Abstract—Study of human behavior patterns is of utmost importance to many areas, such as disease spread, resource allocation, and emergency response. Because of its widespread availability and use, online social networks (OSNs) have become an attractive proxy for studying human behaviors. One of the interesting and challenging problems about OSNs is that how much attention of a post from a user can gain? In this paper, we try to tackle this issue by exploring approaches to predict the amount of reposts any given post will obtain in Sina Weibo, a famous microblogging service in China. Specifically, we propose a RepostsTree based method to model the reposting process in a temporal dynamic manner. Experiments over the real world collected data indicate that our method is effective on repost predicting.

Keywords-Social Media; Sina Weibo; Reposts predicting

I. INTRODUCTION

With the development of the Internet, the place for expressing oneself and contacting peers is transferring rapidly from offline to online. Online social networks are increasingly becoming popular places to exchange information and meet like-minded people. In popular social networking sites like Google+ and Twitter, people share activity updates with their neighbors or followers, while the neighbors will disseminate the received updates with others. In this way, content generated by a user propagates through the network to a wide user population. Because of its usefulness, and wide-accessibility, social network is rapidly changing the public discourse of human society.

Content-centric social networks, such as Twitter and Sina Weibo, promote content spread by allowing users to connect with people having common interests, who may not be their friends. The extent to which a social network spreads content is a key metric that impacts both user engagement and network revenues. From the social network’s perspective, higher content spread can promote user engagement and further improves user retention and audience growth. Furthermore, it provides more opportunities for monetizing the content via online ads, sale of virtual goods, and so on. As a result of the above benefits, it is of importance for researchers to investigate the dissemination of content in social networks.

One of the significant challenges under this trend is that can we predict how much attention of an online post will ultimately gain. Researchers have found that user attention is allocated in a rather asymmetric way, with most posts getting few views, re-propagates, or downloads, whereas some ones receive the most attention [21]. While it is possible to predict the distribution of attention over many items, it is notably difficult to predict the amount that will be devoted over time to any given item.

Twitter is the most influential social media platform around the world, with a large number of users and a huge amount of content distribution every day. Sina Weibo is another remarkable microblogging service provider, although it was founded in 2006, much younger than Twitter, but the number of users, content and traffic rank make it the most popular microblogging site in China, and many special features in Sina Weibo differs from Twitter [17]. However, unlike Twitter which has been well studied on, such as its potential applications and information propagation mechanisms, much less studies have been done on Sina Weibo. Recent researchers mainly focus on characterizing the web structure of Sina Weibo, and make comparison between Weibo and Twitter. Very few studies investigate the content spread in Sina Weibo so as to predict the information diffusion in Weibo. Like Twitter, Sina Weibo also has a retweet-like mechanism so called repost for content propagation.

In this paper we explore the approaches to model and predict reposting patterns in Sina Weibo. We are motivated to investigate the feasibility of building a repost model with topic free measures. We are interested in not only general content features such as URL, mentions, etc., but also some hidden features like the modality of the media types in content of a microblog, and the unique contextual features like VerifiedOrNot (See Section 4.2) in Sina Weibo. We built a tree-based predictive model, taking consideration of the users’ relationship and reposting behavior patterns in Weibo.

The rest of this paper is organized as follows. First, we discuss prior work on microblogging-like social networks. Next, in Section 3 we documented in details on our prediction model. We then introduce our data sets used in our experiment, and describe our analysis on impacting features about repostability, followed by illustrating experiment results on both linear regression model and tree-based model. We conclude our work in Section 5.

1 http://en.wikipedia.org/wiki/Sina_Weibo
II. RELATED WORK

Social networks, on an unprecedented scale, serve as a fertile ground for examining social interactions [2]. Microblogging online social networks focus on sharing information, and as such, have been studied extensively in the context of information diffusion. For example, diffusion and influence have been modeled in blogs [6, 7, emails] [8], and sites such as Twitter [9]. Leskovec et al. [10] studied the explicit graph of product recommendations, Sun et al. [11] studied cascading in page fanning, and Bakshy et al. [12] examined the exchange of user-created content.

