Incorporating User Grouping into Retweeting Behavior Modeling

Jinhan Zhu* Shuai Ma* Jing Wang† Hui Zhang* Chunming Hu* Xiong Li‡

Abstract

Social media applications are emerging, with rapidly growing users and large numbers of retweeted microblogs every minute. The variety among users makes it difficult to model their retweeting activities. Obviously, it is not suitable to cover the overall users by a single model. Meanwhile, building one model per user is not practical. To this end, this paper presents a novel solution, of which the principle is to model the retweeting behavior over user groups. Our approach, GruBa, consists of three key components for extracting user based features, clustering users into groups, and modeling upon each group. Particularly, we look into the user interest from different perspectives including long-term/recent interests and explicit/implicit interests, which results deep analyses towards the retweeting behavior and proper models in the end. We have evaluated the performance of GruBa by datasets of real-world social networking applications and a number of query workloads, showcasing its benefits.

1 Introduction

Social media is overwhelming nowadays, with massive users on Facebook, Twitter and Weibo while the number of users keeps increasing. These users behave variously, knowledge of which is significant in recommendation system, activity prediction and Big Thing analysis. Hence emerges the demand of developing systems and algorithms that could properly model user behaviors, which has already attracted the attention from both academia and industry.

Central to user behavior modeling, is the need to choose the granularity of model (i.e., how many users share one model), as well as the variety of features to be selected for differentiating users. Already, there exist work of building a single model for all the users [8, 31, 32]. Apparently, such model bears the limitation of being coarse. On the other hand, modeling each user is not practical, due to the tremendous number of users.

The key driver of our work is the realization that in social media applications, users could fall into groups and each group shares representative behaviors. As one example, consider the film *Brave Heart*, fans of which are probably addicted to highland, bagpipe and war films, and thus likely to retweet blogs of these topics. Particularly, we study the retweeting behavior of users and our work can be generalized to other behaviors of like and comment as well. In the realm of social network behavior modeling, few work has been done over grouping, which however has been proved to be effective in other fields such as economic research. This motivates us to explore the incorporation of user grouping into the retweeting behavior modeling, filling the gap of existed studies in literature.

Contributions. This work contributes as follows:

(1) We present a system named GruBa with the novel perspective to model user behaviors over groups instead of the mono model as in existed research.

(2) We leverage user interests to facilitate the modeling of retweeting behavior and look into interests with various dimensions, including long-term/recent interests and explicit/implicit interests.

(3) We evaluate the performance of GruBa using real-world datasets, showcasing its benefits against competitive state of the art approaches.

Organization. The rest of this paper is organized as follows. Section 2 first gives the problem formulation and subsequently overviews GruBa's components, principle of which are detailed in Sections 3, 4 and 5 separately. Section 6 provides the performance evaluation. Related work is presented in Section 7. Finally, Section 8 concludes the work.

2 GruBa Overview

2.1 Problem Formulation We consider people's retweeting behavior in social media. For simplicity, assuming the microblogs that a user can retweet come from those owned by his/her followees.

DEFINITION 1. *A microblog* $M_b = (O, T, M, flag)$ *consists of the owner* O *(a.k.a. user in this paper) to whom*

*SKLSDE Lab, Beihang University, China. Email: {zhujh@act., mashuai, zhanghui@act., hucm@}buaa.edu.cn

†Beijing Advanced Innovation Center for Big Data and Brain Computing (BDBC), Beihang University. Email: wjseawind@gmail.com

‡National Computer Network Emergency Response Technical Team/Coordination Center of China. Email:li.xiong@foxmail.com

M_b belongs (either created or retweeted), the timestamp T showing when M_b is generated, the microblog message M and a bit flag denoting M_b is retweeted (1) or originally created (0) by the owner O.

DEFINITION 2. *A user $u = (B_u, R_u, E_u)$ consists of three sets regarding the user's microblogs B_u, followers R_u and followees E_u separately. Each follower/followee per se refers to a user.*

By Definition *1* and *2*, $u = M_b.O$ and $M_b \in B_u$. Informally, providing a set of users $\mathbb{U}$ and the associated microblogs $\mathbb{B}$, as well as a microblog b and a follower of $b.O$ written as f, i.e., $f \in R_{b.O}$, GruBa shall build a retweeting model for $\mathbb{U}$ and $\mathbb{B}$, upon which 1/0 is returned regarding whether f shall retweet b.

2.2 GruBa Framework GruBa is designed from the ground up as a system for modeling users' retweeting behavior in social media. Figure 1 shows the architectural components of GruBa, mainly comprising Sina Microblog Data, Key Modules and Profile Demonstrator.

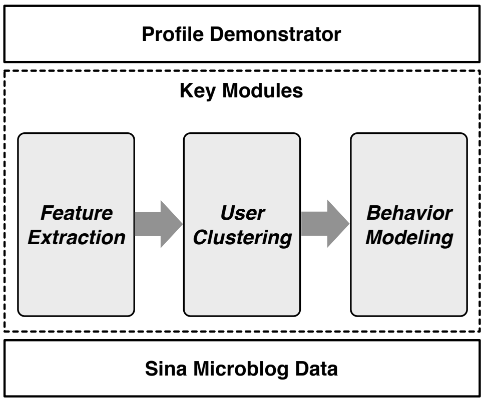


Figure 1: GruBa Architecture

Sina Microblog Data. It is the data to be processed by GruBa, i.e., data of microblogs and users, as shown in *Definition 1* and *2*.

