Incorporating User Grouping into Retweeting Behavior Modeling

PaperID: 201

Abstract. The variety among massive 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 system, 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/short-term interests and explicit/implicit interests. We have evaluated the performance of GruBa using datasets of real-world social networking applications, and shown its benefits.

Keywords: user grouping · social networks · behavior modeling

1 Introduction

Social media is overwhelming nowadays, and popular social networks, e.g., Facebook, Twitter and Weibo, have attracted massive users. These users behave variously, knowledge of which is significant for various applications such as recommendation system and activity analysis. Hence there is an emergent 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 right granularity of model (i.e., how many users share one model), as well as the variety of features to be utilized for differentiating users. Already, there exist works of building a single model for all the users [**?**,**?**]. 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 observation 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 readily 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 behavior analysis. This motivates us to incorporate user grouping into the retweeting behavior modeling, filling the gap of existed studies that build a single model for all users.

The contributions of our work are as follows:

(1) We present a system named GruBa with the novel perspective to model user behaviors over groups instead of a single model for all users.

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

(3) We offer a clustering method K-Gru to deal with complex vectors, serving as an extension for standard K-Prototype algorithm.

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

Our paper is organized as follows. Section **??** gives the problem formulation and system overview, followed by detailed explanations in Sections **??**, **??** and **??**. Section **??** is performance evaluation, followed by related work in Section **??** and conclusions in Section **??**.

2 Overview of System GruBa

2.1 Problem Formulation

We consider the retweeting behavior of users 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* $M_b = (O, T, M, flag)$ *has its owner* O *(a.k.a. user in this paper) to whom* M_b *belongs (either tweeted or retweeted), the generated time* T *of* M_b*, the message context* M*, and* $flag$ *denoting* M_b *is retweeted or originally tweeted by* O*. Here we use 1 and 0 to denote retweeted and tweeted, respectively.*

Definition 2. *Given a user* u*, we adopt* B_u*,* R_u *and* E_u *to represent her/his microblogs* B_u*, followers* R_u *and followees* E_u*, respectively, in which a follower/followee is a user.*

Note that $M_b.O$ is a user, and B_u is a set of microblogs.

Providing a set of users $\mathbb{U}$ and their associated microblogs $\mathbb{B}$, system GruBa builds a retweeting model for each group of $\mathbb{U}$, such that given a microblog b and a follower f of its onwer $b.O$, i.e., $f \in R_{b.O}$, 1 or 0 is returned regarding whether f retweets b or not.

2.2 GruBa Framework

System GruBa is designed from the ground up as a system for modeling users' retweeting behavior in social media, and Figure **??** shows the architectural components of GruBa.

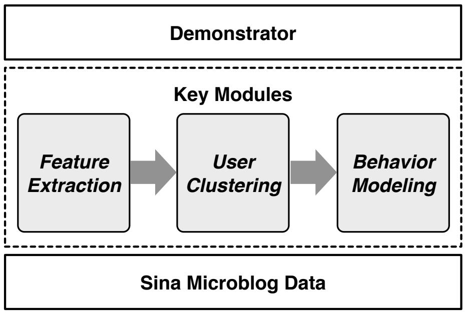


Fig. 1: GruBa Architecture

Sina Microblog Data. It is the data crawled to be processed by GruBa, i.e., data of microblogs and users.

Key Modules. GruBa consists of three key modules.

(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., long and short term interests, as well as explicit and implicit interests). These features are extracted from the stored Sina Weibo data by Feature Extraction module, and serve as the input of the 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 builds a model by employing both positive and negative samples (i.e., microblogs labeled with retweeted and not retweeted), on which the user retweeting behaviors are also tested.

Demonstrator. At the top layer of GruBa, it is the Demonstrator for visualizing all aspects of the system, e.g., profiling of user groups.

The distinctive feature of System GruBa models user retweeting behaviors over groups instead of a single model for all the users [**?**,**?**].

3 Feature Extraction

With the underlying Sina Microblog Data, the Feature Extraction module is responsible for mining the user characteristics, and produces three classes of features for each user: *Basic Feature*, *Behavior Feature* and *Interest Feature*, referred to as *Feature Data* in GruBa.

