network analysis, feature engineering, influence prediction, node embedding, when-to-post scheduling

1. INTRODUCTION

With enormous growth potential in online marketing field these years, curators on social networking sites become prominent and indispensable. Curators interact with the audience, listen to the voice from the audience, and promote some products or the image of the brand at the best time to maximize the profit of their businesses. Recently, some companies attempt to expand their promotion influence and hence, begin to recruit influential users for friendvertising. Nonetheless, with regard to some limited-time offer, it is necessary for businesses to maximize their future influence in a short time window. Thus,

Received July 16, 2019; revised February 2, 2020; accepted March 26, 2020.
Communicated by Tzung-Pei Hong.

it comes to the problem that how to find the most future-influential curators and simultaneously plan the best posting time for them to maximize the promotion influence on their friends.

A large body of this study is pertinent to the problem of influence maximization and prediction. Kempe *et al.* [1] brought up the concept of influence maximization that sought to find seeds to maximize the influence under information cascade models. Influence maximization, yet, is an information cascade simulation and try to estimate the influence using history data; therefore, it fails to appropriately apply to the real scenarios – predicting future influence. Bakshy *et al.* [2] initiated the first study to predict the influence of users on social networking sites. They extracted seed user attributes along with past influence of seed users to predict the average number of reshares per user by regression tree model. On the contrary, in our study, we only need the history activity of users, that is, the posting and response activity, which is enough for us to dig up the information from it to predict the users' future influence (the number of future responses). Finally, the related research about posting time scheduling is Spasojevic *et al.* [3]. They were the first to formulate when-to-post problem in order to maximize the audience response probability for a given user. Still, they predict with history data, which they failed to take future influence into account and only planned the post-time for a given user.

In our work, we take future influence into consideration and analyze the problem by discussing two main sub-problems from four distinct dimensions. Under the most future influential curators prediction, we propose Future Influence Ranking Prediction (FIRP) and Future Influential Leader Prediction (FILP) problems. To deal with these two problems, we present a self-training algorithm and a mutual-training algorithm to solve the problem occurred in imbalanced data under FIRP and FILP. To schedule the best post-time for curators, we implement PastGreedy algorithm (PG) and Forecasting regression prediction (FC) and consider the unlimited and limited time scenarios. Excitingly, the experimental results show that the prediction ability of future-influential curators achieves up to 0.875 in F1 score. Additionally, compared with the overall performance of the curators, the limited influence of the curators in our planned time improves around threefold.

2. RELATED WORK

This study contains the problem of future-influential curators mining and the best post-time scheduling, where the related work of influential curators mining can be categorized into three regarding research subjects, influence maximization, influence quantification and influence prediction.

2.1 Influence Maximization

Influence Maximization [1, 4, 5, 6] is the basis of Viral Marketing which aims to find a few influential opinion leaders and sell the products via them to their online communities for the purpose of reaching the most people on social networks. Technically speaking, given a social network G, an information cascade model ex. Independent Cascade Model or Linear Threshold Model, and a budget of k seed nodes, the objective of the influence maximization problem [1] is to find some k seed nodes which can expand their influence to activate the most nodes under an information cascade model. Recently, influence maximization problem has generated scores of studies. Goyal *et al.* [7] learned the activating probability within peers from the history of post influence to ensure the accuracy of influence spread simulation. Bhagat *et al.* [8] used user rating to examine the problem of product-adoption maximization. Leskovec *et al.* [9] proposed Cost-effective Outbreak Detection to detect influence spread with the least delay. Lappas *et al.* [10] managed to find which nodes are effectors given a social network and influenced nodes. Li *et al.* [11] identified a set of key players who control the bottlenecks of influence propagation in a social network. Li *et al.* [12, 13] jointly forecasted the activated users and selected the influential users from the perspective of information diffusion data. A framework [14] that combines influence maximization and team formation is also developed to find influential teams in event organization.

2.2 Influence Quantification

Post-Level Influence Quantification

On social networks, when we need to evaluate the marketing or the spread effect of a post, it is necessary to quantify the influence of the post. The quantification of the post influence depends upon how the post is spread on social networks, such as the history of share, like, comment and other similar action to reshare the post, to introduce sorts of influence quantification indicators. Over the related work, Cha *et al.* [15] were the first to work on influence quantification on Twitter. Cha *et al.* considered the rate of post spread and defined levels of post influence by the number of reshares. The major finding of their work is that hot posts are not necessary to have a positive link with rate influence as well as spread influence. Tang *et al.* [16] introduced influence quantification indicators for diverse post topics and proposed Topical Affinity Propagation to demonstrate how varied post topics impact the influence quantification indicators on social networks. Ho *et al.* [17] procured the tree structure of information propagation process and also combined some factors like the rate of reshare, the number of reshares and the geographical distance of influence propagation to define the influence of posts. In the latest study, Subbian *et al.* [18] defined and extracted user's features under the social network structure which considered the initial timeframe of the posts' influence flow. Employing the methods in survival analysis, Subbian *et al.* created a classifier and tried to spot the influential posts at the beginning of the information propagation. Li *et al.* [19] predicted the influence popularity of posts by considering how they propagate in the social network. The idea of concept drift is exploited to better capture the temporal dynamics in influence propagation.

User-Level Influence Quantification

Compared with post-level influence quantification, user-level influence quantification draws more attention to the researchers. Agarwal *et al.* [20] formulated the problem of blogger influence quantification. They computed the blogger influence indicators from multiple dimensions like social network, citation of posts, user activity and semantic analysis to find different types of influential users. Goyal *et al.* [21] constructed several information propagation graphs from the information sharing history and quantified influential leaders capable of exerting leverage on a number of users within a time period through mining frequent-pattern subgraphs. Topical Affinity Propagation proposed by Tang *et al.* [16] could simulate the influence flow of various posts' topics on social networks. After learning influence probability of dissimilar topics between users, they utilized PageRank to find influential users. Zhang *et al*. [22] deemed not only should we consider the extent of information propagation, but the time efficiency along with the area in geographical space of propagation it induced. Given a post, they gave users' geographical social influence a definition. Bernstein *et al*. [23] attempted to compute the number of users' invisible audience so as to quantify the user influence. By means of survey analysis, the users' invisible audience were often underestimated and the number of friends, like, and comment failed to respond the number of invisible audience as well. Li *et al*. [24] put forward Conductance Eigenvector Centrality, the indicator which simulated the conduction of electric current via random walk to quantify the user influence on social networks.

2.3 Influence Prediction

Post-Level Influence Prediction

The main difference between post-level influence prediction and quantification is that influence prediction forecasts future influence given the past posts which spread on social networks and the spreading information of the posts at the initial timeframe, influence quantification analyzes which factors or features will affect the influence of posts given all spreading information and the definition of various influence quantification indicators without forecasting. Cheng *et al*. [25] intended to predict would the retweet size double in the long run given a post, the spreading history and the number of influenced people k. They extracted the features of text, user, initial retweet, social network and time series and adopted logistic regression to be their predictive model. Gao *et al*. [26] similarly selected retweet size to be the influence of a post, yet they only applied time series of retweet time to train the predictive model modified from reinforced poisson process. Krishnan *et al*. [27] considered the prediction-related features from the tree structure and extracted them from the evolving structure of node-level and tree-level. They also used logistic regression to predict would the influence will be doubled. Li *et al*. [28] proposed to employ DeepCas model (End-to-End Deep Learning model) to predict the post influence. Merely input the cascade (the structure of resharing), then the DeepCas model could learn an embedding-vector feature to train the predictive model.