For Twitter, there have been many studies on retweeting that try to analyze retweeting behaviors and related factors. For retweeting behaviors, various motivations are explored in [22], while the propagation graph and statistics are studied in [20] and [23]. For retweeting-related factors, [24] and [1] find that retweeted and normal tweets are different in dimensions such as the inclusion of URLs and hashtags, publish time, wording, author publicity, and even the URL shortening service used.

There have been some researchers that try to build a model to predict the retweeting decisions of a targeted network. The authors of [25] address such problem by means of constrained optimization using factor graphs in a generative manner. In [26], the authors try to solve the same problem using conditional random fields, and in order to improve the prediction runtime performance, they investigate approaches to partition the social graph and construct the network relations for retweet prediction. Yet, such fine grained predictive model for retweeting may not be well adopted in practice, since it is difficult to aware of the targeted users the content will arrive. Unlike the understanding or characterizing work on the retweet mechanism of Twitter, few studies have been done to predict the number of retweets that will be obtained for a given tweet. One study toward this problem is [1], which build s a regressor to predict the aggregate number of retweets for a given tweet. While, since the number of followers for a Twitter user ranges from none to millions, using such an aggregate prediction is very difficult to estimate the information spread.

Up to now, few studies on Sina Weibo have been conducted. New features in Sina Weibo differ from Twitter have been analyzed [17, 19]. Qu et al. [15] focused on empirical descriptions of the spreading pattern of disaster-related information in Sina Weibo systems. Chen et al. [16] made a comparison between Weibo and Twitter in terms of web structure, and some researchers conducted such comparisons from cultural perspective [13, 14]. Another interesting study on Sina Weibo was conducted by Sandes et al. [3], which proposed a logical relationship model and explored the optimizing of queries in Weibo. Zhang et al. [4] did some initial work on predicting retweet behavior in Sina Weibo, yet their study is limited on focusing the correlation of number of followers and retweet probability.

Being able to explain and predict online human activity (behavior), much work has been done with statistical tools [27]. And Barabási [28] reported that deliberate human activity is inherently non-Poissonian. The authors in [29] give a Poissonian explanation for the heavy tails in e-mail communication. Similar to our approach, in [30], the authors give a logarithm linear model based on the Gaussian random variables to model and predict the dynamics of the popularity growth of online content. While, the former researchers mainly detected retweets (retweets in Twitter) associated with the original microblogs based on the textual analysis (e.g., check the “RT”, “/@” etc.) approach, which may suffer from the free text style of reposted microblogs [24]. We instead leverage the real world reposting lists data of given microblogs in Sina Weibo and build a tree-based model to generate an dividing rule for the online time serials of reposting behaviors.

III. TREE-BASED PREDICTION MODEL

To gain some intuition about reposting activity pattern in Sina Weibo, let us consider a fictitious user (termed as $A$) who published a microblog $M$. After $M$ is posted, a subset (who happen to be on the Sina Weibo within a certain time window after $M$ is posted) of the followers of $A$, have approximately the equal probability to view $M$. Of course, the exact time each follower browses the post differs, and the latter the follower view the post, the less opportunity the post will be viewed, hence, the less chance to be reposted.

The process that whether user $A$ published microblog $M$ and whether her followers reposted $M$ within a certain time duration can be approximately considered as a single Poisson process [5]. After $M$ is reposted, the reposters’ followers who may or may not be following $A$, also have similar possibility to view $M$ and re-post it. Again, this can be considered as another single Poisson process which may parameterly differs from the former Poisson process. Thus, the whole reposting process is a compound Poisson process which consists of one or more individual Poisson processes.

Based on our hypothesis, we propose a RepostsTree based method to model the reposting process in a dynamic manner. First, we explain the details on how to construct such a RepostsTree, and an illustrative example is given; then, we formulate the RepostsTree based prediction model.

A. RepostsTree Construction

We propose the RepostsTree model, which describes the reposting temporal dynamics in a hierarchical fashion. Original microblogs are represented as the root node, and the boosting reposts are placed as the children or grandchildren of the root. The boosting reposts are the ones with high contribution value. Here we define contribution as the property that whether or not a specific repost can fuel the reposting process, in other words, the higher the contribution value is, the more reposts will be gained by the original post. Of course, the exact time each follower browses the post differs, and the latter the follower view the post, the less opportunity the post will be viewed, hence, the less chance to be reposted.