3.1 Basic Feature

The *Basic Feature* employs a vector I to depict the basic characteristics of a user u.

$$I_u = (G_u, P_u, \#R_u, \#E_u, R_{ee,u}, U_{t,u}), \tag{1}$$

Table 1: Illustration of Variables in Basic Feature

Variables	**Illustration**
G_u	gender of user u
P_u	province of user u
$\#R_u$	number of followers
$\#E_u$	number of followees
$R_{ee,u}$	a ratio defined as the number of followers over that of followees,i.e.,$\frac{\#R_u}{\#E_u}$
$U_{t,u}$	user type (as illustrated in Table **??**)

Table 2: Category of User Type

Types	**Illustration**
0	$\#E_u \leq 50$ **and** $\#R_u \leq 50$
1	$\frac{\#E_u}{\#R_u} \geq 5$
2	$\frac{\#R_u}{\#E_u} \geq 5$
3	other cases

in which the variable details are illustrated in Table **??**.

3.2 Behavior Feature

Unlike *Basic Feature*, the *Behavior Feature* of a user u is certain statistics regarding the retweeting behavior of u, shown below:

(a) the number of owned microblogs $\#B_u$,

(b) the ratio $R_{oc,u}$ that is the number of retweeted microblogs over that of originally tweeted, i.e., $\frac{\#(B_u|flag==1)}{\#(B_u|flag==0)}$,

(c) the average number of retweeted and tweeted microblogs per week: $\#W_{r,u}$ and $\#W_{t,u}$,

(d) the normalized vectors regarding the time distribution of a user's retweeting/tweeting behavior: $P_{rt,u} = (p'_{r0},\ p'_{r1},\ ...,\ p'_{r11})$, $P_{tt,u} = (p'_{t0},\ p'_{t1},\ ...,\ p'_{t11})$, where p'_{r0}/p'_{t0} is the probability that the retweeting/tweeting activity happens from 0am to 2am, p'_{r1}/p'_{t1} is the probability that the retweeting/tweeting activity happens from 2am to 4am, and so on, and

(e) the normalized vectors with respect to the gap distribution of a user's retweeting/tweeting behavior: $P_{rg,u} = (p''_{r0},\ p''_{r1},\ ...,\ p''_{r5})$, $P_{tg,u} = (p''_{t0},\ p''_{t1},\ ...,\ p''_{t5})$, in which p''_{r0}/p''_{t0} is the probability that the gap between two retweeted/tweeting microblogs is within 1 min. Ditto for p''_{r1}/p''_{t1} (1 min to 1 hour), p''_{r2}/p''_{t2} (1 to 12 hours), p''_{r3}/p''_{t3} (12 to 24 hours), p''_{r4}/p''_{t4} (24 to 48 hours) and p''_{r5}/p''_{t5} (more than 48 hours).

In summary, the *Behavior Feature* H_u of user u consists of the following:

$$(\#B_u, R_{oc,u}, \#W_{r,u}, P_{rt,u}, P_{rg,u}, \#W_{t,u}, P_{tt,u}, P_{tg,u}). \qquad (2)$$

3.3 Interest Feature

Different from the slightly straightforward *Basic Feature* and *Behavior Feature*, *Interest Feature* involves a process of labeling users with their interested topics based on their tweeted and retweeted microblogs. In short, with a given lexicon (made by some professionals) consisting of several *topics*, the interest feature of a user u is a normalized vector, in which each entry refers to the degree that u is interested in the corresponding *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, in which each cell word depicts an aspect of the topic.*

Definition 4. *For a user u, each microblog $b \in B_u$ is represented by a set of words w.*

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

$$(p_0, p_1, \dots, p_{x-1}), \tag{3}$$

in which user u matches x topics in lexicon L, and p_i refers to the degree that u is interested in topic i.

The similarity $p_i \quad (i \in [0, x-1])$ will 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 $sim(w, c)$ of a word pair (w, c) could be generalized to the similarity of a microblog against one topic $sim(b, t)$, and finally to a user u versus each topic t in lexicon $sim(u, t)$; topics with similarity satisfying certain thresholds are allocated to user u and constitute the interests of u.