User-Level Influence Prediction

User-level influence prediction is the most relevant study to our work. The objective of user-level influence prediction is to predict who will bring the biggest number of post-resharings by looking into the post-resharing history of users. The primary distinction between post-level and user-level influence prediction is that user-level influence prediction would not predict the influence from the users' initial resharing history of new posts. In comparison with influence quantification which uses the resharing history of posts to quantify the influence and influence maximization which is merely the influence-spreading simulation, user-level influence prediction is a more direct study to evaluate the future influence of users. This kind of study, however, remains minor in the research field. Ghosh *et al*. [29] were the first to predict the user-level influence. They adopted centrality indicators in the social network structure, *e.g*. Closeness, Betweenness, PageRank and Katz, to measure influence and also defined the rating of users to be the true influence. By sorting, they analyzed which quantification of centrality is closest to the true influence. Bakshy *et al*. [2] predicted the user influence on social media sites. They defined user influence to be the average final number of resharings of users and captured features of user attributes, *e.g.* number of followers, friends, posts, and features of past influence, *e.g.* the maximum and minimum number of resharings, to predict the influence by regression tree model. Cui *et al.* [30] created a recommender system to evaluate the user influence. They constructed an influence matrix where the element of the matrix is the post influence of the user (*e.g.* the number of post-sharings) and proposed hybrid factor non-negative matrix factorization to predict the future influence of users. Iwata *et al.* [31] proposed Shared Cascade Poisson processes to measure the influence between friends on social network. It only needed to input the history of post-resharing; still, the model failed to forecast the user influence and did not evaluate the accuracy along with standard error in the experiment.

Nonetheless, we consider there are some limitations to the previous studies. (a) Ghosh *et al.* [29] solely considered the structure of social network, but the user influence is mostly evaluated by the user-post interaction; (b) Bakshy *et al.* [2] required some private information, *e.g.* gender, age, the number of followers, and the content of posts; (c) Cui *et al.* [30] defined the influence only from friends' response and sharing; (d) Iwata *et al.* [31] only predicted the future influence between users and did not define the user influence for prediction.

2.4 Post-Time Schedule

The goal of post-time scheduling is to find the best timeframe in order to earn the most replies for a user. Spasojevic *et al.* [3] formulated two methods for post-time schedule. The first one is accumulating the number of responses to the target user in a specified timeframe and then computing the ratio to the value of other timeframes (*i.e.* the probability of response acquiring in a specified timeframe). The timeframe with the biggest probability is the best one to post for the user. The second one is to compute the timeframe with the most replies directly then this timeframe would be the best one to post. Although Spasojevic *et al.* brought out the methods for post-time scheduling, there were two limitations to them; (a) It solely applied to a given user and failed to consider for multiple users; (b) The best timeframe for the present posts did not imply it would be the best one for future posts. Karimi *et al.* [32] and Zarezade *et al.* [33] presented a stochastic optimal control algorithm, notwithstanding, the limitations were failed to be considered.

3. PROBLEM FORMULATION

3.1 Notation Definition

We first enumerate the notations of this study in the following Table 1.

Definition 1 (History of Creating Post C). *Define the history that user u_i created a post p_{ik} at post time t_p to be $C = (u_i, p_{ik}, t_p)$.*

Definition 2 (History of Reply R). *Define the history that actor a_j replied user u_i the post p_{ik} at reply time t_r to be $R = (u_i, p_{ik}, a_j, t_r)$.*

Table 1. Notation.

symbol	interpretation
u_i	user i, where $u_i \in U$
a_j	actor j, where $a_j \in A$
p_{ik}	the k^{th} post of user i, where $p_{ik} \in P$
r_{ikm}	the m^{th} reply in the k^{th} post of user i
t_p	post time
t_r	reply time
D	future timeframe
T	the set of all timeframes
n_{24}	the number of posts within last 24 hours
n_{train}	the number of posts during training time
(u,v) or e_{uv}	the edge between u and v, where $u,v \in U$ or A

Definition 3 (History of Interaction I)**.** *Define the history that user u_i interacted with post p_{ik} and actor a_j to be $I = (C,R)$.*

3.2 Predict the Most Future-Influential Curators

We define that if a user earns the big number of replies, the user has potential to be the influential curator. In this study, we aim to predict the estimated number of replies earned by users U in the next 24 hours, which refers to the future influence in Definition 4. With the purpose to mine the most future-influential curators, we consider two possible application scenarios and come up with two corresponding predictive methods, which will be shown in Sections 3.2.1 and 3.2.2.

Definition 4 (future influence σ_D)**.** *Given a user u, current time t_c and future timeframe D, we define future influence*

$$\sigma_D(u_i) = \sum_{t_p=t_c}^{t_c+D} |I_{t_p}(u_i)|, \;\; u_i \in U,$$

where I_{tp} is the history of interaction of a user at time t_p.

3.2.1 Task 1: Future influence ranking prediction

To find the most future-influential curators, we predict the estimated number of replies of users in the next 24 hours. As companies may update their marketing time period, the cost for online marketing can be affected and accordingly, the number of curators (the budget k) is subject to be dynamic. With unfixed number of curators, we take it as Future Influence Ranking Prediction (FIRP) problem, and the problem statement is shown below.

Task 1	**Problem Statement**
Problem	Future Influence Ranking Prediction (FIRP)
input	the history of interaction I, current time t_c and budget k
goal	For posts created in next 24 hours ($t_c + 24$), we predict and generate a ranked list L_s to all users $u \in U$ so that the most future-influential users are listed top-k. *i.e.*, $\sigma_D(L_s(i)) > \sigma_D(L_s(i+1))$, where $L_s(i)$ is the rank-i user in the ranked list L_s, and $\|L_s\| = k$.

where the history of interaction I is the training data for the predictive model.

3.2.2 Task 2: Future influential leader prediction

Compared with the application scenario in FIRP, some companies prefer to have the assurance of post influence. They request a threshold b of the number of replies for their profit assurance. In this study, we consider it as Future Influential Leader Prediction problem (FILP), a problem of classification with 1 as future influential leader and 0 as otherwise (see Eq. (1)).

$$y_u = \begin{cases} 1, & \text{if } \sigma_D(u) \geq b \\ 0, & \text{if } \sigma_D(u) < b \end{cases}, \text{ where } u \in U \tag{1}$$

Thus, we have the problem statement shown below.

Task 2	**Problem Statement**
Problem	Future Influential Leader Prediction (FILP)
input	the history of interaction I, current time t_c and future influence threshold b
goal	For posts created in next 24 hours ($t_c + 24$), we predict and classify the future influence of all users $u \in U$ by future influence threshold b so that we can generate a classified list L_h with accuracy. *i.e.* $\forall v \in L_h,\ \hat{y}_v = \begin{cases} 1, & \text{if } \sigma_D(v) \geq b \\ 0, & \text{if } \sigma_D(v) < b \end{cases}$

where the history of interaction I is the training data for the predictive model.

3.3 Predict the Most Influential Post Timeframe

From the previous problems mentioned above, we can mine the most future influential curators on social networks. Still, if the companies fail to plan the best post timeframe for the curators, the curators will not be able to exert the biggest leverage on the audience and the cost will rise accordingly. Here, we formulate the problem that how to predict the most influential post timeframe to address the issue, and also schedule the post timeframe under two different application scenarios. We define the timeframe to be a 3-hour time period; that is, we have 8 timeframes within 24 hours (T) – 1-3 a.m. (T_1) to 10-12 p.m. (T_8), *i.e.* $T = \{T_1 = [1:00, 3:59] \cup T_2 = [4:00, 6:59] \cup \cdots \cup T_8 = [22:00, 24:59]\}$.