The process that whether user $A$ published microblog $M$ and whether her followers reposted $M$ within a certain time duration can be approximately considered as a single Poisson process [5]. After $M$ is reposted, the reposters’ followers who may or may not be following $A$, also have similar possibility to view $M$ and re-post it. Again, this can be considered as another single Poisson process which may parameterly differs from the former Poisson process. Thus, the whole reposting process is a compound Poisson process which consists of one or more individual Poisson processes.

Based on our hypothesis, we propose a RepostsTree based method to model the reposting process in a dynamic manner. First, we explain the details on how to construct such a RepostsTree, and an illustrative example is given; then, we formulate the RepostsTree based prediction model.

A. RepostsTree Construction

We propose the RepostsTree model, which describes the reposting temporal dynamics in a hierarchical fashion. The boosting reposts are placed as the children or grandchildren of the root. The boosting reposts are the ones with high contribution value. Here we define contribution as the property that whether or not a specific repost can fuel the reposting process, in other words, the higher the contribution value is, the more reposts will be gained by the original post as a whole. To simplify our method, we use followers count as the contribution measurement, and determine this value in both relative and absolute ways. In addition, the reposts which cannot be considered as boosting reposts are treated as the members of the time serials attached to those boosting reposts.

We propose the RepostsTree model, which describes the reposting temporal dynamics in a hierarchical fashion.
Original microblogs are represented as the root node, and the boosting reposts are placed as the children or grandchildren of the root. The boosting reposts are the ones with high contribution value. Here we define contribution as the property that whether or not a specific repost can fuel the reposting process, in other words, the higher the contribution value is, the more reposts will be gained by the original post as a whole. To simplify our method, we use followers count as the contribution measurement, and determine this value in both relative and absolute ways. In addition, the reposts which cannot be considered as boosting reposts are treated as the members of the time serials attached to those boosting reposts.

The structure of RepostTree is based on the follower/followee relationship of the authors who published the posts and/or reposts. However, since it is common for users to add or delete his/her followees in Sina Weibo, and due to the strict policy on obtaining followers’ IDs of a specific user via open API in Sina Weibo, it is difficult to obtain the complete figure of follower/followee relationship of the users who are involved in a certain reposting process. Thus, to relieve negative effect brought by this situation, we provide a remedy in which the reposts are collected by “orphan collector”. The “orphan collector” collect those reposts which are published by the users who did not follow (may be he/she followed, but we did not recognize it) any “previous” users who had published the boosting reposts to the same microblog before them.

The detailed steps on constructing RepostTree are illustrated in Fig.1. Where O is the original microblog, and RepostsList is the repost-list collected via Sina Weibo open API (note that, whether or not RepostsList’s empty is relative, since that when the delay time between N and O is longer than the window of our training set, we also considered the RepostsList is empty, although it is not really empty). R is the root node of the first tree; here, we note that the RepostTree is actually a forest rather than tree, since there may be more than one root in the RepostTree.

**Procedure RepostTree Constructor**(O, RepostsList)

1. O → R
2. While RepostsList is not empty
   1. Pop the first repost R from RepostsList
   2. Search the latest tree node L which is followed by R
   3. IF L exists, then go to 2
   4. ELSE go to 3
3. IF N is boosting repost
   1. Add N as a child of L into the tree
   2. Add N into the reposts time line serials owned by L
4. ELSE “Orphan collector” add N to its own reposts time serials

Figure 1. The procedure constructs the RepostTree based on user relationship

B. An Example

To illustrate our RepostTree construction process, an example is given as follows (Fig.2).

At the beginning, the original microblog (denoted as R, and its author as A) is placed as the root node of the RepostTree, and the “orphan collector” is empty.

Firstly, we scan the repost-list of R, and pop the first repost (denoted as N) from the list. And, perhaps the user who published N is one of the followers of A, and its contribution value satisfies the absolute condition (e.g., the author of N has more than 10,000 followers), thus N is placed as the child node of R (shown as Fig.2.a).