System GruBa employs a well established lexicon to discover the explicit interests of users. When no proper explicit interests are found, two algorithms are leveraged to identify implicit interests: (1) TF-IDF (Term-Frequency and Inverse Document-Frequency) [**?**], and (2) Twitter-LDA [**?**] (a method to discover topics from Twitter, by applying the standard Latent Dirichlet Allocation (LDA) model [**?**]), both of which adopt word2vector [**?**] to measure word similarity.

We now explain the detailed process for mining the interest features of a user.

Step 1: Each microblog b in B_u of user u is decomposed into a word set WS.

Step 2: Explicit interests are explored. Specifically, every word w in WS is sent to match each cell word c of lexicon topics. If w and c are identical, $sim(w, c) = 1$, and $sim(w, c) = 0$, otherwise.

The similarity of b with a lexicon topic t is defined as

$$sim(b, t) = \sum_{i,j} sim(w_i, c_j). \tag{4}$$

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 L, the explicit interests of u is derived, in the form of interest feature (Definition **??**).

Step 3:. If the $sim(b,t)$ in **Step 2** cannot meet the threshold, i.e., explicit interest discovery over user u fails, the implicit interests of user u are further mined, by running the following steps 3.1&3.2 in parallel.

Step 3.1: A metric *TF-IDF weight* W_f is computed by employing TF-IDF to calculate the weight distribution of words in microblog b:

$$W_f = \{(w_i, p_i)\}, \tag{5}$$

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:

$$p_i'' = \frac{|b_i|}{|b|} * \log(\frac{|D|}{|D_i|+1}), \tag{6}$$

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. $|D|$ is the total number of microblogs in the dataset and $|D_i|$ is the number of microblogs where $|b_i|$ appears. 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*.

TF-IDF based similarity is then calculated. For example, the similarity of W_f over a single topic t in lexicon, written as $sim(W_f, t)$, is defined as formula **??**. Here $W_f = \{(w_i, p_i)\}$, $t = \{c_j\}$, VEC_{W_f} is defined as formula **??**, VEC_t is defined as formula **??**, where N_t is the number of words in topic t and $vec[w]$ is the word vector of word w returned by word2vector [**?**].

$$sim(W_f, t) = VEC_{W_f} \cdot VEC_t, where \tag{7}$$

$$VEC_{W_f} = \sum_i p_i * vec[w_i], and \tag{8}$$

$$VEC_t = \sum_j \frac{1}{N_t} * vec[c_j]. \tag{9}$$

Step 3.2: Similarly, another metric *Twitter-LDA weight* W_w is obtained by using Twitter-LDA 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:

$$W_t = \{(w_i', p_i')\}, \tag{10}$$

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.

Step 4: Hence, the similarity of a microblog b against a lexicon topic t is :

$$sim(b,t) = \alpha * sim(W_f, t) + (1-\alpha) * sim(W_t, t), \tag{11}$$

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

Step 5: Repeat Steps 1 to 4 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:

$$S(u, L) = (s_0, s_1, \ldots,\ s_{n-1}), \tag{12}$$

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:

$$s_k = \sum_m sim(b_m, t_k). \tag{13}$$

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_u as described in Definition **??**. Similarly, interest features of all users are returned.

As a result, the *Feature Data* F_u for a user u is

$$F_u = (I_u, H_u, P_u), \tag{14}$$

where I_u, H_u and P_u are the *Basic Feature*, *Behavior Feature* and *Interest Feature* of u, respectively.

4 User Clustering

Providing the *Feature Data*, the User Clustering module takes the charge of grouping each user concerned into a proper cluster, as illustrated in Algorithm **??**. The idea is to enumerate a number of clustering trials (line **??**) and select the optimal solution with the best Silhouette coefficient value (v in line **??**). In principle, each trial (referred by t in line **??**) first performs a clustering task (line **??**; to be detailed in section **??**), resulting a cluster (by $l(u)$) for each user u (line **??**); then, each user obtains a Silhouette coefficient value $v(u)$ stemmed from the in/out-cluster distances (lines **??**–**??**; shall be illustrated in section **??**); finally, the averaged Silhouette coefficient value of all users serves as the Silhouette coefficient value of the current trial, written as $v(t)$ (line **??**), by which the said selection process is conducted (line **??**).