3.3.1 Task 3: Accumulated influence post-time scheduling

When a limited offer is about to start, the companies call for the biggest amount of accumulated replies received within the next 24 hours (*i.e.* accumulated influence, see Definition 5, which sums up the future influence received from the posts sent within T_j until the reply time t_r is later than the due time t_z). Under this scenario, we formulated accumulated influence post-time scheduling to offer the curators the best post-time for yielding influential posts. Apart from that, we also consider two possible conditions to post, unlimited time and limited time. Unlimited time means we do not restrict our curators posting at the same timeframe, they can post at their each best post-time where the received amount of replies is maximized for the companies to gain the maximum profit. By contrast, limited time suggests the curators are requested to post at the same timeframe. For instance, when there is a breaking news out, curators in the press post it at the very first moment to attract most of the audience to win the biggest readership.

Definition 5 (Accumulated Influence σ_{acc})**.** *Given the history of interaction I of user u_i, post timeframe T_j and due time t_z, we define accumulated influence*

$$\sigma_{acc}(u_i, T_j) = \sum_{t_p \in T_j} \sum_{t_r = t_p}^{t_z} |I(u_i, t_p, t_r)|,$$

where $I(u_i, t_p)$ is the history of interaction of a user u_i at post-time t_p.

Here, we have the problem statement shown below.

Task 3	**Problem Statement**
Problem	Accumulated Influence Post-time Scheduling
input	the history of interaction I, current time t_c, the ranked list L_s, the classified list L_h and the estimated number of curators (budget) k
goal	In the next 24 hours ($t_c + 24$), under the condition unlimited time/limited time, we predict the best post timeframe T^* for user $u \in \{L_s \text{ } or \text{ } L_h\}$ so that we can find the best k influential curators U_k and their influence accumulated to the due time t_z is greater than the other timeframes. *i.e.* $\sigma_{acc}(T^*) > \sigma_{acc}(T_i)$, $T_i \in T \setminus T^*$

where the history of interaction I is the training data for the predictive model.

3.3.2 Task 4: Limited influence post-time scheduling

Sometimes, companies feel like receiving the most replies at the time the posts are out, like live interaction. For example, broadcasters want to receive responses from the viewers during their live broadcasts. Hence, we formulate limited influence post-time scheduling problem, where the definition of limited influence is shown in Definition 6, which sums up the future influence received within T_j where the user u_i put out posts. As limited time posting is not reasonable under this setting, we merely consider unlimited time here.

Definition 6 (Limited Influence σ_{lim}). *Given the history of interaction I of user u_i and post timeframe T_j, we define limited influence*

$$\sigma_{lim}(u_i, T_j) = \sum_{t_p \in T_j} |I(u_i, t_p)|,$$

where $I(u_i, t_p)$ is the history of interaction of a user u_i at time t_p.

Therefore, we have the problem statement shown below.

Task 4	**Problem Statement**
Problem	Limited Influence Post-time Scheduling
input	the history of interaction I, current time t_c, the ranked list L_s, the classified list L_h and the estimated number of curators (budget) k
goal	In the next 24 hours ($t_c + 24$), under the condition unlimited time, we predict the best post timeframe T^* for user $u \in \{L_s \; or \; L_h\}$ so that we can find the best k influential curators U_k and their limited influence at the timeframe T^* is greater than the other timeframes. *i.e.* $\sigma_{lim}(T^*) > \sigma_{lim}(T_i)$, $T_i \in T \setminus T^*$

where the history of interaction I is the training data for the predictive model.

4. PROPOSED METHODS

4.1 Feature Engineering

To make the profile of future influential curators, we set out to extract the basic features from them, which can be acquired directly from data, *e.g.* the number of posts in the last 24 hours. Besides, we consider advanced features which can be indirectly obtained from the data, such as mean time interval between post-time in the last 24 hours. In addition, we learn that social network can be represented in graph, and inspired by the representation learning algorithm proposed by Grover *et al.* [34], we transform our network graph into learned features to capture the behavior we missed.

4.1.1 Basic feature

Basic features are descriptive statistics of users, which are shown in Table 2. We categorize these features into four classes for the better interpretation, where the difference between present and past is that the activity time is in the last 24 hours or in the history (training time).

4.1.2 Advanced feature

To complete the profile of future influential curators, we add advanced features to portray the users. Also, we categorize the features into four classes for the better interpretation, see Table 3.

Table 2. Basic features, F^b.

	present number of posts
f_{npp24}	estimated number of posts in the next 24 hours
f_{p24}	whether post in the last 24 hours, a binary value (0/1)
f_{np24}	the number of posts in the last 24 hours
	present number of replies
f_{nr24}	the number of replies received in the last 24 hours
	past number of posts
f_{npt}	the number of posts in the history
	past number of replies
f_{nrt}	the number of replies in the history
f_{nUA}	the number of unique actors a user has
f_{mrp}	average number of replies per post
f_{UAmrp}	average number of unique actors per post

Table 3. Advanced features, F^a (unit of time: sec).

	present speed of post
f_{p24m}	mean interval between post-time in the last 24 hours ($i.e.$ $\sum_{k=1}^{n_{24}-1} \frac{t_{p_{i(k+1)}} - t_{p_{ik}}}{n_{24}-1}$, $t_p \in ((t_c - 24), t_c])$
	present speed of reply
f_{r24m}	average time of delay of the first reply in the last 24 hours ($i.e.$ $\sum_{k=1}^{n_{24}} \frac{t_{r_{ik1}} - t_{p_{ik}}}{n_{24}}$, $t_r, t_p \in ((t_c - 24), t_c])$
	past number of posts
f_{npp_i}	the number of posts per timeframe (1-4 a.m., …, 21-24 p.m.), where $i = 1$ indicates 1-4 a.m., $i = 2$ implies 5-8 a.m., *etc*.
	past speed of reply
f_{rm}	average time of delay of the first reply ($i.e.$ $\sum_{i=1}^{n_{train}} \frac{t_{r_{ik1}} - t_{p_{ik}}}{n_{train}}$, $t_p, t_r \in (-\infty, t_c])$
f_{rmin}	the shortest time of delay of the first reply ($i.e.$ $\min(t_{r_{ik1}} - t_{p_{ik}})$, $t_p, t_r \in (-\infty, t_c])$
$f_{r_i m}$	the mean time period to accumulate i response(s), $i = 1, 10, 20, ..., 200$
$f_{UAr_i m}$	the mean time period to accumulate i unique actor(s), $i = 1, 10, 20, ..., 200$
f_{mr_i}	the average number of replies after posting in i hour(s), $i = 1, 3, 6, 12$

4.1.3 Learned feature

Basic and advanced features seem complete for user profiles, though, they fail to reflect the interaction between users and user-actor. Fortunately, Grover *et al*. [34] proposed node2vec algorithm to capture the node relationship based on the network graph, which we introduce *Similarity Graph* and *Bipartite Graph* to respond to the previous concern.

$$cos(f(u), f(v)) = \frac{f(u) \cdot f(v)}{\|f(u)\| \, \|f(v)\|}, \; u \neq v \tag{2}$$

Similarity graph (G_{sim}) captures the similarity of behavior between users. It implies the higher the similarity between users, the closer the number of replies the users receive. We define the behavior is the combination of basic and advanced features. The similarity of behavior is measured by cosine similarity (see Eq. 2), where $f(u)$ is the feature vector of user u. The edges of the similarity graph are created if the similarity of pair of users is ranked top-δ against each user u. *Bipartite graph* (G_{bip}) considers the relationship between users and actors. Somehow the post influence of users does not hinge upon users themselves; conversely, it is expanded by actors. If the actors with who the users interact are influential, as means the actors bring most of the responses from their audience, the users would then be influential indirectly. Therefore, we construct the graph on the basis of user-actor interaction, where the edges are created if the actors have replied to the users and the weights on them indicate the number of replies.

4.2 Task 1: Future Influence Ranking Prediction

To find the most future-influential curators, we can take it to be the sorting problem of future influence. In this study, we implement Learning to Rank algorithm to sort the order of future influence immediately, which has been proved to be an efficient way to the sorting problem. Specifically, we perform LambdaMART algorithm[35] with Gradient Boosting Machine (GBM) to be the proposed method of this problem.