Key Modules. In the heart of GruBa lies Key Modules, which consists of three major components.

(1) Feature Extraction: By coalescing the microblog data, each user is depicted by a bunch of features, which are grouped into three categories. They are features of *Basics* (e.g., the number of followers and followees), *Behavior* (e.g., the frequency and the popular slots of retweeting) and *Interest* (e.g., the long-term/recent interests, as well as the explicit/implicit interests). These features are extracted from the stored data by Feature Extraction module and serve as the input of User Clustering module.

(2) User Clustering: Providing the user-based features, User Clustering takes charge of the clustering task such that each user falls into a proper group.

(3) Behavior Modeling: For each group obtained by User Clustering, Behavior Modeling employs both positive and negative samples (i.e., microblogs that were labeled with retweeted and not retweeted) to train a model, over which the testing of users' retweeting behavior is performed.

Profile Demonstrator. At the top layer of GruBa, it is the Profile Demonstrator for visualization.

3 Feature Extraction

With the underlying Sina Microblog Data, Feature Extraction is responsible for "mining" the user characteristics, resulting three features for each user. These features are *Basic Feature*, *Behavior Feature* and *Interest Feature*, which constitute the *Feature Data* in GruBa, i.e., *Feature Data* = {*Basic Feature*, *Behavior Feature*, *Interest Feature*}.

3.1 Basic Feature *Basic Feature* employs a vector I to cover the basic info of user.

$$I = (\#R_u,\ \#E_u,\ R_{ee},\ \ U_t) \tag{3.1}$$

where each variable is illustrated in Table 1.

Table 1: Illustration of Variables in Basic Feature

variable	illustration
$\#R_u$	number of followers
$\#E_u$	number of followees
R_{ee}	a ratio defined as the number of followers over that of followees, i.e., $\frac{\#R_u}{\#E_u}$
U_t	user type (as detailed in Table 2)

3.2 Behavior Feature Unlike *Basic Feature*, *Behavior Feature* shows several statistics regarding the user's retweeting behavior. Such statistics include:

- a value for the number of owned microblogs $\#B_u$
- a ratio R_{oc} that is defined as the number of retweeted microblogs over that of originally created, i.e., $\frac{\#(B_u|flag==1)}{\#(B_u|flag==0)}$
- a value showing the average number of retweeted microblogs per week: $\#W_r$

Table 2: Category of User Type

value	illustration
0	$\#E_u \leq 50$ & $\#R_u \leq 50$
1	$\frac{\#E_u}{\#R_u} \geq 5$
2	$\frac{\#R_u}{\#E_u} \geq 5$
3	other cases

- a normalized vector regarding the time distribution of a user's retweeting behavior: $P_t = (p'_0,\ p'_1,\ ...,\ p'_{11})$, where p'_0 is the probability that the retweeting activity happens from 0am to 2am, p'_1 is the probability that the retweeting activity happens from 2am to 4am, and so on.
- a normalized vector with respect to the gap distribution of a user's retweeting behavior: $P_g = (p''_0,\ p''_1,\ ...,\ p''_5)$, in which p''_0 is the probability that the gap between two retweeted microblogs is within 1 min. Ditto for p''_1 (1 min to 1 hour), p''_2 (1 to 12 hours), p''_3 (12 to 24 hours), p''_4 (24 to 48 hours) and p''_5 (more than 48 hours).

Hence, *Behavior Feature* H per user comes with:

$$H = (\#W_r,\ P_t,\ P_g) \tag{3.2}$$

where $\#W_r$, P_t and P_g are illustrated as above.

3.3 Interest Feature Different from the straightforward notions of *Basic Feature* and *Behavior Feature*, *Interest Feature* involves a process of labeling users by their interested topics. In short, with a given lexicon(made by some professionals) consisting of several *topics*, the interest feature of a user is a normalized vector, in which each dimension refers to the probability that the said user matches a *topic*.

Definition 3. *A lexicon L consists of a set of topics t such that each topic is associated with a set of cell words c. Each cell word depicts an aspect of the topic.*

Definition 4. *With a given user u, each microblog $b \in B_u$ could be decomposed into a set of words w.*

Definition 5. *The interest feature of a given user is a normalized vector*

$$P_f = (p_0,\ p_1,\ ...,\ p_{x-1}) \tag{3.3}$$

in which the said user matches x topics in lexicon and p_i refers to the similarity of the user and each matched topic (interest). The definition of such similarity shall be detailed in each scenario (explicit/implicit interest analysis, towards words/topics, etc).

Next, we shall present the "mining" process for interest features. In short, GruBa employs a well established lexicon to discover the explicit interests of users. In case no proper explicit interests are found, TF-IDF and Twitter-LDA are leveraged to explore the implicit interests, during which word2vector participates to provide the similarity between two words.