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 K-Gru: Clustering in GruBa

In GruBa, the clustering rests on an optimized K-Prototype [**?**] algorithm, named K-Gru in this work. Similar as K-Prototype, K-Gru randomly selects the

Algorithm 1 User Clustering in GruBa

```
Input: *Feature Data* $F$ of a set of users, 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**
    perform K-Gru over F to get t clusters
    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 $d_o(u)$;
        Silhouette 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** $R$.
```

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 **??**), of which the data type regarding each dimension is shown as Table **??**.

As aforementioned, the clustering of K-Gru rests on the distance between vector samples, where the dimensions are combined with numbers (normalized to 0-1 range), categories and normalized vectors. For simplicity, we shall first illustrate the distance calculation of the simple vectors with a single data type on each dimension, and then demonstrate that of complex vectors.

For numerical vectors $Y' \ = (y'_0, y'_1, \ldots)$ and $Z' \ = (z'_0, z'_1, \ldots)$, the Euclidean distance [**?**] between Y' and Z' is given by :

$$D_n(Y', Z') = \sum_e (y_e - z_e)^2. \tag{15}$$

For categorical vectors $Y'' \ = (y''_0, y''_1, \ldots)$ and $Z'' \ = (z''_0, z''_1, \ldots)$, the Hamiltonian distance [**?**] of Y'' and Z'' is:

$$D_h(Y'', Z'') = \sum_e H_e, \tag{16}$$

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.

Table 3: Dimension Types in *Feature Data* Vector

Types	**Data Dimensions**
numerical data	$\#R_u$, $\#E_u$, $R_{ee,u}$, $\#B_u$, $R_{oc,u}$, $\#W_{r,u}$,$\#W_{t,u}$
categorical data	$G_u, P_u, U_{t,u}$
normalized vectors	$P_{rt,u}$, $P_{rg,u}$, $P_{tt,u}$, $P_{tg,u}$, P_u

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:

$$D_v(Y^*, Z^*) = 1 - \sum_e Y_e^* \cdot Z_e^*, \tag{17}$$

where $\cdot$ refers to the dot product operation between two normalized vectors Y_e^* and Z_e^*.

Putting these together, the distance regarding the complex vectors $Y = (Y_0, Y_1, ...)$ and $Z = (Z_0, Z_1, ...)$ in K-Gru, referred to as GruBa Distance, is defined as:

$$D_g(Y, Z) = \sum_e G_e, \tag{18}$$

where the distance on each dimension G_e is given by:

$$G_e = \begin{cases} (Y_e - Z_e)^2 & \text{if } Y_e/Z_e \text{ is numerical} \\ H_e \ (1 \ or \ 0) & \text{if } Y_e/Z_e \text{ is categorical} \\ 1 - Y_e \cdot Z_e & \text{if } Y_e/Z_e \text{ is of normalized vector} \end{cases} \tag{19}$$

4.2 Silhouette Coefficient Metric Computation

In system GruBa, the Silhouette coefficient metric serves as the fundamental criteria for deriving an optimal clustering result. Providing a clustering result, each user is associated with a cluster.

Definition 6. *The in-cluster distance $d_i(u)$ is the average distance to all the other users in the same cluster:*

$$d_i(u) = Avg\{D_g(F_u, F_u'') \mid u'' \in l \wedge u \neq u''\}. \tag{20}$$

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

$$d_o(u) = Min\{d^*(u, l') \mid l' \neq l\}, \tag{21}$$

where d^* is given by:

$$d^*(u, l') = Avg\{D_g(Y_u, Y_u') \mid u' \in l'\}. \tag{22}$$

Definition 8. *The Silhouette coefficient value $v(u)$ is defined as:*

$$v(u) \ = \ \frac{(d_o - d_i)}{max(d_o,\ d_i)}. \tag{23}$$

Intuitively, a good clustering solution should have a bigger d_o and a 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 the Group Modeling module 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.*

$$E \in \begin{cases} positive\ items & if\ f\ \ retweeted\ b \\ negative\ items & if\ f\ \ did\ not\ retweet\ b \end{cases} \tag{24}$$

And the data of item E consists of three parts.

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

(2) **Microblog Info** refers to metrics related to the microblog b. The number that b has been retweeted, commented, liked and the length of $b.M$ (microblog message) are considered. What is more, we consider the correlation between b and recent event, where the latter is expressed as several core words returned by Ring [**?**]. Here we compute *TF-IDF weight* W_f of $b.M$, and get correlation metric C_h of b and event by formula **??**.