4.3 Task 2: Future Influential Leader Prediction

In this problem, we set a threshold b to categorize the estimated future influence $\sigma_D(u)$ using random forest on the training data to predict the future influential leader. However, the distribution of the number of replies received by users is close to power-law distribution. To be concise, more users receive the lower number of replies. Accordingly, as the threshold b is set to be large, it would induce the imbalanced data problem that almost all the users would be classified to the same class. It is acknowledged that we can avoid this problem by setting b low, while, it is not the case businesses desire. To tackle this problem, we propose two algorithms for training the imbalanced data – *self-training algorithm* and *mutual-training algorithm.*

4.3.1 Self-training algorithm

The idea of self-training algorithm is to transform the imbalanced data into balanced one for higher classification accuracy. It is incremental-learning-like approach that it trains itself until the break conditions are met.

The self-training algorithm can be followed as below. For the preparation step, sample the imbalanced training data to be the balanced one. For example, sample the imbalanced data with the ratio of class-0 to class-1 instances being 66:1 to be 1:1. Then step 1 in the main procedure, model this balanced training data for classification prediction. Step 2, predict the probabilities of the testing data with the model. Step 3, set a probability threshold p_c for binary class prediction. If the classification outcome of testing data is greater than p_c, then this testing data will be taken into the balanced training data and be thrown out of testing data to enhance the detection ability of class 1. Step 4, encode the classification result this round then go back to Step 1. The procedure breaks when no testing data pass the threshold p_c or testing data become an empty set.

4.3.2 Mutual-training algorithm

To take advantage of both unlearned (*e.g.* basic and advanced features) and learned features, we propose another mutual-training algorithm (see Algorithm 1) to secure the strong points of these two dissimilar features. The algorithm is performed as below. For preparation step, we similarly sample the imbalanced data to the balanced one. Then Step 1 in the main procedure, model the balanced unlearned training data $T_r^{u\prime}$ for classification prediction. Step 2, predict the probabilities of the unlearned testing data with the unlearned model M^u. Step 3, set a probability threshold p_c for binary class prediction. If the classi-

Algorithm 1 : Mutual-training Algorithm

```
Input: $M^u$: unlearned model, $M^l$: learned model,
       $T_r^u$: training data in $M^u$, $T_r^l$: training data in $M^l$,
       $T_e^u = T_e^{u\prime}$: testing data in $M^u$,
       $T_e^l = T_e^{l\prime}$: testing data in $M^l$, $p_c$: probability threshold
       $T_e^{u\prime}(p)$: subset of $T_e^{u\prime}$ with the row including predicted probability $p$
       $T_e^{l\prime}(p)$: subset of $T_e^{l\prime}$ with the row including predicted probability $p$
Output: $\hat{Y}_h^{u\prime}, \hat{Y}_h^{l\prime}$: the classification results
 1: sample $T_r^u$, $T_r^l$ to generate the balanced data $T_r^{u\prime}$, $T_r^{l\prime}$ with the same number of samples
    in each class
 2: $i = 1$
 3: repeat
 4:     $M^u = Classifier(T_r^{u\prime})$ // train the unlearned model
 5:     $\hat{Y}_h^u = M_{prob}^u(T_e^{u\prime})$, $\hat{y}_h^u \in \hat{Y}_h^u$ // predict the probabilities with the unlearned model
 6:     if $\hat{y}_h^u > p_c$ then $T_e^{u\prime} = T_e^{u\prime} \backslash T_e^{u\prime}(\{\hat{y}_h^u\})$
 7:     if $T_e^{u\prime}(\{\hat{y}_h^u\}) = \emptyset$ then break // break condition 1
 8:     if $T_e^{u\prime} = \emptyset$ then break // break condition 2
 9:     convert $T_e^{u\prime}(\{\hat{y}_h^u\})$ into the form of $T_r^{l\prime}$ to yield $T_e^{u\prime\prime}$
10:     $T_r^{l\prime} = \{T_r^{l\prime} \cup T_e^{u\prime\prime}\}$
11:     $M^l = Classifier(T_r^{l\prime})$ // train the learned model
12:     $\hat{Y}_h^l = M_{prob}^l(T_e^{l\prime})$, $\hat{y}_h^l \in \hat{Y}_h^l$ // predict the probabilities with the learned model
13:     if $\hat{y}_h^l > p_c$ then $T_e^{l\prime} = T_e^{l\prime} \backslash T_e^{l\prime}(\{\hat{y}_h^l\})$
14:     convert $T_e^{l\prime}(\{\hat{y}_h^l\})$ into the form of $T_r^{u\prime}$ to yield $T_e^{l\prime\prime}$
15:     $T_r^{u\prime} = \{T_r^{u\prime} \cup T_e^{l\prime\prime}\}$
16:     // encode results per loop in the list //
17:     $\hat{Y}_h^{u\prime}[[i]] = M^u(T_e^u)$
18:     $\hat{Y}_h^{l\prime}[[i]] = M^l(T_e^l)$
19:     $i = i + 1$
20: until break condition 1 or 2
21: return $\hat{Y}_h^{u\prime}, \hat{Y}_h^{l\prime}$
```

fication outcome of unlearned testing data is greater than p_c, then this unlearned testing data will be thrown out of unlearned testing data. Step 4, convert those passed unlearned testing data into the form of learned features and add them to the balanced learned training data $T_r^{l\prime}$. Step 5, model the balanced learned training data for classification prediction. Step 6, predict the probabilities of the learned testing data with the learned model M^l. Step 7, If the classification outcome of learned testing data is greater than p_c, then this learned testing data will be thrown out of learned testing data. Step 8, convert those passed learned testing data into the form of unlearned features and add them to the balanced unlearned training data. Step 9, encode the classification result this round then go back to step 1. The procedure breaks when no testing data pass the threshold p_c or testing data become an empty set.

4.4 Tasks 3 & 4: Predict the Most Influential Post Timeframe

After obtaining the list of most future-influential curators, we propose two post-time scheduling methods, unsupervised *PastGreedy* algorithm and supervised *Forecasting* regression prediction considering the condition under unlimited time and limited time.

4.4.1 PastGreedy algorithm – unlimited

The aim of *PastGreedy* (*PG*) is to compute which timeframe would yield most of the influence in the past and then assume this timeframe will also be the best post-time for curators in the next 24 hours. Considering the condition of unlimited time, we find the influential candidates and have them post within their best post timeframe where it could be overlapped. This way, we can mine the most future-influential curators and their best post-time simultaneously. Nevertheless, it could cause the problem of common audience, which means the users might interact with the same group of actors. As a result, the influence fails to be propagated. In *PG*, we avoid this problem by deducting all the audience of the TOP1 (the most influential) user, and we find out TOP2 in the remaining user set. Repeatedly run the previous step until we score TOP-k curator and her best post-time, then the procedure is terminated.

In accumulated influence problem, we focus on the timeframe where users post can receive most of the influence accumulated until the due time in the history. Hence, we compute the best post-time based on users' post timeframe. On the other hand, in limited influence problem, we are only interested in the timeframe where actors reply. If we post within the frequent timeframe where actors reply, it is believed that the users gain more limited influence. Thus, we take this reply-time to be the users' best post timeframe.

4.4.2 PastGreedy algorithm – limited

Considering limited-time posting, *PG* only leaves those whose influence is above the threshold to be the candidates of future-influential curators. With this subset of users, *PG* clusters it to 8 groups based on their each post-time and mines top-k curators through using common audience deduction within each cluster. After summing up the influence of those k curators in each group, we can derive the final answer by selecting the group with the highest influence.