Next, the second repost (switch the symbol N to it, the same below) is popped from the list, and again its author is one of followers of A, but N doesn’t satisfy the absolute condition, neither the relative condition (e.g., the number of followers recorded for the author of N is more than A’s, and we call the absolute and relative conditions as the privilege conditions). Hence, N is placed as one of the members of the sub time serials attached to R (Fig.2.b).

Then, some reposts are popped, and none of them satisfy the privilege conditions, so they are appended to the sub time serials of the nodes (according to the follower/followee relationships). And Fig.2.c depicts the situation that a repost N satisfies the relative condition of privilege conditions (i.e., its author has more followers than A’s).

When the author of a repost N did not follow anyone who published the boosting microblog/reposts before him/her, and N does not satisfy the privilege conditions, then it will be placed in the “orphan collector” (Fig.2.d). After that, a repost like the N in Fig.2.d is popped, but it does satisfy the privilege conditions, then place it as root node of another tree (Fig.2.e). Finally, the RepostTree is constructed as shown in Fig 2.f.

---

C. The prediction Model

Let $X$ be the time serials of repost-list where each $x_i$ represents a time delay between original microblog and the repost $i$. Then, with the RepostsTree constructed, we can divide $X$ into several sub-time serials, so

$$X = \bigcup X_i$$

The dividing rule is based on the RepostsTree, where each node $D$ and its direct children consist a subset. It includes the time serials attached to $D$, and the time delays between the direct children of $D$ and original microblog. Then, given the dividing of $X$, we make the estimation on each subset $X_i$ with maximum likelihood estimation to calculate parameter $\lambda$ so that the single Poisson distribution can be obtained. Finally, we get the compound Poisson distributions set $P = \{P_1, P_2, \ldots, P_n\}$, where $n$ is decided by the above dividing. Given distribution set $P$ after original microblog had been published for time duration $t_0$, we can get the predicted reposts count for time duration $t_0 + \Delta t$ by calculating the sum of the predicted value of each $P_i$ for time duration $t_0 + \Delta t$.

Note that, our model works for predicting in a dynamic way, including the construction of RepostsTree and the dividing of the time serials.

IV. EXPERIMENTAL RESULTS

A. Data Sets

Two data sets were collected for analysis. First, we collected 13 million microblogs with Sina Weibo open API\(^3\). And, we select 142K microblogs (142K dataset) to perform an exploratory data analysis using Principal Components Analysis (PCA) and Linear Regression Modeling.

In addition to the 142K dataset, we also selected 294 popular microblogs, and collected the full repost-list for each microblog we selected (294 repost-lists data set), and such data set was used to perform tree-based modeling and predicting.

1) The 142K Data Set

The purpose of our exploratory data analysis is to understand the features related to repostability. Conceptually, one would like to focus on a set of microblogs, use their features as independent variables, and treat the reposts count as the dependent variable to be predicted from the samples.

For our exploratory analysis, we arbitrarily chose a date and collected microblogs since that date (12,928,673 microblogs published by about 100K users since January 1, 2012). Reposts were filtered from this sample to yield 888,888,9 original microblogs. And, we randomly selected 142,498 microblogs which are posted for more than one day. Out of the 142K microblogs, 117,498 microblogs are reposted more than once and 130 of them are reposted over 10,000 times.

2) The 294 Repost-Lists Data Set

<table>
<thead>
<tr>
<th>HasPicOrNot</th>
<th>Whether or not include picture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modality</td>
<td>Number of media types included</td>
</tr>
<tr>
<td>MaxMediaWeight</td>
<td>Max media weight value</td>
</tr>
<tr>
<td>Mention</td>
<td>Sum of followers count of the users mentioned in a microblog</td>
</tr>
<tr>
<td>URL</td>
<td>Maximum value of the numeral properties related to the URLs included in a microblog</td>
</tr>
<tr>
<td></td>
<td>(numeral properties related with a URL include clicks count, share count and comments count)</td>
</tr>
<tr>
<td>Followers</td>
<td>Number of users who follow the author of a microblog</td>
</tr>
<tr>
<td>Followeres</td>
<td>Number of friends that the author is following</td>
</tr>
<tr>
<td>Statuses</td>
<td>Number of microblogs generated by the author since the creation of the account</td>
</tr>
<tr>
<td>Favorites</td>
<td>Number of favorited microblogs by a user</td>
</tr>
<tr>
<td>Activity</td>
<td>Activity level of the author</td>
</tr>
<tr>
<td>VerifedOrNot</td>
<td>Whether or not the author is verified</td>
</tr>
<tr>
<td>Days</td>
<td>Number of days since the author created the Sina Weibo account</td>
</tr>
<tr>
<td>Reposts</td>
<td>Number of reposts recorded for a given microblog</td>
</tr>
</tbody>
</table>