In GruBa, a word, either in the form of c or w, acts as the minimum unit for analysis. Hence, the similarity of a word pair (w, c), i.e., $sim(w, c)$, could be generalized to the similarity of a microblog against one topic $sim(b, t)$, and finally to a user versus each topic in lexicon $sim(u, t)$; topics with similarity satisfying certain thresholds are allocated to the user u and constitute the interests of u.

For instance, the following steps depict the "mining" process of the interest feature P_f for a user u.

Step 1: Each microblog of u is decomposed into a word set.

Step 2: Explicit interests are explored. Specifically, every word w is sent to match each cell word c of lexicon topics. If w and c are identical, $sim(w, c) = 1$. Otherwise, $sim(w, c) = 0$. As to the similarity of b against a lexicon topic t, it is:

$$sim(b, t) = \sum_{i,j} sim(w_i, c_j) \tag{3.4}$$

where $sim(w_i, c_j)$ refers to the similarity of a word pair.

If $sim(b, t)$ satisfies a certain threshold (3 by default) , topic t is labeled to microblog b; the user u is then discovered having an explicit interest (topic) t. Thus, by looking into the similarity of b against all topics in lexicon, the explicit interests of u is returned, in the form of interest feature (see Definition 5).

If none of $sim(b, t)$ could meet the threshold, i.e., explicit interest discovery over user u fails, go to **Step 3** and **Step 4** in parallel, so as to "mine" the implicit interests of u.

Step 3: A metric *TF-IDF weight* W_f is computed, i.e., employing TF-IDF (term-frequency and inverse document-frequency) to calculate the weight distribution of words in microblog b:

$$W_f = \{(w_i, p_i)\} \tag{3.5}$$

where w_i refers to a single word, of which the weight is p_i, with $\sum_i p_i = 1$.

To compute such weight p_i for word w_i, a metric p''_i

is first calculated as:

$$p''_i = \frac{|b_i|}{|b|} * log(\frac{|D_i|}{|D|}) \quad (3.6)$$

in which we use the operator $|\ |$ to measure the cardinality, such that $|b_i|$ is the occurrences of word w_i in microblog b and $|b|$ the total occurrences of all words in b. Ditto for $|D_i|$ and $|D|$, except that the scope is the overall dataset, rather than a single microblog b.

Hence, each word w_i shall get an initial weight of p''_i, upon which the normalization is performed and p_i is obtained, resulting the *TF-IDF weight* (see Definition 3.5). Go to **Step 5**.

Step 4: Similarly, another metric *Twitter-LDA weight* W_w is obtained, i.e., using Twitter-LDA [35] to result the word weight distribution of microblog b. Unlike TF-IDF, Twitter-LDA first trains the overall microblogs, allocating each microblog with a *tag*. The structure of *tag* is as follows:

$$W_t = \{(w'_i, p'_i)\} \quad (3.7)$$

where w'_i refers to a word in *tag* W_t, and p'_i is the probability that w'_i appears in microblogs with the said *tag*, with $\sum_i p'_i = 1$ ($|W_t| = 30$ in this work by default). Subsequently, W_t are leveraged to conclude W_w, i.e., $W_w = W_t$, which shares the format with that of W_f. Go to **Step 6**.

Step 5: TF-IDF based similarity is calculated. For example, the similarity (in the form of a value) of W_f over a single topic t in lexicon, written as $sim(W_f, t)$, is defined as:

$$sim(W_f, t) = \sum_i p_i * sim(w_i, t) \quad (3.8)$$

where $W_f = \{(w_i, p_i)\}$, $t = \{c_j\}$, and $sim(w_i, t)$ is the averaged word similarity $sim(w_i, c_j)$ returned by word2vector [22]. Go to **Step 7**.

Step 6: Accordingly, Twitter-LDA based similarity is available. Again, a single topic t in lexicon is used for yardstick and the similarity of W_w over t, written as $sim(W_w, t)$, is defined as:

$$sim(W_w, t) = \sum_i p'_i * sim(w'_i, t) \quad (3.9)$$

where W_w is a set of (w'_i, p'_i), t covers each c_j, and $sim(w'_i, t)$ is the averaged word similarity $sim(w'_i, c_j)$ returned by word2vector. Go to **Step 7**.

Step 7: Hence, the similarity of a microblog b against a lexicon topic t is given by:

$$sim(b, t) = \alpha * sim(W_f, t) + (1 - \alpha) * sim(W_t, t) \quad (3.10)$$

where the α is a parameter by which GruBa could set flexible priorities between TF-IDF and Twitter-LDA. Go to **Step 8**.

Step 8: Repeat the above steps (Step 1 to Step 7) for the microblog b over every topic in lexicon, i.e., $\forall t_k \in L$ results one similarity value of $sim(b, t_k)$. Such computation further extends to all the microblogs owned by user u, such that: $\forall b_m \in B_s(u), \forall t_k \in L$, there exists a similarity of $sim(b_m, t_k)$. Hence, the overall similarity of user u over lexicon topics $\{t\}$ (i.e., L), written as $S(u,\ L)$, could be denoted by a vector:

$$S(u, L) = (s_0,\ s_1,\ ...,\ s_{n-1}) \quad (3.11)$$

where n refers to the cardinality of L (i.e., number of topics in L) and s_k is the overall similarity of user u over topic t_k, which is given by:

$$s_k = \sum_m sim(b_m, t_k) \quad (3.12)$$

Among the n dimensions of $S(u,\ L)$, those with top x (3 in GruBa) similarity values are selected to label the implicit interests of user u, which results an x dimensional vector P_f as described in Definition 5. Similarly, interest features of all users are returned.