4.4.3 Forecasting regression prediction – unlimited

We can discuss the unlimited-time posting problem from the supervised prediction angle. We use regression to predict the future influence per timeframe (Y_i) with user behavior features (X). As there are 8 timeframes to predict, we perform the regression 8 times to see which one possessing the highest estimated future influence. Similar to the task 1, we employ GBM under Gaussian distribution to predict.

In terms of accumulated influence problem, response variable Y_i in training data signifies the influence yielded from the posts posted in the i^{th} post-time. In the limited influence problem, on the other hand, response variable Y_i in training data suggests the influence generated in the i^{th} reply-time. To schedule the best post-time and find out k curators, under the condition of unlimited-time posting, we pick up the highest k influence values with their corresponding curators and post-time to be the final answer. If there are some users selected more than once, we only consider the timeframe with higher influence per user and keep picking up the highest influence in the remaining prediction result until k distinct curators are chosen.

4.4.4 Forecasting regression prediction – limited

Under limited-time posting condition, after the regression prediction of future influence, *Forecasting* (*FC*) sums up the estimated influence per timeframe, and the best post-time will be the timeframe with the biggest sum. Plus, we can derive the most future-influential k curators in this time period.

5. EXPERIMENTAL EVALUATION

5.1 Dataset

The dataset is downloaded from `https://github.com/klout/opendata` which is the one that Spasojevic *et al*. [3] ran the experiments. This Facebook data record the history of users' and actors' activity and also include post id, user id, actor id, post timestamp, action timestamp, and user timezone. In this study, we merely experiment on the data which record the activity history from 2014/10/14 to 2014/12/15 due to the time complexity. This experimental data involve 0.58 million users, 9.48 million actors, 7.8 million posts and 30 million replies.

5.2 Experiment Setting

We train and test the data on 5 groups for experiment stability. Targeting a date to predict the future influence, we use previous 10-day data to do the task. A group is formed by training 10-day data to predict the future influence in the next 24 hours, then we move forward the training period by 1 day and test it to predict the future influence in another next 24 hours.

For example, in the training data, we use 10-day data 11/4 – 13 to build a model which predicts the future influence on 11/14; in the testing data, we move forward training period by 1 day to 11/5 – 14 for the prediction of the future influence on 11/15. This entire model forecasting task forms 1 group.

Owing to time complexity in the experiment, we randomly choose 2000 users for our following analysis.

In terms of the features in the model, we use different feature combination on the model to see which combination is the best to predict the future influence. *all* feature means the combination of basic, advanced features in addition to the learned feature of similarity/bipartite graph. As we have two graphs constructed for the learned feature, we will compare the prediction results of these two graphs in the following tasks and also combine these two elements with all the other features to be a new feature combination – *allPlus* to see if the performance will be enhanced.

5.3 Task 1: FIRP

We select *precision* to measure the performance, which is defined to be the ratio of the intersection of predicted top N users and real top N users to N users (see Eq. 3).

$$precision@N = \frac{\#\{pred(top\,N) \cap real(top\,N)\}}{N} \tag{3}$$

We can learn from Fig. 1 that *allPlus* and *all* have better prediction performance as N variates and which shows the precision increases if we add learned features. As to the comparison of graphs, we can barely tell the difference between them.

5.4 Task 2: FILP

Different probability threshold p_c could lead to several combinations of recall and precision. In order to measure the performance of these two metrics, we adopt $F1$ measure (see Eq. 4) for evaluation, where P is precision and R is recall.

$$F_1 = 2 \times \frac{P \times R}{P + R} \tag{4}$$

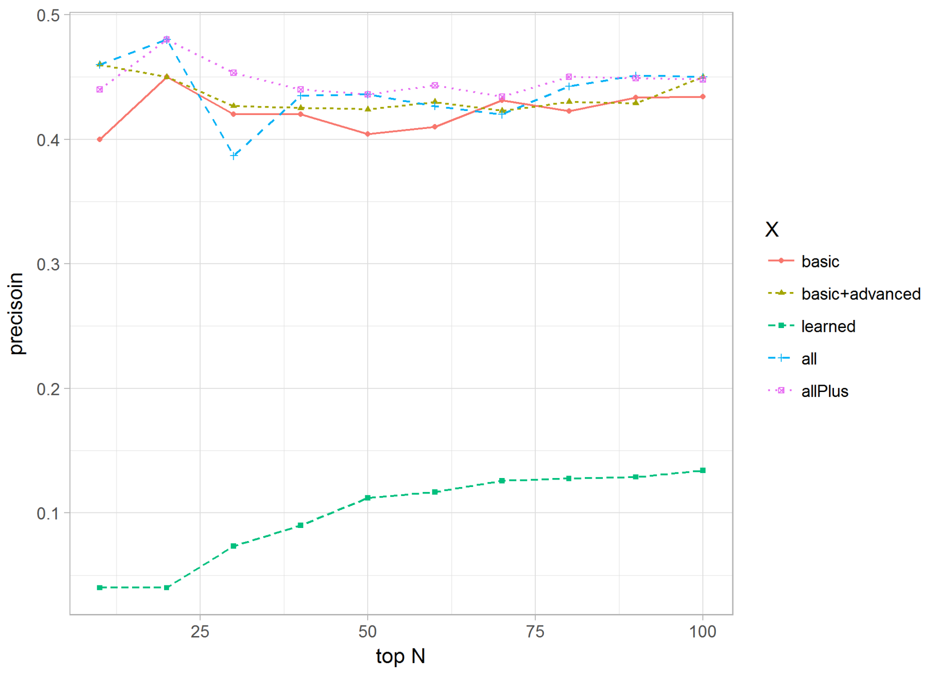
From Fig. 2 we can find that mutual-learning model outperforms other methods. When $p_c = 0.5$, mutual-learning model can achieve up to 0.875 in $F1$ score, which improves *basic* by more than 150%. As p_c increases, probability threshold becomes more rigid and fewer samples are fed into the model for training. The future influential leader recognition ability thus is weaker than that when p_c is lower.

5.5 Task 3& 4: Predict the Most Influential Post Timeframe

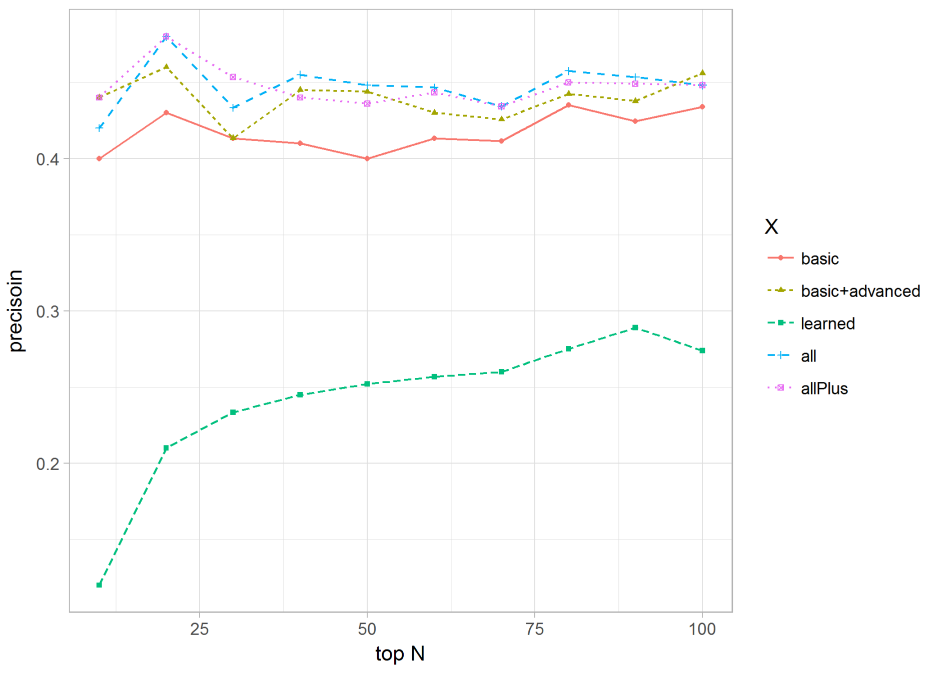
5.5.1 Evaluation metrics

RG_k for Limited Influence Post-time Scheduling

Derived from *ReactionGain* (*RG*) proposed by Spasojevic *et al*. [3], we modify its formula to support for k users in that it originally applies to only one user. The meaning of RG_k (see Eq. 6) is the same as RG, which indicates the ratio of user's influence per post in the best post-time to user's influence per post in overall time. If $RG > 1$, it implies the predicted best post-time T^* can effectively augment the users' limited influence.