(3) **Interaction Info** includes seven correlation metrics: $\#B_u$, $R_{oc,u}$, $\#W_{r,u}$, $\#W_{t,u}$, microblog b versus the user u's *Interest Feature* P_u (a.k.a. long-term/stable interest in this work), and b's timestamp versus the time distribution of u's retweeting behavior $P_{rt,u}$. In addition, we consider u's short-term interest, which is mined from u's recent microblogs (e.g., within 30 days) and calculated by *TF-IDF*, namely W_s. The correlation between microblog and u's short-term interest is computed by W_f and W_s (using formula **??**).

The modeling of retweeting behavior of groups is treated as a classification problem, and utilize the random forest classifier to address it. Details for random forest [**?**] are omitted here for space reason. The advantage of this classification model lies in that it could integrate different features conveniently, and the obtained group behavior model could learn what a positive/negative item looks like over each metric mentioned above.

Here we use accuracy to evaluate our model. To define accuracy, we set four variables: E_{tp},E_{fp},E_{tn} and E_{fn}. For a given item E, if E is a positive item and our model also determines it a positive item, then we set E_{tp} to 1, else we set E_{tp} to 0. If E is a negative item and our model determines it a positive item, then we set E_{fp} to 1, else we set E_{fp} to 0. If E is a negative item and our model determines it a negative item, then we set E_{tn} to 1, else we set E_{tn} to 0. If E is a positive item and our model determines it a negative item, then we set E_{fn} to 1, else we set E_{fn} to 0. Formally, accuracy is defined as:

$$accuracy = \frac{\sum_E (E_{tp} + E_{tn})}{\sum_E (E_{tp} + E_{tn} + E_{fp} + E_{fn})}. \tag{25}$$

6 Performance Evaluation

In this section, we shall first detail the experimental setting, and we then present the evaluation 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 Weibo 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, for the Feature Extraction module, for practical reasons, we employed a smaller testing dataset (with manually labeled topics for yardstick) to obtain the proper value of α for extracting *Interest Feature*; For the User Cluster module, we studied the clustering solutions with the minimum/maximum number of clusters 2 and 10; For the Behavior Modeling module, the recent 30 days microblogs of users are used for their short-term interest analysis, and popular words in 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 system GruBa over each component.

Exp-1: Interest Extraction Fig. **??** shows the testing results of using various α values. We report the interest accuracy with various α values. Suppose that the label set manually labeled for each microblog is A, the label set that our method labeled is B. The interest accuracy is defined as $\frac{A \cap B}{A \cup B}$.

System GruBa reaches 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. In general, it performs well when α falls into $[0, 0.8]$.

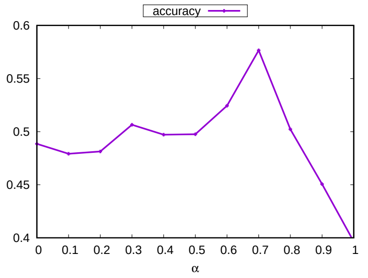


Fig. 2: Interest Extraction: Varying α.

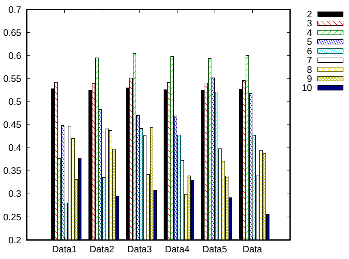
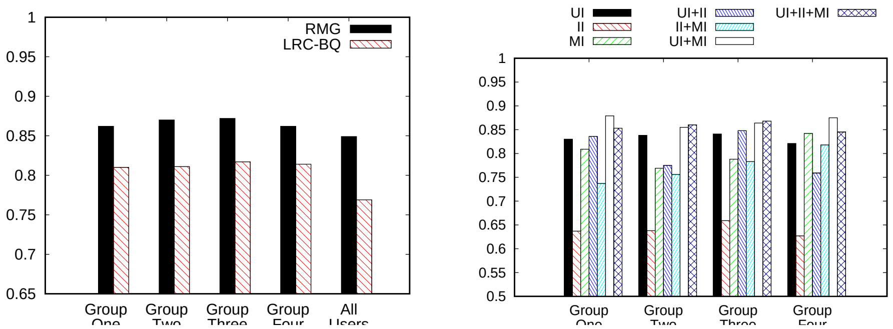


Fig. 3: Silhouette Coefficient Tests.