After the repost-lists of the 294 original microblogs downloaded, we found that 41 out of 294 repost-lists contain less than 10K reposts, which may result from the deletion of the original microblog or occasional unavailability of Weibo open platform. Then, we filtered those lists including less than 10K reposts. This yielded a set of 253 repost-lists.

B. Data Analytics and Feature Extraction

For the data sets collected above (142K data set), we extracted a set of features which are language independent. We divided these features into two sets: the first is concerned with the content features (HasPicOrNot, Modality, URL, Mention, MaxMediaWeight), and the second set is mainly related to the authors (Followers, Followeres, Favorites, Statuses, Activity, VerifedOrNot, Days). The features are defined in TABLE I.

The Modality feature in TABLE I aims to capture the variey of media types, like if the content of a microblog contains the image, audio and video, then its Modality is 3. The extraction of this feature is implemented by extracting URLs from the content, and detecting the URL type via Sina Weibo open API\(^4\). In Sina Weibo, the URL has five types: ordinary web page, video, music, activity and voting, and each type has related value, such as ordinary web page assigned to 0, video assigned to 1, music assigned to 2, etc. According to our observation, the voting and activity URL type are more attractive than video and music, so we arbitrarily assign weight to the media types (URL types), voting and activity is assigned to 3 respectively, video and music is assigned to 2 respectively, and ordinary web page is assigned to 0. So, the MaxMediaWeight feature is the maximum value of weight assigned to the media type.

---

\(^3\) http://open.weibo.com/wiki/2/statuses/user_timeline

\(^4\) http://open.weibo.com/wiki/2/short_url/info.html
In order to obtain the accurate reposts count, we select the 142K data set by eliminating the microblogs had been published for less than 24 hours.

1) Pearson Correlation Analysis

We first check the relevance between extracted features and reposts count by using Pearson correlation coefficient. The 142K data set was used in this experiment. The result is shown in Fig. 3.

We can observe that the reposts count is strongly associated with MaxMediaWeight of content features and Followers of context features. Interestingly, the features of Followees and VerifiedOrNot are marginally negatively associated with reposting.

However, as we can see that, Pearson correlation coefficients of all features are less than 0.2, which suggest that there is little linear correlation between the features and the repostability of given content, and further underlying factors should be unsealed.

2) PCA analysis

PCA (Principal Component Analysis) is widely used in an attempt to reveal underlying structure that maximally accounts for the variance in the data set. We used the varimax rotation technique to produce orthogonal factors. TABLE II presents the factors (principal components) extracted by PCA.

There are many criteria for determining the proper number of factors about data variance analysis, such as Kaiser Criterion, Variance explained criteria, and Scree plot test etc [31]. Each criterion suggests retaining at least 5 factors (the first 5 factors). It indicates that our original extracted features can be well adopted to be the independent variables for explaining the repostability of a microblog.

C. Results of the Linear Regression Model

Linear Regression is a traditional approach to model the relationship between a scalar dependent variable \( y \) and one or more explanatory variables denoted \( X \). Here, in our scenario, \( y \) is the Reposts features, and \( X \) is the first twelve features in TABLE I. We used the least-squares approach to fit out linear regression model on the 142K data set. The results are shown in TABLE III (with the confidence 1 - \( \alpha \) = 0.99).

The results in TABLE III correspond to the findings that the author feature of the followers has significant effects on repost probability. The content features of Mention and URL have significant effects on repostability.

The \( p \) value is less than \( \alpha \), suggesting that the linear regression model holds. The statistical \( s^2 \) is really high, which indicates that exogenous influence plays an important role in reposting of Sina Weibo.

