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 blogs 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 [30, 7, 29]. 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 attracted the attention from both academia and industry.

Central to user behavior modeling, is the need to choose the unit of model (i.e., how many users share one model), as well as the variety of features to be selected for differentiating these units. Already, there exist work of building a single model for all the users [reference]. 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. Particularly, we study the retweeting behavior of users and our work can be generalized to other behaviors of like and comment as well. 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.

Contributions. The contributions of this work include:

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

(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, with a given user, we assume that blogs created or retweeted by his/her followees cover the overall candidates, from which the said user may retweet. All our results could straightforwardly generalize to alternative candidate scopes.

Definition 1. *A blog $B = (O, T, M, R)$ consists of the owner O (a.k.a. user in this paper) to whom B belongs (either created or retweeted), the timestamp T showing when B is generated, the blog message M and a bit R denoting B is retweeted (1) or originally created (0) by the owner O.*

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

†National Computer Network Emergency Response Technical Team/Coordination Center of China. E-mail:li.xiong@foxmail.com

DEFINITION 2. *A user $U = (B_s, R_s, E_s)$ consists of three sets regarding the user's blogs B_s, followers R_s and followees E_s separately. Each follower/followee per se refers to a user.*

The mapping between blog B and user U is a bilateral operation, i.e., $U = O(B)$ and $B \in B_s(U)$, through ID(s) of user and blog respectively.

Informally, providing a set of users $\{U\}$ and the associated blogs $\{B\}$, as well as a blog query b and a follower of $O(b)$ written as f, i.e., $f \in R_s(O(b))$, GruBa shall build a retweeting model for $(\{U\},\ \{B\})$, upon which Y/N 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 **??** shows the architectural components of GruBa, comprising three subsystems: Data Storage, Processing Runtime and Profile Demonstrator.

Data Storage. The underlying Data Storage subsystem stores data to be processed by GruBa, i.e., data of blogs and users, as shown in *Definition 1* and *2*.

Processing Runtime. In the heart of GruBa lies the Processing Runtime subsystem, which consists of three major components as follows.

1. Feature Extractor: By coalescing the blog data, each user is depicted by a bunch of features, which are grouped into three categories. They are features of *Info* (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 Extractor and serve as the input of User Clusterer.

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

3. Group Modeler: For each group obtained by User Clusterer, Group Modeler employs both positive and negative samples 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 subsystem for visualization. For the time being, Profile Demonstrator presents [To Be Completed].

3 Feature Extraction

With the underlying data in Data Storage subsystem, Feature Extractor is responsible for "mining" the user characteristics, resulting three features for each user. These features are *Info Feature*, *Behavior Feature* and *Interest Feature*, which constitute the *Feature Data* in GruBa, i.e., *Feature Data* = $\{$*Info Feature*, *Behavior Feature*, *Interest Feature*$\}$.

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

$$I = (\#R_s,\ \#E_s,\ R_{ee},\ \#B_s,\ R_{oc},\ U_t) \tag{3.1}$$

where each variable is illustrated in Table 1. Specifically, we use $\#(B_s|R(B) == 1)$ and $\#(B_s|R(B) == 0)$ to represent the number of retweeted blogs and blogs that are originally created by the user.

Table 1: Illustration of Variables in Info Feature

variable	illustration
$\#R_s$	number of followers
$\#E_s$	number of followees
R_{ee}	a ratio defined as $\frac{\#R_s}{\#E_s}$
$\#B_s$	number of blogs owned by the user
R_{oc}	a ratio defined as $\frac{\#(B_s\|R(B)==1)}{\#(B_s\|R(B)==0)}$
U_t	user type (as detailed in Table 2)

Table 2: Category of Info

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

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

- a value showing the average number of retweeted blogs per week: $\#W_r$

- a normalized vector regarding the time distribution of a user's retweeting behavior: $P_t = (p'_0,\ p'_1,\ \ldots,\ p'_{11})$, 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: $P_g = (p''_0,\ p''_1,\ \ldots,\ p''_5)$, in which p''_0 is the probability that the gap between two retweeted blogs 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 *Info Feature* and *Behavior Feature*, *Interest Feature* involves a process of labeling users by their interested topics. In short, with a given lexicon 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 $L = \{t\}$ consists of a set of topics while each topic $t = \{c\}$ is associated with a list of cell words $\{c\}$. Each cell word c refers to one unique word in L.*

DEFINITION 4. *With a given user u, each blog $b \in B_s(u)$ could be decomposed into a set of words $\{w\}$, i.e., $b = \{w\}$.*

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

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

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).