(a) GruBa Vs. LRC-BQ

(b) Varying Items

Fig. 4: Accuracy Evaluation.

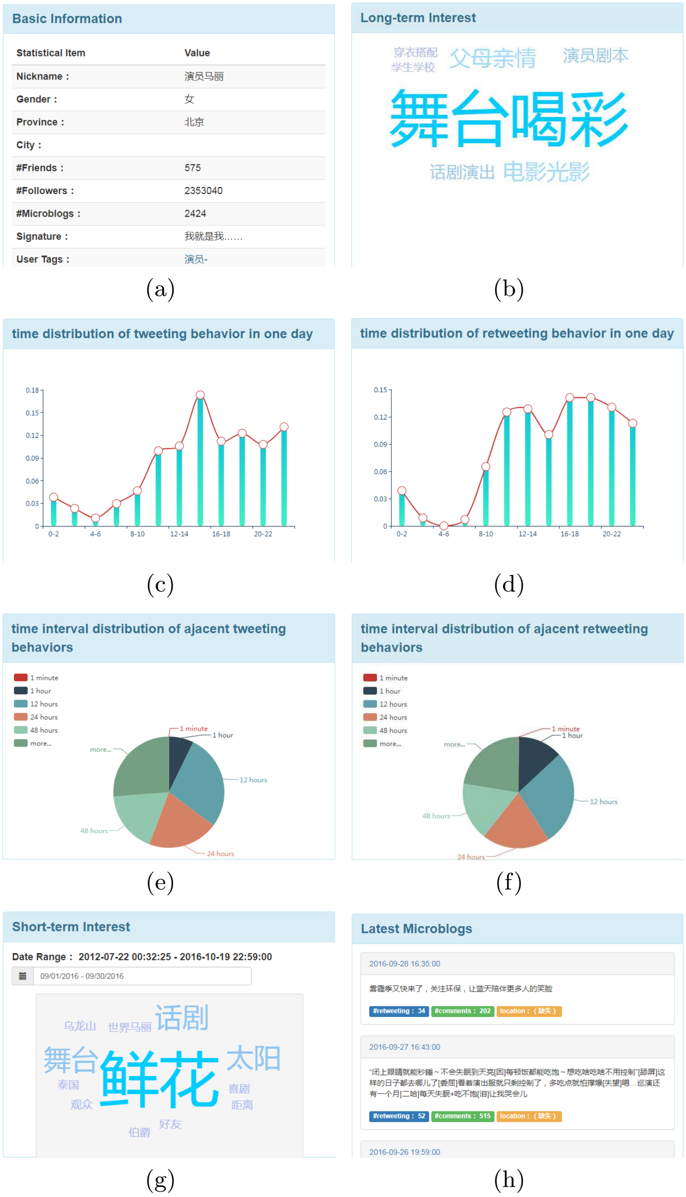
Exp-2: User Clustering Fig. **??** depicts the Silhouette Coefficient Values of multiple clustering solutions, with the cluster number varied from 2 to 10. Specially, we used different testing datasets, with *Data* containing the overall 43.5K users, and each of {*Data1*,..., *Data5*} contains 10K randomly selected users. Except for *Data1*, solutions for {*Data2*,..., *Data5*} are the best for 4 clusters.

Exp-3: Behavior Modeling Fig. **??** shows the performance of GruBa against the state of the art approach LRC-BQ [**?**]. We evaluate the performance using the metrics of accuracy. 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 a single group (i.e., "All-Users"). The results show 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 single group. Ditto for LRC-BQ.

Fig. **??** 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). As shown in Fig. **??**, the default setting wins in most cases.