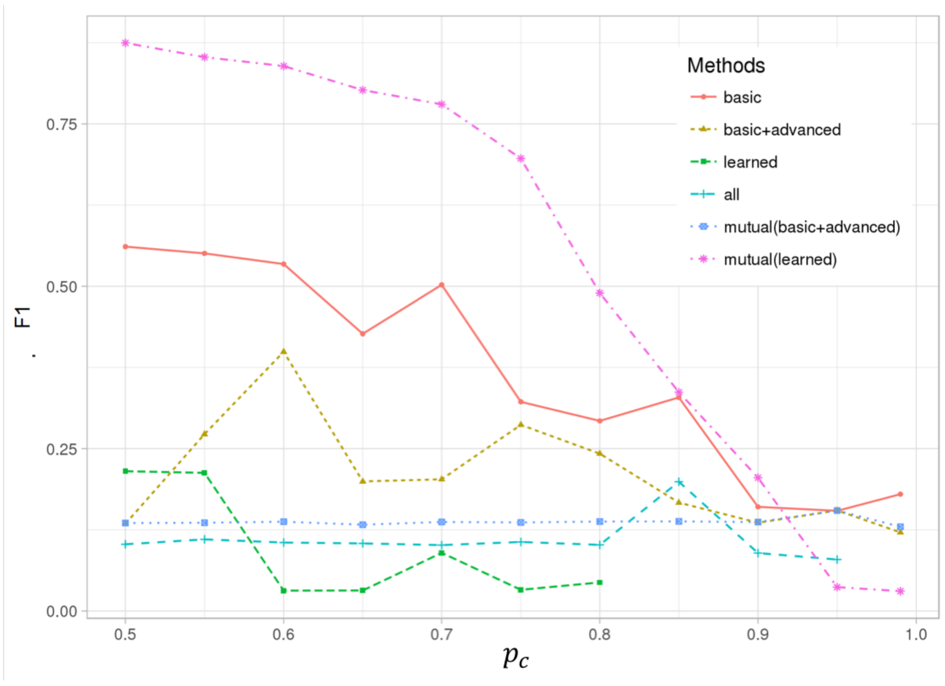


(a) Evaluation of features combination in G_{sim}.

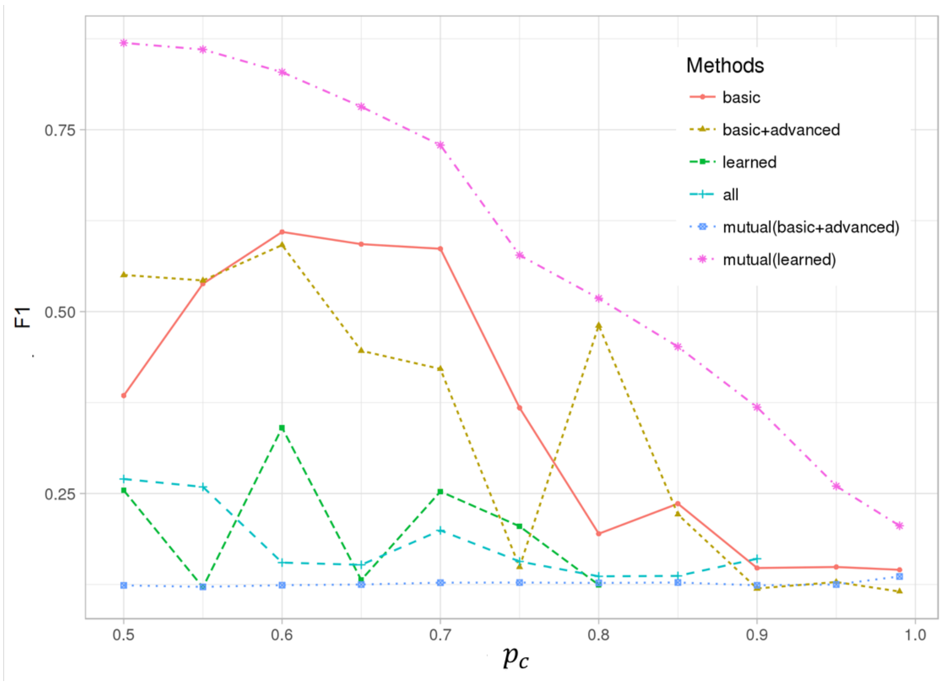


(b) Evaluation of features combination in G_{bip}.

Fig. 1. FIRP evaluation ($\delta = 15$).



(a) Evaluation of features combination in G_{sim}.



(b) Evaluation of features combination in G_{bip}.

Fig. 2. FILP evaluation ($p_c = 0.5$, $b = 50$).

$$RPM(U_k, T^*) = \frac{\sum_{j=1}^{M} r_{T^*,j}(U_k)}{\sum_{j=1}^{M} p_{T^*,j}(U_k)} \tag{5}$$

$$RG_k = \frac{RPM(U_k, T^*)}{RPM(U_k)} \tag{6}$$

where $r_{T^*,j}(U_k)$ is the number of replies top-k users U_k received in T^* on the j-th day; $p_{T^*,j}(U_k)$ is the number of posts U_k posted in T^* on the j-th day; M in RPM (see Eq. 5) is the number of timeframes and U_k is k curators we mined.

RG_k^* for Accumulated Influence Post-time Scheduling

Considering *ReactionGain* can only apply to limited influence post-time scenario, we re-devise the evaluation metrics RG_k^* (see Eq. 8) for accumulated influence post-time problem. The difference between RG_k^* and RG_k is that we compute not only the limited influence at the best post-time but the accumulated influence from the posts posted at the best post-time, but the meanings are the same.

$$RPM^*(U_k, T^*) = \frac{\sum_{j=1}^{M} r_j(U_k, p_{T^*})}{\sum_{j=1}^{M} p_{T^*,j}(U_k)} \tag{7}$$

$$RG_k^* = \frac{RPM^*(U_k, T^*)}{RPM^*(U_k)} \tag{8}$$

where $r_j(U_k, p_{T^*})$ is the number of replies top-k users U_k received from the posts they posted in T^*.

As for the baselines in the experiments, we also re-devise *Most Frequently Used (MFU)* that Spasojevic *et al*. proposed (see Eq. 9). It shows the timeframe where curators post the most frequently is the best post-time.

$$MFU = \arg\max_i \left(\frac{p(U_k, t_p = T_i)}{\sum_i p(U_k, t_p = T_i)} \right) \tag{9}$$

where $p(U_k, t_p = T_i)$ is the number of posts top-k users U_k posted in T_i.

5.5.2 Experiment results

We present the following experiment results in the order of evaluation metrics.

Limited Influence Post-time Scheduling

Here, we solely discuss unlimited-time posting in this problem because it does not seem to make sense for limited-time posting.