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 blog 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 blog of u is decomposed into a word set, i.e., $b = \{w\}$ where $b \in B_s(u)$.

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 blog 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 blog 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:

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

in which we use the operator $|\ |$ to measure the cardinality, such that $|b_i|$ is the occurrences of word w_i in blog 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 blog 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 [reference] to result the word weight distribution of blog b. Unlike TF-IDF [reference], Twitter-LDA [reference] first trains the overall blogs, allocating each blog with a *tag*. The structure of *tag* is as follows:

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

where w'_i refers to a word in *tag* W_t, and p'_i is the probability that w'_i appears in blogs with the said *tag*, with $\sum_i p'_i = 1$ ($|W_t| = 30$ 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 [reference] 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:

$$sim(W_f, t) = \sum_i p_i * sim(w_i, t) \tag{3.8}$$

where $W_f = \{(w_i, p_i)\}$, $t = \{c_j\}$, and:

$$sim(w_i, t) = \sum_j sim(w_i, c_j) \tag{3.9}$$

Go to *Step 7.*

Step 6: Accordingly, Twitter-LDA [reference] 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:

$$sim(W_w, t) = \sum_i p'_i * sim(w'_i, t) \tag{3.10}$$

where $W_w = \{(w'_i, p'_i)\}$, $t = \{c_j\}$, and:

$$sim(w'_i, t) = \sum_j sim(w'_i, c_j) \tag{3.11}$$

Go to *Step 7.*

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

$$sim(b, t) = \alpha * sim(W_f, t) + (1-\alpha) * sim(W_t, t) \tag{3.12}$$

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

Step 8: Repeat the above steps (Step 1 to Step 7) for the blog 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 blogs 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:

$$S(u, L) = (s_0,\ s_1,\ \ldots,\ s_{n-1}) \tag{3.13}$$

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:

$$s_k = \sum_m sim(b_m, t_k) \tag{3.14}$$

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:

$$F(u) = (I,\ H,\ P_f) \tag{3.15}$$

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

$$F(u) = (\#R_s, \#E_s, R_{ee}, \#B_s, R_{oc}, U_t, \#W_r, P_t, P_g, P_f) \tag{3.16}$$

where each dimension refers to a data item of GruBa.

4 User Clustering

Providing the *Feature Data*, User Clusterer 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 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 coefficient value $v(u)$ stemmed from the in/out-cluster distances (lines 8–10; shall be illustrated in section 4.2); finally, the averaged coefficient value of all users serves as the coefficient value of the current trial, 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 regarding the metric of coefficient value.

4.1 Clustering in GruBa In GruBa, the clustering rests on an optimized K-Prototype [12] 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

Algorithm 1 User Clustering in GruBa

Input: *Feature Data* of users $\{F(u)\}$, the minimum/maximum number of clusters N_i and N_a
Output: Optimal user clustering result R

for all $t \in [N_i, N_a]$ **do**
  group users $\{u\}$ into t clusters by $\{F(u)\}$
  clustering result $R'(t) = \{(u,\ l(u))\}$ with cluster info $l(u)$ for each user u
  for all $u \in \{u\}$ **do**
    in-cluster distance $d_i(u)$
    out-cluster distance $d_o(u)$
    coefficient value $v(u) = \frac{(d_o - d_i)}{max(d_o,\ d_i)}$
  end for
  $v(t) = Avg\{v(u)\}$
end for
if $v(a) == Max\{v(t)\}$ **then**
  $R = R'(a)$
end if
return R

one normalized vector. Recall the sample data for User Clusterer, i.e., *Feature Data* in form of vectors (see formula 3.16), of which the data type regarding each dimension is shown as Table 3.