Statistical Item	Value
Nickname :	演员马丽
Gender :	女
Province :	北京
City :	
#Friends :	575
#Followers :	2353040
#Microblogs :	2424
Signature :	我就是我……
User Tags :	演员-

(a) (b)

(c) (d)

(e) (f)

Date Range： 2012-07-22 00:32:25 - 2016-10-19 22:59:00

09/01/2016 - 09/30/2016

2016-09-28 16:35:00

雾霾季又快来了，关注环保，让蓝天陪伴更多人的笑脸

#retweeting： 34 #comments： 202 location：（缺失）

2016-09-27 16:43:00

“闭上眼睛就能秒睡～不会失眠到天亮[困]每顿饭都能吃饱～想吃啥吃啥不用控制”[舔屏]这样的日子都去哪儿了[委屈]看着演出服就只剩控制了，多吃点就怕撑爆[失望]嗯…巡演还有一个月[二哈]每天失眠+吃不饱[泪]让我哭会儿

#retweeting： 52 #comments： 515 location：（缺失）

2016-09-26 19:59:00

(g) (h)

Fig. 5: Visualization for User Features.

Exp-4: Case Study for Feature Extraction In this test, 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 actress in China) as an example.

The feature extraction result for Mary is depicted in Fig. **??**. Then we can see the basic information of Mary in Fig. **??**: her nickname is Actress Mary (演员马

丽) and she is from Beijing (北京). Mary has more followers than followees. The Sina Microblog tag she made for herself is “actress” (演员). As to the long-term interest, she is interested with stage performance (舞台表演), drama (话剧表演),film (电影光影) and so on as shown in Fig. **??**, which is consistent with her tag. The probability distribution of tweeting and retweeting indicates that she is more active at night than daytime as shown in Fig. **??** and **??**. According to Fig. **??** and **??**, the interval between her two tweeted/retweeted microblogs is mostly within 48 hours, showing she is an active user. Mary’s short-term interest, e.g. from 09/01/2016 to 09/30/2016, is shown in Fig. **??**, which indicates she had been busy with promoting the drama “Earl of Oolong Mountain” (乌龙山伯爵). So the results are in line with expectations, as drama (话剧) and stage (舞台) in Fig. **??**.

To conclude, by modeling user behaviors over groups instead of a single model for all users such as LRC-BQ, we improve the average accuracy over LRC-BQ by 6%. What deserves to be mentioned is that the performance of LRC-BQ is also improved significantly by user clustering.

7 Related Work

In this section, we review related work in literature 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.

For feature analysis, there has been existing work of mining user features, such as race [**?**], gender [**?**], age [**?**], political preference [**?**,**?**] and occupation [**?**]. Our work, however, does not focus on the mining process, but uses the mined features as the input for user clustering and group based behavior modeling.

Studies of behavior analysis put emphasis on exploring the characteristics. For example, [**?**] proposed a model that can properly explain various time distributions of user behaviors by theoretical analysis; [**?**] studied the user activity distribution of one day/week; [**?**] 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. [**?**] mined the user interests by exploring keywords of microblogs with the aid of word frequency and machine translation. [**?**] proposed a method of extending the topic model to obtain use interests. Also, [**?**] used a knowledge base and [**?**] provided a solution of using hashtag for interest analysis. [**?**] summarized user interest by exploring the mentioned celebrities; Similarly, [**?**] leveraged the followed experts to result interest characteristics. Different from these existing solutions, GruBa employs a cell lexicon to properly express user interests, in which Twitter-LDA [**?**] and TF-IDF are employed.

Approaches of grouping users in social networks could fall into a variety of categories. [**?**] grouped users by the info of race, political view and etc. [**?**] studied the social groups on Facebook and Wechat, resulting various patterns of group evolution. [?] 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. [?] employed matrix decomposition to mine user community; [?] and [?] considered followees info for user clustering; [?] proposed an incremental algorithm to mine user community using modular degree as the clustering yardstick. Providing the diversity of user features, GruBa employs basic, behavior and interest features 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. [?,?] discovered that users' retweeting behavior is largely influenced by their followees, whereas [?] employed matrix decomposition, [?] used collaborative filtering methods and [?] leveraged statistical models for retweeting analysis. [?] and [?] 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 a single model for all users or one model per user.

8 Conclusions

In this work, we have presented GruBa, a system to model the retweeting behavior of users in social media. GruBa departures from existing work by grouping users into clusters, during which features of basics, behavior and interests are extracted. Specially, we have studied interest features from various perspectives, such as long-term/short-term interests and explicit/implicit interests. Finally, we have provided a performance evaluation of GruBa by using real-world datasets to demonstrate the benefits of our system GruBa.