As a result, the *Feature Data* for every user u, written as $F(u)$, is given by:

$$F(u) = (I,\ H,\ P_f) \quad (3.13)$$

where I, H and P_f refer to *Basic Feature*, *Behavior Feature* and *Interest Feature* separately (see formulas 3.1, 3.2 and 5). And it could be written as a vector:

$$F(u) = (\#R_s, \#E_s, R_{ee}, \#B_s, R_{oc}, U_t, \#W_r, P_t, P_g, P_f) \quad (3.14)$$

where each dimension refers to a data item of GruBa.

4 User Clustering

Providing the *Feature Data*, User Clustering takes the charge of grouping each user concerned into a proper cluster. Algorithm 1 illustrates such overall procedure. The idea is to enumerate a number of clustering trials (line 4) and select the optimal solution with the best Silhouette coefficient value (v in line 14). In principle, each trial (referred by t in line 4) first performs a clustering task (line 5; to be detailed in section 4.1), resulting a cluster (by $l(u)$) for each user u (line 6); then, each user obtains a Silhouette coefficient value $v(u)$ stemmed from the in/out-cluster distances (lines 8–10; shall be illustrated in section 4.2); finally, the averaged Silhouette coefficient value of all users serves as the Silhouette coefficient value of the current trial,

Algorithm 1 User Clustering in GruBa

1: Input: *Feature Data* of users $\{F(u)\}$, the minimum/maximum number of clusters N_i and N_a
2: Output: Optimal user clustering result R
3:
4: **for all** $t \in [N_i, N_a]$ **do**
5: group users $\{u\}$ into t clusters by $\{F(u)\}$
6: clustering result $R'(t) = \{(u,\ l(u))\}$ with cluster info $l(u)$ for each user u
7: **for all** $u \in \{u\}$ **do**
8: in-cluster distance $d_i(u)$
9: out-cluster distance $d_o(u)$
10: Silhouette coefficient value $v(u) = \frac{(d_o - d_i)}{max(d_o,\ d_i)}$
11: **end for**
12: $v(t) = Avg\{v(u)\}$
13: **end for**
14: **if** $v(a) == Max\{v(t)\}$ **then**
15: $R = R'(a)$
16: **end if**
17: **return** R

written as $v(t)$ (line 12), by which the said selection process is conducted (line 14).

Next, we shall now first detail how GruBa performs the clustering task and subsequently illustrate the computation for the metric of Silhouette coefficient value.

4.1 Clustering in GruBa In GruBa, the clustering rests on an optimized K-Prototype [14] algorithm, named K-Gru in this work. Similar as K-Prototype, K-Gru randomly selects the cluster kernels among samples and employs the minimum distance between them to determine an initial result, upon which the clustering tasks are iterated until the results are stable.

Unlike K-Prototype that supports vector samples in which each dimension is of numerical/categorical, K-Gru could also handle the case where a dimension is one normalized vector. Recall the sample data for User Clustering, i.e., *Feature Data* in form of vectors (see formula 3.14), of which the data type regarding each dimension is shown as Table 3.

Table 3: Dimension Types in *Feature Data* Vector

type	data dimensions
numerical data	$\#R_u$, $\#E_u$, R_{ee}, $\#B_u$, R_{oc}, $\#W_r$
categorical data	U_t
normalized vectors	P_t, P_g, P_f

As aforementioned, the clustering of K-Gru rests on the distance between vector samples, where the dimensions are combined with numbers, categories and normalized vectors. For simplicity, we shall first illustrate the distance calculation of the simple vectors with mono data type on each dimension and then demonstrate that of complex vectors in K-Gru.

Given two numerical vectors $Y' = (y'_0, y'_1, ...)$ and $Z' = (z'_0, z'_1, ...)$, the Euclidean distance [12] between Y' and Z' is given by :

$$D_n(Y', Z') = \sum_e (y_e - z_e)^2 \tag{4.15}$$

As to the categorical vectors $Y'' = (y''_0, y''_1, ...)$ and $Z'' = (z''_0, z''_1, ...)$, the Hamiltonian distance [14] of Y'' and Z'' is:

$$D_h(Y'', Z'') = \sum_e H_e \tag{4.16}$$

where H_e refers to the Hamiltonian distance over each dimension, with $H_e = 1$ if y''_e and z''_e share the identical value, and $H_e = 0$ otherwise.

Regarding two vectors where each dimension is a normalized vector per se, Cosine Similarity is leveraged to compute the distance. Then, the distance between such two vectors $Y^* = (Y^*_0, Y^*_1, ...)$ and $Z^* = (Z^*_0, Z^*_1, ...)$ is:

$$D_v(Y^*, Z^*) = 1 - \sum_e Y^*_e \cdot Z^*_e \tag{4.17}$$

where $\cdot$ refers to the dot product operation between two normalized vectors Y^*_e and Z^*_e.