Table 3: Types of Dimensional Data in *Feature Data* Vector

type	data dimensions
numerical data	$\#R_s$, $\#E_s$, R_{ee}, $\#B_s$, 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 O's Distance [reference] between Y' and Z' is given by :

$$D_n(Y', Z') = \sum_e (y_e - z_e)^2 \tag{4.17}$$

As to the categorical vectors $Y'' = (y''_0, y''_1, ...)$ and $Z'' = (z''_0, z''_1, ...)$, the H's Distance [reference] of Y'' and Z'' is:

$$D_h(Y'', Z'') = \sum_e H_e \tag{4.18}$$

where H_e refers to the H's 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 [reference] 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:

$$D_v(Y^*, Z^*) = \sum_e Y^*_e \cdot Z^*_e \tag{4.19}$$

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 G's Distance, could be deduced as:

$$D_g(Y, Z) = \sum_e G_e \tag{4.20}$$

where the distance on each dimension G_e is given by:
(4.21)

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

4.2 Coefficient Metric Computation In GruBa, 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 3.16.

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

$$d_i(u) = Avg\{D_g(Y_u, Y''_u)\} \tag{4.22}$$

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.:*

$$d_o(u) = Min\{d^*(u, l')\} \tag{4.23}$$

where d^* is given by:

$$d^*(u, l') = Avg\{D_g(Y_u, Y'_u)\}\ \forall u' \in l' \tag{4.24}$$

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

$$v(u) = \frac{(d_o - d_i)}{max(d_o,\ d_i)} \tag{4.25}$$

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, coefficient value approaches to 1. Hence, the larger coefficient value is, the better clustering performs, by which the optimal solution is selected.

5 Group based Behavior Modelling

Recall the central problem of GruBa, where the retweeting behaviors of users are modeled. Specifically, such model is built by Group Modeler 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 blog b and a user f such that $f \in R_s(O(b))$, i.e., f is a follower of b's owner.*

$$(5.26) \quad E \in \begin{cases} \textit{positive items} & \textit{if } f \textit{ retweeted } b \\ \textit{negative items} & \textit{if } f \textit{ did not retweet } b \end{cases}$$

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

- *User Info*: contains a list of aforementioned metrics $\{\#R_s, \#E_s, R_{ee}, \#B_s, \#W_r\}$.
- *Blog Info*: consists of a metric C_h, referring to the correlation between blog contents and recent events returned by Ring [1]. C_h is in the form of a normalized vector with each dimension represents one event (similar as P_f in formula 3.3). Specifically, each event could be viewed as a topic t, over which the correlation of a blog b could be obtained by formula 3.12.
- *Interaction Info*: includes three correlation metrics. They are of blog 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 blogs (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.

As a result, the obtained group model could learn what does a positive/negative item look like over each metric mentioned above.

6 Performance Evaluation

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 blogs 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 Extractor, 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 Modeler, a user's recent 30 days blogs 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 Extractor Fig. 1 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 blogs.

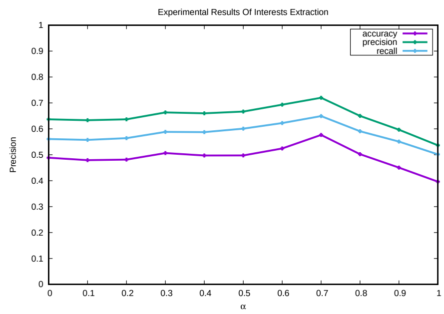


Figure 1: Testing Results with Various α.

User Clusterer Fig. 2 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 clusters sweep.