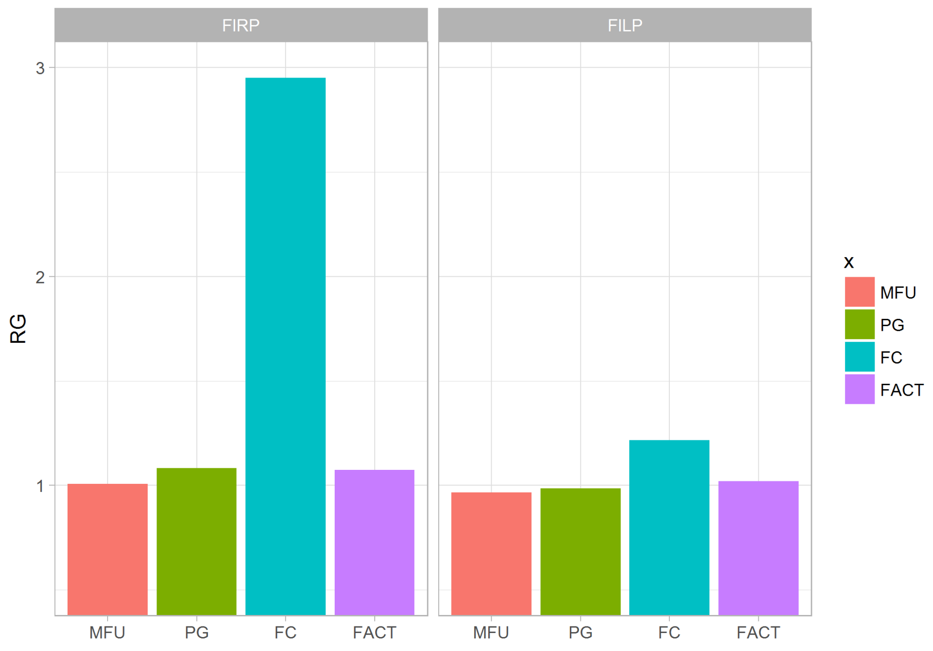


Fig. 3. Experiment result of limited influence scheduling.

It is clear in Fig. 3 that under both FIRP and FILP, the performance of FC surpasses baseline and PG, even the true answer, FACT. Here, true answer is calculated through U_k posting in their observed best post-time where they received the highest future influence. In FIRP, the RG_k almost achieves 3 by using FC, which shows when top k curators post at the predicted best post-time, their limited influence will be improved by about 300%. If we compare it with FACT, in FIRP, RG_k generated by FC is approximately 2.7 times the one of FACT; in FILP, it is around 1.15 times the one of FACT.

When we compare the result of FIRP with FILP, it is evident that the performance of FIRP is better than that of FILP. It is reasonable inasmuch as the default of b is 50, only a few number of users are classified as influential leaders, which means we can only pick up top k users from them. Therefore, it fails to optimize the RG_k.

Accumulated Influence Post-time Scheduling

Fig. 4 is the result under the condition of unlimited-time posting. We can also see that *FC* outperforms all the other methods and FACT, and RG_k^* in FIRP is higher than in FILP. In both FIRP and FILP, RG_k^* of FC is about 1.1 times the one of FACT, which implies we improve the curators' accumulated influence effectively via FC method.

Fig. 5 is the result under the condition of limited-time posting. In FIRP, FC works better than the other methods, and the RG_k^* of FC is 1.36 times of that of FACT. In FILP, by contrast, unsupervised PG performs better than FC, and the RG_k^* of PG is around 2.76 times of that of FACT. It is presumed that because few future-influential leader candidates pass the threshold b, it induces high uncertainty when they are thrown into the GBM regression model. To derive the best post-time and U_k, summing up the predicted future influence in each T exacerbates the uncertainty of the result.

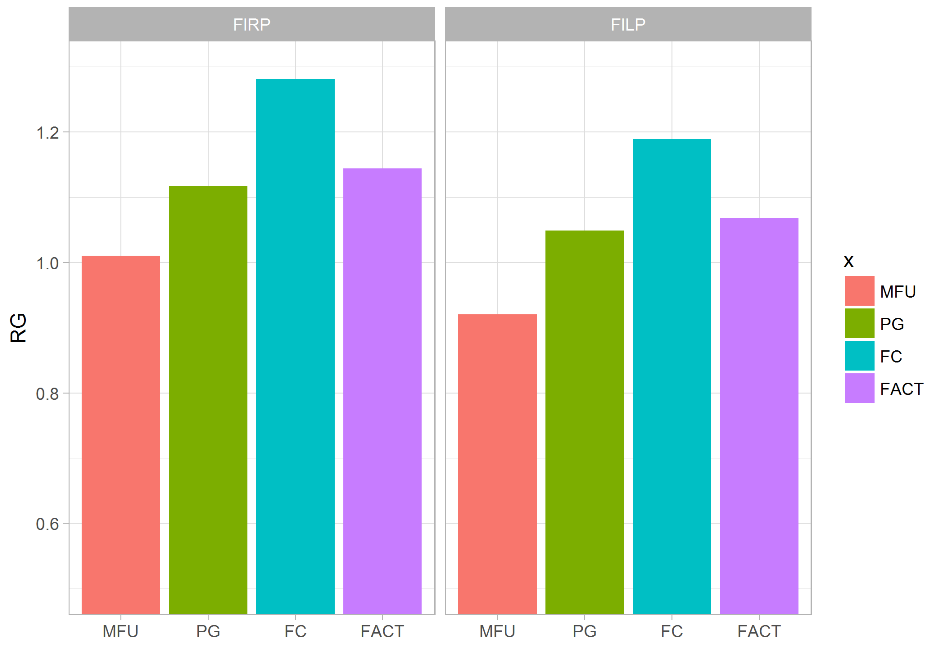


Fig. 4. Experiment result of accumulated influence scheduling (unlimited-time condition).

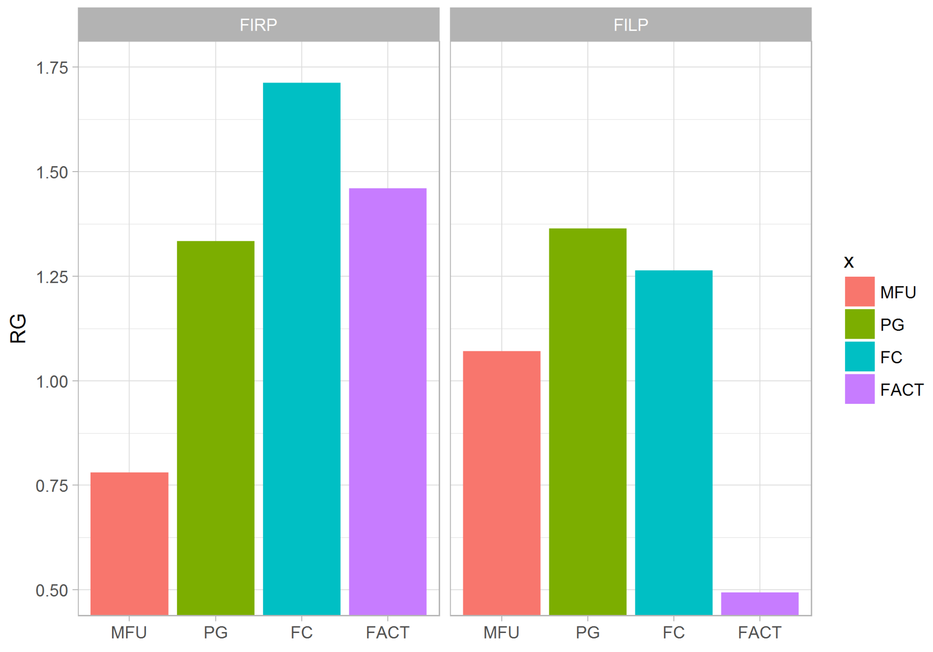


Fig. 5. Experiment result of accumulated influence scheduling (limited-time condition).