Hence, the said distance regarding the complex vectors ($Y = (Y_0, Y_1, ...)$ and $Z = (Z_0, Z_1, ...)$) in K-Gru, named GruBa Distance, could be deduced as:

$$D_g(Y, Z) = \sum_e G_e \tag{4.18}$$

where the distance on each dimension G_e is given by:

$$G_e = \begin{cases} (Y_e - Z_e)^2 & \text{if } Y_e/Z_e \text{ is numerical} \\ H_e \ (1 \ or \ 0) & \text{if } Y_e/Z_e \text{ is categorical} \\ Y_e \cdot Z_e & \text{if } Y_e/Z_e \text{ is of normalized vector} \end{cases} \tag{4.19}$$

4.2 Silhouette Coefficient Metric Computation In GruBa, Silhouette coefficient value serves as the fundamental criteria for the optimal clustering selection. Providing a clustering result, each user is associated with a cluster.

For a given user u of cluster l, we employ the vector Y to denote the *Feature Data* as in formula 3.14.

DEFINITION 6. *The in-cluster distance $d_i(u)$ is the average distance to all the other users in the same cluster, i.e., $\forall u'' \in l$ & $u \neq u''$:*

$$d_i(u) = Avg\{D_g(Y_u, Y''_u)\} \tag{4.20}$$

Definition 7. *The out-cluster distance $d_o(u)$ is measured as the minimum of the distances $\{d^*\}$ between u and other clusters ($\forall l' \neq l$), i.e.:*

$$d_o(u) = Min\{d^*(u, l')\} \tag{4.21}$$

where d^* is given by:

$$d^*(u, l') = Avg\{D_g(Y_u, Y'_u)\} \ \forall u' \in l' \tag{4.22}$$

Definition 8. *The Silhouette coefficient value $v(u)$ is thus concluded:*

$$v(u) \ = \ \frac{(d_o - d_i)}{max(d_o, \ d_i)} \tag{4.23}$$

Intuitively, a good clustering solution should result bigger d_o and smaller d_i, such that samples with obvious differences go to various clusters and vice versa. When d_o is far more than d_i, Silhouette coefficient value approaches to 1. Hence, the larger Silhouette coefficient value is, the better clustering performs, by which the optimal solution is selected.

5 Group based Behavior Modeling

Recall the central problem of GruBa, where the retweeting behaviors of users are modeled. Specifically, such model is built by Group Modeling for each user group and thus named as group model. To avoid ambiguity, we shall use the term of *items* to denote the data for training the group model. A given *item* is either positive or negative.

Definition 9. *An item E involves a microblog b and a user f such that $f \ \in R_{b.O}$, i.e., f is a follower of the owner of microblog b.*

$$E \in \begin{cases} \textit{positive items} & \textit{if } f \textit{ retweeted } b \\ \textit{negative items} & \textit{if } f \textit{ did not retweet } b \end{cases} \tag{5.24}$$

And the data of item E could be further divided into three parts.

(1) **User Info** contains a list of aforementioned metrics $\{\#R_u, \#E_u, R_{ee}, \#B_u, \#W_r\}$.

(2) **Microblog Info** considers the correlation between microblog contents and recent events, where the latter is returned by Ring [1]. The correlation metric C_h is in the form of a normalized vector with each dimension represents one event (similar as P_f in formula 3.3). Each event could be viewed as a topic t, over which the correlation of a microblog b could be obtained by formula 3.10.

(3) **Interaction Info** includes three correlation metrics. They are of microblog b versus the user u's *Interest Feature* $P_f(u)$ (a.k.a. long-term/stable interest in this work), b versus u's short-term interest $P_s(u)$ that is mined from u's recent microblogs (e.g., within 30 days) in the same manner of $P_f(u)$, and b's timestamp versus the time distribution of u's retweeting behavior P_t. In this paper, we see modeling retweeting behavior of group as a classification problem and use random forest classifier to solve the problem. The advantage of using classification model is that we can integrate different combinations of the features into the model conveniently. As a result, the obtained group behavior model could learn what does a positive/negative item look like over each metric mentioned above.

6 Performance Evaluation

In this section, we shall first detail the experimental setting of machine info, dataset and parameters; and then present the result and analysis, showing the benefit of GruBa against state of the art approaches.

6.1 Experimental Setting Experiments were run on a machine with two Intel Xeon E5C2630 2.4GHz CPUs and 64 GB of Memory, running 64 bit Windows 7 professional system. We have employed a real-world dataset Sina that consists of 24 million microblogs that are associated with 43.5K users.

With respect to the parameters of GruBa, we use the default values as mentioned in previous sections. Particularly, in Feature Extraction, for practical reason, we employed a smaller testing dataset to obtain the proper value of α for extracting *Interest Feature*; in User Cluster, we studied the clustering solutions with the minimum/maximum number of clusters 2 and 10; in Behavior Modeling, a user's recent 30 days microblogs are used for short-term interest analysis and popular words of the latest 24 hours are returned by Ring as the Hot Event keywords.