Behavior Modeler Fig. 3 shows the performance of GruBa against state of the art approach LRC-BQ [29]. 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.,

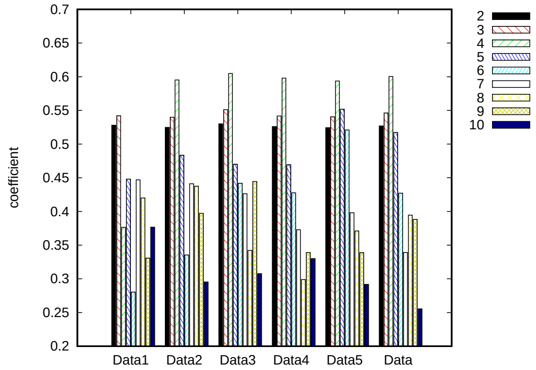


Figure 2: Silhouette Coefficient Values for Various Cluster Numbers over Different Testing Datasets.

"All-Users"). The results are interesting in that:

- With user clustering, GruBa performs better than LRC-BQ in most cases.
- For GruBa, having user clustering is better than the alternative mono group. Ditto for LRC-BQ.

Fig. 4 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), blogs (MI) and interactions (II). How about using other combinations of the above item(s)? As shown in Fig. 4, the default setting wins in most cases.

7 Related Work

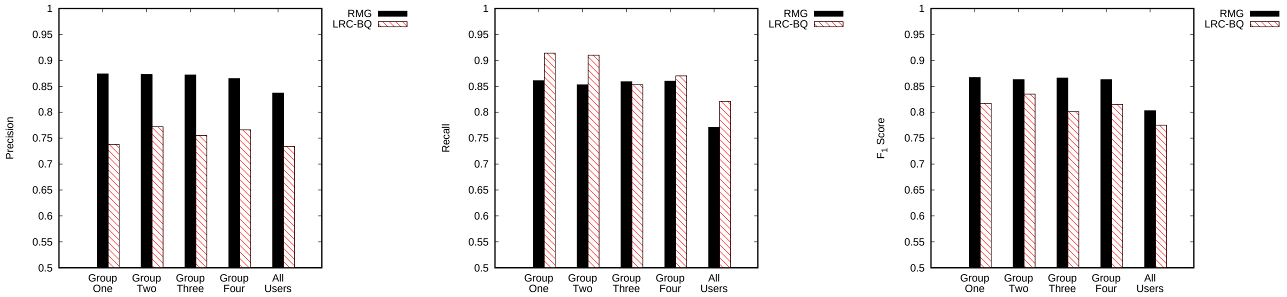
In this section, we will introduce the related work of user modeling in social network from the aspects of user features analysis, user groups mining and user behaviors modeling. Social media users's features we mentioned here include users's basic information, behavior features and interests features.

Users's basic information we mentioned here include the users's gender, age, region, occupation and other personal information. There are many researches for analyzing users's basic information, such as users's racial information analysis [21], users's gender inference [3], users's actual age inference [20], user policy orientation analysis (e.g. [21, 25, 16]), user's geo-location and occupation mining [6, 5], etc. Here, we did not do basic information mining for users. We used the user's basic information as important features for user clustering and behavior modeling.

Social medias users's behaviors mainly refers to the tweeting and retweeting behaviors. Users's behaviors also have certain characteristics and regularity, which have been proved by a lot of existing work. Jiang et al. proposed a behavior dynamics model [15], theoretical analysis shows this model can properly explain various heavy-tailed inter-event time distributions, including a regular power law and some non-power-law deviations. Guo Z et al.[10] analyzed the behavior of microblog users, the difference between the activity of users in different periods, and obtained the distribution of individual behavior on time. Pravallika Devineni et al. analyzed the wall activities of users focusing on identifying common patterns and proposed PowerWall distribution [4]. What's more, it helped them spot surprising behaviors and anomalies. Considering the regularity of user behavior on time, we extracted characteristics of the user behavior as important features for user clustering and behavior modeling.