6. CONCLUSION

So far, this is the first study that explores the problem of mining future-influential curators and scheduling the best post-time for them at the same time. To predict the most future-influential curators, we discuss Future Influence Ranking Prediction (FIRP) and Future Influential Leader Prediction (FILP) to answer companies' need, and also define the features to profile the future-influential curators. With descriptive basic features, advanced features and learned features learned from the similarity graph and bipartite graph as independent variables, we find that learned features definitely augment the prediction ability in FIRP and FILP. Due to the problem of imbalanced data in FILP, we propose self-training algorithm and mutual-training algorithm to avoid it. From the experiment results, it is apparent that mutual-training algorithm is an efficacious solution and enhances the accuracy of prediction.

Finally, we formulate the problems of Accumulated Influence Post-time Scheduling and Limited Influence Post-time Scheduling. We come up with PastGreedy algorithm (PC) and Forecasting regression prediction (FC) to find out k future-influential curators and their corresponding best post-time simultaneously under either unlimited-time or limited-time posting condition. The experiment results show that our post-time schedule for the next 24 hours can improve the curators' future influence notably.

ACKNOWLEDGMENT

This work is supported by Ministry of Science and Technology (MOST) of Taiwan under grants 109-2636-E-006-017 and 108-2218-E-006-036, and also by Academia Sinica under grant AS-TP-107-M05.

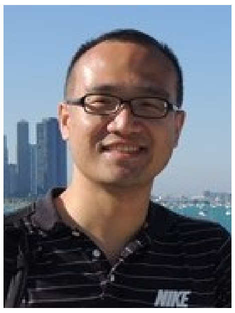
REFERENCES

1. D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003, pp. 137-146.
2. E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: Quantifying influence on twitter," in *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, 2011, pp. 65-74.
3. N. Spasojevic, Z. Li, A. Rao, and P. Bhattacharyya, "When-to-post on social networks," in *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 2127-2136.
4. W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 199-208.
5. Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2015, pp. 1539-1554.

6. Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2014, pp. 75-86.
7. A. Goyal, F. Bonchi, and L. V. S. Lakshmanan, "A data-based approach to social influence maximization," *Proceedings of VLDB Endowment*, Vol. 5, 2011, pp. 73-84.
8. S. Bhagat, A. Goyal, and L. V. Lakshmanan, "Maximizing product adoption in social networks," in *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, 2012, pp. 603-612.
9. J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007, pp. 420-429.
10. T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila, "Finding effectors in social networks," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010, pp. 1059-1068.
11. C.-T. Li, S.-D. Lin, and M.-K. Shan, "Finding influential mediators in social networks," in *Proceedings of the 20th International Conference Companion on World Wide Web*, 2011, pp. 75-76.
12. C.-T. Li, Y.-J. Lin, and M.-Y. Yeh, "The roles of network communities in social information diffusion," in *Proceedings of IEEE International Conference on Big Data*, 2015, pp. 391-400.
13. C.-T. Li, Y.-J. Lin, and M.-Y. Yeh, "Forecasting participants of information diffusion on social networks with its applications," *Information Sciences*, Vol. 422, 2018, pp. 432-446.
14. C.-T. Li, M.-Y. Huang, and R. Yan, "Team formation with influence maximization for influential event organization on social networks," *World Wide Web*, Vol. 21, 2018, pp. 939-959.
15. M. Cha, A. Mislove, and K. P. Gummadi, "A measurement-driven analysis of information propagation in the flickr social network," in *Proceedings of the 18th International Conference on World Wide Web*, 2009, pp. 721-730.
16. J. Tang, J. Sun, C. Wang, and Z. Yang, "Social influence analysis in large-scale networks," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 807-816.
17. C. T. Ho, C. T. Li, and S. D. Lin, "Modeling and visualizing information propagation in a micro-blogging platform," in *Proceedings of International Conference on Advances in Social Networks Analysis and Mining*, 2011, pp. 328-335.
18. K. Subbian, B. A. Prakash, and L. Adamic, "Detecting large reshare cascades in social networks," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 597-605.
19. C.-T. Li, M.-K. Shan, S.-H. Jheng, and K.-C. Chou, "Exploiting concept drift to predict popularity of social multimedia in microblogs," *Information Sciences*, Vol. 339, 2016, pp. 310-331.
20. N. Agarwal, H. Liu, L. Tang, and P. S. Yu, "Identifying the influential bloggers in a community," in *Proceedings of International Conference on Web Search and Data Mining*, 2008, pp. 207-218.

21. A. Goyal, F. Bonchi, and L. V. Lakshmanan, "Discovering leaders from community actions," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, 2008, pp. 499-508.
22. C. Zhang, L. Shou, K. Chen, G. Chen, and Y. Bei, "Evaluating geo-social influence in location-based social networks," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 2012, pp. 1442-1451.
23. M. S. Bernstein, E. Bakshy, M. Burke, and B. Karrer, "Quantifying the invisible audience in social networks," in *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 21-30.
24. X. Li, Y. Liu, Y. Jiang, and X. Liu, "Identifying social influence in complex networks," *Neurocomputing*, Vol. 210, 2016, pp. 141-154.
25. J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec, "Can cascades be predicted?" in *Proceedings of the 23rd International Conference on World Wide Web*, 2014, pp. 925-936.
26. S. Gao, J. Ma, and Z. Chen, "Modeling and predicting retweeting dynamics on microblogging platforms," in *Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, 2015, pp. 107-116.
27. S. Krishnan, P. Butler, R. Tandon, J. Leskovec, and N. Ramakrishnan, "Seeing the forest for the trees: New approaches to forecasting cascades," in *Proceedings of the 8th ACM Conference on Web Science*, 2016, pp. 249-258.
28. C. Li, J. Ma, X. Guo, and Q. Mei, "Deepcas: An end-to-end predictor of information cascades," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 577-586.
29. R. Ghosh and K. Lerman, "Predicting influential users in online social networks," *CoRR*, Vol. abs/1005.4882, 2010.
30. P. Cui, F. Wang, S. Liu, M. Ou, S. Yang, and L. Sun, "Who should share what?: Item-level social influence prediction for users and posts ranking," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2011, pp. 185-194.
31. T. Iwata, A. Shah, and Z. Ghahramani, "Discovering latent influence in online social activities via shared cascade poisson processes," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 266-274.
32. M. R. Karimi, E. Tavakoli, M. Farajtabar, L. Song, and M. Gomez Rodriguez, "Smart broadcasting: Do you want to be seen?" in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1635-1644.
33. A. Zarezade, U. Upadhyay, H. R. Rabiee, and M. Gomez-Rodriguez, "Redqueen: An online algorithm for smart broadcasting in social networks," in *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, 2017, pp. 51-60.
34. A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.
35. C. J. Burges, "From RankNet to LambdaRank to LambdaMART: An overview," Microsoft Research Technical Report MSR-TR-2010-82, 2010.

Chieh-Cheng Hsia received the B.B.A. and the M.B.A. degrees in Statistics respectively from National Taipei University, Taipei, Taiwan and National Cheng Kung University, Tainan, Taiwan, respectively. She is currently working as a Research Assistant in Department of Statistics, National Cheng Kung University. Her research interests include social network analysis, network embedding learning, and machine learning.



Cheng-Te Li is now an Associate Professor at Institute of Data Science and Department of Statistics, National Cheng Kung University (NCKU), Tainan, Taiwan. Dr. Li received his Ph.D. degree (2013) from Graduate Institute of Networking and Multimedia, National Taiwan University. Before joining NCKU, he was an Assistant Research Fellow (2014-2016) at CITI, Academia Sinica. Dr. Li's research interests target at machine learning, deep learning, data mining, social networks and social media analysis, recommender systems, and natural language processing.