6.2 Result and Analysis Next, we shall report the performance of GruBa over each component.

Feature Extraction Fig. 2 shows the testing results of using various α values. Apparently, GruBa results the optimal results when α is 0.7, upon which the interest feature extracting is performed for the overall dataset with 43.5K users and 24 million microblogs.

User Clustering Fig. 3 depicts the Silhouette Coefficient Values of multiple clustering solutions, with the cluster number from 2 to 10. Specially, we used different testing datasets, with *Data* containing the overall 43.5K users and *Data1,...,5* contains 10K randomly selected users each. Except for *Data1*, solutions with 4

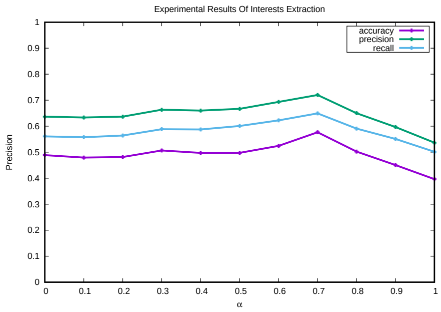


Figure 2: Testing Results with Various α.

clusters sweep.

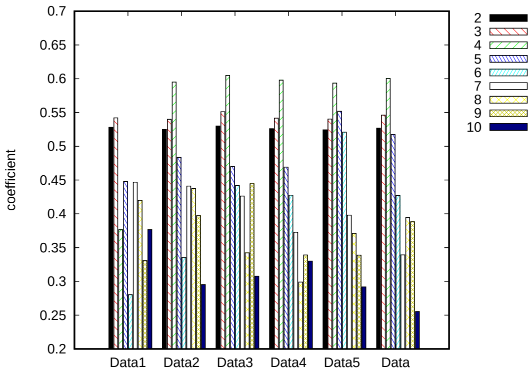


Figure 3: Silhouette Coefficient Values for Various Cluster Numbers over Different Testing Datasets.

Behavior Modeling Fig. 4 shows the performance of GruBa against state of the art approach LRC-BQ [31]. We compare the metrics of precision, recall, as well as F_1 score. LRC-BQ does not deal with user grouping. Hence, we not only study the modeling effect per group (i.e., “Group-One/Two/Three/Four” with user clustering), but also examine GruBa versus LRC-BQ in the case that all users are in mono group (i.e., “All-Users”). The results are interesting in that:

(1) With user clustering, GruBa performs better than LRC-BQ in most cases.

(2) For GruBa, having user clustering is better than the alternative mono group. Ditto for LRC-BQ.

Fig. 5 explores the performance of GruBa when using alternative data items for modeling. By default, GruBa uses “UI+II+MI”, i.e., items of users (UI), microblogs (MI) and interactions (II). How about using other combinations of the above item(s)? As shown in Fig. 5, the default setting wins in most cases.

Case Study for Feature Extraction In this section, we show the results of our demo system for user features extraction. Considering the huge amount of users, we carefully selected one typical user for analysis. Here we chose Mary(a famous drama and movie actor in China) as a representative. The features extraction result for Mary is displayed in Fig. 6. Then we can see the basic information of Mary from the result. Her nickname is Actor Mary() and she is from Beijing(). The number of followers of Mary is much more than the number of her friends. The Sina Microblog tag she made for herself is "actor"(). As the result of the long-term interest extraction shows, she is interested in stage performances(), drama performances(),films() and so on, which is consistent with her actual situation. What’s more, the probability distribution of tweeting and retweeting behavior on different time periods in one day indicates that her activities at night is much more than daytime. The time interval distribution of her behaviors is also shown in Fig. 6. The time interval is mostly distributed within 48 hours, which indicates that she is a active user. We developed a function for querying user’s short-term interest. Here we queried Mary’s short-term interest between 09/01/2016 and 09/30/2016, and got it as shown in Fig. 6. We analyzed her microblogs in the this period, she has been working tirelessly to promote the drama "Earl of Oolong Mountain"(). So the results are in line with expectations.

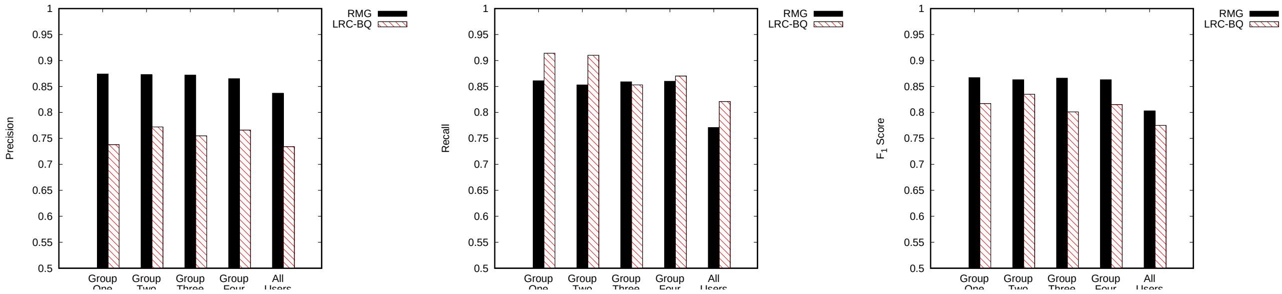
7 Related Work

In this section, we shall review related work in literature mainly from the aspects of analyzing features, mining groups and modeling behavior within the realm of social network modeling. As aforementioned, GruBa leverages the user features of basics, behavior and interest.