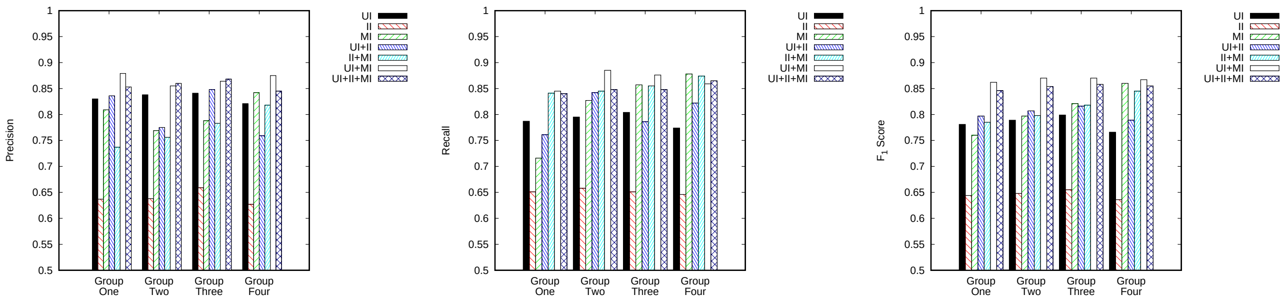
As for users's interest, many methods has been proposed to do interests extraction. Zhiyuan et al.[18] modeled users's interests through mining keywords. They extracted keywords from the users's microblogs by the combination of words frequency and translation model. Xu Z et al.[27] proposed a method which extended user topic model to analyze users's interests. Michelson M et al.[19] analyzed interests based on a knowledge base. They used a knowledge base to identify and classify the entities in twitters of one user, then generate the user's interests category subtree to express his interests. Lim K H et al.[17] analyzed the celebrities a user mentioned, then they got the preference degree of the user in different interests categories. Bhattacharya P et al.[2] proposed a method to extract a user's interests by analyzing the experts he followed. By digging a list of certain topics of the custom experts the user follows, they got the user's interests profiling. Wei Feng et al.[8] studied the methods mapping tweets to hashtags to get users's preferences for hashtags. The existing methods for interest extraction have some shortcomings. The granularity of the result obtained by the methods based on keyword extraction is too small, and the result obtained by the user topic model is implicit and difficult to display. What's more, for the lack of a complete knowledge base, it's not easy to do interests extraction by using knowledge base. To solve the problems, we employed a cell lexicon to express user's interest. What's more, we combined Twitter-LDA [33] and TF-IDF to extract the distribution of users's interest.

There have also many studies on the user groups analysis. Some researches studied how to classify users under a specific situation. For instance, Marco Pennacchiotti et al.[21] classified users by race, political tendencies and so on. [32] analyzed the evolution of social groups, such as Facebook groups and Wechat groups, and proposed a new model for group evolution, which can provide insights about different evolution pattern-



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

Figure 3: Performance of GruBa Versus LRC-BQ.



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

Figure 4: GruBa Performance of Using Various Data Items for Modeling.

s of social groups. Xin Wang et al. [26] implemented a time-varying factorization to measure the user-group affinity for recommending groups to users. In addition, research for users community mining is a hot spot. Jaewon Yang et al.[28] proposed a method based on non-negative matrix decomposition to mine user community. Yiye Ruan et al.[23] considered user's friends and user's text content into their method when measuring the similarity between users for clustering. He et al. [11] only considered the relationship between friends, and used the edge aggregation coefficient as a measure of clustering. Hiroaki Shiokawa et al.[24] used the modular degree as a clustering standard and proposed an incremental algorithm to mine user community. As we can see, current analysis of user groups are diverse. However, users in social networks have multiple features, and many existing methods do not take advantage of these features to group users. Here we took user's basic, behavioral and interest information into account when performing user clustering, and we got distinguish groups with different statistical characteristics.

As for behavior modeling, Jing Zhang et al.[30] proposed a method of analyzing users's retweeting behavior from the perspective of influence. Some researchers (e.g. [7], [29]) find that the user's reposting behavior is largely influenced by the relationship between friends, so they consider user's friend structure information into their models. Bo Jiang et al. [14] proposed a method that combined matrix decomposition and microblogs clustering to analyze the user's reposting behavior. Zhang et al.[31] proposed non-parametric statistical models to combine structural, textual, and temporal information together to predict reposting behavior. Considering the situation that users's not reposting does not mean that users are not interested in it, the users may just did not see it, [13] applied a collaborative filtering method to the reposting behavior analysis. Maria Giatsoglou et al. explore the identification of fraudulent and genuine retweet threads and developed a realistic generator that mimics the behaviors of both honest and fraudulent users [9]. Suhas Ranganath et al. [22] were inspired by sociological theories of protest participation and proposed a framework to predict from the users past status messages and interactions whether the next post of the user will be a declaration of protest. They evaluated the framework using data from Twitter on protests during the recent Nigerian elections and demonstrated that it could effectively work. Considering the fact that the amount of users in Social Medias is so huge, modeling for each user is not a good idea. However, modeling for a single user may make the model too particular. In addition, modeling for the whole users makes our model inaccuracy. So we divided users into several groups by users clustering. Then we get different behavior model for each group users respectively.