With respect to feature analysis, there have been existed works of mining users’ info, such as race [24], gender [3], age [23], political preference [24, 27, 18] and occupation [7, 6]. Our work, however, does not focus on the mining process per se; we use the mined info as the input for user clustering and group modeling.

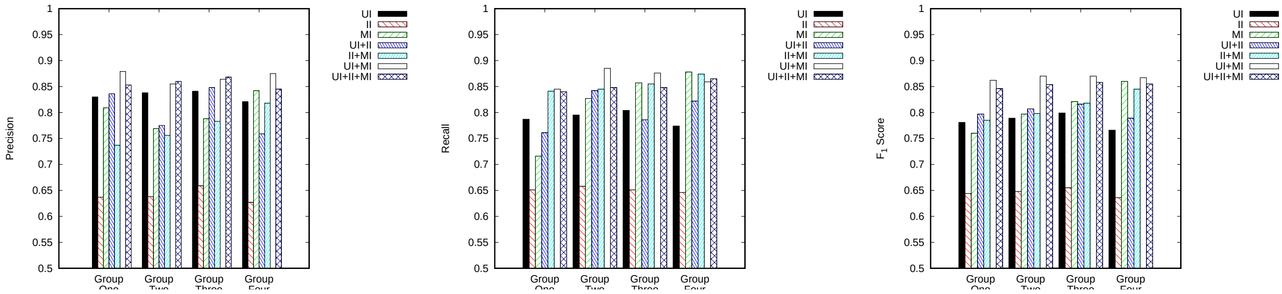
Studies of behavior analysis put emphasis on exploring the characteristics. For example, [17] proposed a model that can properly explain various time distributions of user behaviors by theoretical analysis; [11] studied the user activity distribution of one day/week; [4] provided the PowerWall distribution of Facebook users, identifying a number of surprising behaviors and anomalies. Considering the behavior characteristics, GruBa makes use of them to feed the modeling process.

There have been established work of extracting user interests. [20] mined the user interests by exploring keywords of microblogs with the aid of word frequency and machine translation. [29] proposed a method of



(a) Precision (b) Recall (c) F_1 Score

Figure 4: Performance of GruBa Versus LRC-BQ.



(a) Precision (b) Recall (c) F_1 Score

Figure 5: GruBa Performance of Using Various Data Items for Modeling.

extending the topic model to obtain use interests. Also, [21] used a knowledge base and [9] provided a solution of using hashtag for interest analysis. [19] summarized user interest by exploring the mentioned celebrities; Similarly, [2] leveraged the followed experts to result interest characteristics. Unlike the existed solutions, GruBa employs a cell lexicon to properly express user interest in which Twitter-LDA [35] and TF-IDF are employed.

Approaches of grouping users in social network could fall into a variety of categories. [24] grouped users by the info of race, political view and etc. [34] studied the social groups on Facebook and Wechat, resulting various patterns of group evolution. [28] proposed a time-varying factor to measure the affinity between users and groups such that proper group proposals are recommended. More recent studies also look into mining user communities. [30] employed matrix decomposition to mine user community; [25] and [13] considered followees info for user clustering; [26] proposed an incremental algorithm to mine user community using modular degree as the clustering yardstick. Providing the diversity of user features, GruBa employs feature of info, behavior and interest into user clustering.

The main problem of current approaches for behavior modeling lies in that the model is for either the overall users or a single user. [32, 8, 31] discovered that users' retweeting behavior is largely influenced by their followees, whereas [16] employed matrix decomposition, [15] used collaborative filtering methods and [33] leveraged statistical models for retweeting analysis. [10] and [5] focused on identifying whether the retweeting is fraudulent or of protest. Whereas our work GruBa builds the retweeting model for each user group, instead of the mono model for all users or one model per user.

8 Conclusions

In this work, we have presented GruBa, a system to model users' retweeting behavior in social media. GruBa departures from related works by grouping users into clusters, during which features of basics, behavior and interests are involved. Specially, we have studied interest features from various perspectives, such as long-term/recent interests and explicit/implicit interests. Last, we provided the performance evaluation of GruBa by using real-world datasets and comparing against state of the art approaches, showing its benefits.

References

[1] `http://ring.cnbigdata.org/`.

Figure 6: Visualization for User Features.

[2] P. Bhattacharya, M. B. Zafar, N. Ganguly, S. Ghosh, and K. P. Gummadi. Inferring user interests in the twitter social network. In *Eighth ACM Conference on Recommender Systems, RecSys '14*, pages 357–360, 2014.

[3] M. Ciot, M. Sonderegger, and D. Ruths. Gender inference of twitter users in non-english contexts. In *EMNLP*, pages 1136–1145, 2013.

[4] P. Devineni, D. Koutra, M. Faloutsos, and C. Faloutsos. Facebook wall posts: a model of user behaviors. *Social Netw. Analys. Mining*, 7(1):6:1–6:15, 2017.

[5] S. R. et al. Predicting online protest participation of social media users. *CoRR*, abs/1512.02968, 2015.

[6] Y. Fan, Y. Chen, K. Tung, K. Wu, and A. L. P. Chen. A framework for enabling user preference profiling through wi-fi logs. In *ICDE*, pages 1550–1551, 2016.

[7] Q. Fang, J. Sang, C. Xu, and M. S. Hossain. Relational user attribute inference in social media. *IEEE Trans. Multimedia*, 17(7):1031–1044, 2015.

[8] W. Feng and J. Wang. Retweet or not?: personalized tweet re-ranking. In *WSDM*, pages 577–586, 2013.

[9] W. Feng and J. Wang. We can learn your #hashtags: Connecting tweets to explicit topics. In *ICDE*, pages 856–867, 2014.

[10] M. Giatsoglou, D. Chatzakou, N. Shah, C. Faloutsos, and A. Vakali. Retweeting activity on twitter: Signs of deception. In *PAKDD*, pages 122–134, 2015.

[11] Z. Guo, Z. Li, H. Tu, and L. Li. Characterizing user behavior in weibo. In *Third FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing, MUSIC 2012*, pages 60–65, 2012.

[12] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques, 3rd edition.* Morgan Kaufmann, 2011.

[13] C. He, H. Ma, S. Kang, and R. Cui. An overlapping community detection algorithm based on link clustering in complex networks. In *Military Communications Conference (MILCOM), 2014 IEEE*, pages 865–870, 2014.

[14] Z. Huang. Clustering large data sets with mixed numeric and categorical values. In *PAKDD*, pages 21–34, 1997.

[15] B. Jiang, J. Liang, Y. Sha, R. Li, W. Liu, H. Ma, and L. Wang. Retweeting behavior prediction based on one-class collaborative filtering in social networks. In *SIGIR*, pages 977–980, 2016.

[16] B. Jiang, J. Liang, Y. Sha, and L. Wang. Message clustering based matrix factorization model for retweeting behavior prediction. In *CIKM*, pages 1843–1846, 2015.

[17] Z. Jiang, Y. Zhang, H. Wang, and P. Li. Understanding human dynamics in microblog posting activities. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(02):P02006, 2013.

[18] M. Kosinski, D. Stillwell, and T. Graepel. Private traits and attributes are predictable from digital records of human behavior. *PNAS*, 110(15):5802–5805, 2013.

[19] K. H. Lim and A. Datta. Interest classification of twitter users using wikipedia. In *Proceedings of the 9th International Symposium on Open Collaboration*, pages 22:1–22:2, 2013.

[20] Z. Liu, X. Chen, and M. Sun. Mining the interests of chinese microbloggers via keyword extraction. *Frontiers of Computer Science in China*, 6(1):76–87, 2012.

[21] M. Michelson and S. A. Macskassy. Discovering users' topics of interest on twitter: a first look. In *Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data, AND,(in conjunction with CIKM)*, pages 73–80, 2010.

[22] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[23] S. Park, S. P. Han, S. Huh, and H. Lee. Preprocessing uncertain user profile data: Inferring user's actual age from ages of the user's neighbors. In *ICDE*, pages 1619–1624, 2009.

[24] M. Pennacchiotti and A. Popescu. A machine learning approach to twitter user classification. In *Proceedings of the Fifth International Conference on Weblogs and Social Media*, 2011.

[25] Y. Ruan, D. Fuhry, and S. Parthasarathy. Efficient community detection in large networks using content and links. In *WWW*, pages 1089–1098, 2013.

[26] H. Shiokawa, Y. Fujiwara, and M. Onizuka. Fast algorithm for modularity-based graph clustering. In *AAAI*, 2013.

[27] S. Volkova, G. Coppersmith, and B. V. Durme. Inferring user political preferences from streaming communications. In *ACL*, pages 186–196, 2014.

[28] X. Wang, R. Donaldson, C. Nell, P. Gorniak, M. Ester, and J. Bu. Recommending groups to users using user-group engagement and time-dependent matrix factorization. In *AAAI*, pages 1331–1337, 2016.

[29] Z. Xu, R. Lu, L. Xiang, and Q. Yang. Discovering user interest on twitter with a modified author-topic model. In *Proceedings of the 2011 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2011*, pages 422–429, 2011.

[30] J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *WSDM*, pages 587–596, 2013.

[31] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li. Social influence locality for modeling retweeting behaviors. In *IJCAI*, pages 2761–2767, 2013.

[32] J. Zhang, J. Tang, J. Li, Y. Liu, and C. Xing. Who influenced you? predicting retweet via social influence locality. *TKDD*, 9(3):25:1–25:26, 2015.

[33] Q. Zhang, Y. Gong, Y. Guo, and X. Huang. Retweet behavior prediction using hierarchical dirichlet process. In *AAAI*, pages 403–409, 2015.

[34] T. Zhang, P. Cui, C. Faloutsos, Y. Lu, H. Ye, W. Zhu, and S. Yang. comengo: A dynamic model for social group evolution. *TKDD*, 11(4):41:1–41:22, 2017.

[35] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *European Conference on Information Retrieval*, pages 338–349, 2011.