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 info, 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/.

[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, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014*, pages 357–360, 2014.

[3] M. Ciot, M. Sonderegger, and D. Ruths. Gender inference of twitter users in non-english contexts. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, 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] 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 *32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16-20, 2016*, pages 1550–1551, 2016.

[6] 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.

[7] W. Feng and J. Wang. Retweet or not?: personalized tweet re-ranking. In *Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013*, pages 577–586, 2013.

[8] W. Feng and J. Wang. We can learn your #hashtags: Connecting tweets to explicit topics. In *IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31 - April 4, 2014*, pages 856–867, 2014.

[9] M. Giatsoglou, D. Chatzakou, N. Shah, C. Faloutsos, and A. Vakali. Retweeting activity on twitter: Signs of deception. In *Advances in Knowledge Discovery and Data Mining - 19th Pacific-Asia Conference, PAKDD 2015, Ho Chi Minh City, Vietnam, May 19-22, 2015, Proceedings, Part I*, pages 122–134, 2015.

[10] 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, Vancouver, Canada, June 26-28, 2012*, pages 60–65, 2012.

[11] 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. IEEE, 2014.

[12] Z. Huang. Clustering large data sets with mixed numeric and categorical values. In *Proceedings of the 1st pacific-asia conference on knowledge discovery and data mining,(PAKDD)*, pages 21–34. Singapore, 1997.

[13] 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 *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 977–980, 2016.

[14] B. Jiang, J. Liang, Y. Sha, and L. Wang. Message clustering based matrix factorization model for retweeting behavior prediction. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 1843–1846, 2015.

[15] 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.

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

[17] K. H. Lim and A. Datta. Interest classification of twitter users using wikipedia. In *Proceedings of the 9th International Symposium on Open Collaboration, Hong Kong, China, August 05 - 07, 2013*, pages 22:1–22:2, 2013.

[18] 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.

[19] 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 2010, Toronto, Ontario, Canada, October 26th, 2010 (in conjunction with CIKM 2010)*, pages 73–80, 2010.

[20] 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 *Proceedings of the 25th International Conference on Data Engineering,*

ICDE 2009, March 29 2009 - April 2 2009, Shanghai, China, pages 1619–1624, 2009.

[21] 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, Barcelona, Catalonia, Spain, July 17-21, 2011*, 2011.

[22] S. Ranganath, F. Morstatter, X. Hu, J. Tang, and H. Liu. Predicting online protest participation of social media users. *CoRR*, abs/1512.02968, 2015.

[23] Y. Ruan, D. Fuhry, and S. Parthasarathy. Efficient community detection in large networks using content and links. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 1089–1098, 2013.

[24] H. Shiokawa, Y. Fujiwara, and M. Onizuka. Fast algorithm for modularity-based graph clustering. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA.*, 2013.

[25] S. Volkova, G. Coppersmith, and B. V. Durme. Inferring user political preferences from streaming communications. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 186–196, 2014.

[26] 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 *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 1331–1337, 2016.

[27] 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, Campus Scientifique de la Doua, Lyon, France, August 22-27, 2011*, pages 422–429, 2011.

[28] J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013*, pages 587–596, 2013.

[29] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li. Social influence locality for modeling retweeting behaviors. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 2761–2767, 2013.

[30] 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.

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

[32] 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.

[33] 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. Springer, 2011.

9 Appendix: Extra Experiments

9.1 Case Study for Feature Extraction

9.2 Case Study for User Clustering