D. Results of Tree-based model

As mentioned in Section 3.1, our RepostsTree is based on the user relationship of ones who participated in the
reposting of certain microblog in Sina Weibo. In order to make our relationship map mostly cover the ties of follower/followee relationship, we collect the followees’ IDs of the users who are involved in the reposting process. As Sina Weibo open API allows to obtain at most 5,000 followees’ user IDs, and it is rarely for users to follow more than 5,000 users in microblogging service sites as we observed. Hence, we collect the follower/followee relationship from the following perspective (that is we collect the followees’ IDs of certain users), and the symmetry of the follower/followee relationship in microblogging system make our approach feasible.

In order to simulate the real situation where the reposts accumulated over time, we recursively carry out our experiments. For instance, after the microblog $M$ had been published for one hour, we use our tree-based model to get the compound Poisson distribution set $P$ (see Section 3.2) based on the repost-list for $M$, and calculate the predicted reposts count for, like two hours, and compare the predicted value to the real reposts count. Then, we repeat such procedures to compare the dynamics of predicted reposts with the real data we collected.

According to Kwak et al. [20], retweeting probability in Twitter drops rapidly with the increasing of time, and more than 50% of retweets take place within one hour. Here we show our reposting prediction result within 72 hours. Note that, our approach is not limited within 72 hours.

Firstly, we test our tree-based prediction model on 253 repost-lists in 2 hours, 10 hours and 24 hours after original microblog published respectively. Here, the learning step is set to 1 hour, that is setting 1 hour to $\Delta t$ (see Section 3.2). The predictive error rates are illustrated in Fig.4. Note that, the error rate on case 139 (WeiboID = 3492391866790186) fluctuate severely. It is because that there is too much missing data in the repost-list of this microblog.

After eliminating the low quality sample, we present the prediction error rate in Fig. 5. We can see that, as the time elapses, the predictive error rate drops, which suggest that our approach can learn iteratively, and the more empirical knowledge it learned, the better precision it will obtain.

So far, our approach only demonstrate its predictive precision on the popular content (with higher reposts count), we also need to investigate if our approach works well over less-popular contents.

We test 182 unpopular microblogs with reposts recorded less than 10,000, the result is shown in Fig.6. The precision on the unpopular content is also acceptable although it is not well performed as the popular ones.

---

5 http://open.weibo.com/wiki/2/friendships/friends/ids
E. Case Analysis

We randomly select a specific microblog (Weibo ID = 3467177816562372) to evaluate the effectiveness of the tree-based model on predicting the reposting dynamics.

First, we use the time serials data within first hour as the training set, and the next hour as the test set. Then, we add the second hour data into the training set, and third hour as the test set, and so on. So, our learning approach works in an iterative way.

From Fig.7, we can see that our model can commendably predict the reposting dynamics for a given content. The number of reposts count estimated is more than real data in the later phase of the reposting. This is because that, the time delay between consecutive reposts became much longer with the time elapsed, hence the heavy tails dominate the Poisson process (see Section 4.6). In Fig.7, we also set our learning step as one hour.

F. Why Poisson works?

As we mentioned in Section 2, online human activities follow non-Poisson statistics, which termed as heavy tailed distribution. A Poisson distribution decreases exponentially, forcing the consecutive events to follow each other at relatively regular time intervals and forbidding very long waiting times. In contrast, the slowly decaying heavy-tailed processes allow for very long periods of inactivity that separate bursts of intensive activity [28].

We conduct experiments to investigate the inter-event distributions of reposting behaviors in Sina Weibo (as shown in Fig.8). Indeed, the time delays between consecutive reposts follow approximately a power law, which is one of typical characteristics of heavy-tailed distributions. However, as Fig.8 shows, among most of the reposting cases, the heavy tails intensively appeared in the final phase, which differs from other online human behaviors [28], and that is why our approach can successfully work. We also conduct experiments over unpopular posts, where such phenomenon is not so obvious as the popular ones.

V. Conclusion

In this paper, we examined the factors that affects the repostability of contents in Sina Weibo using Pearson Correlation Analysis, PCA, and Linear Regression. Generally, we find that in addition to intrinsic features such as content and context, exogenous elements which the traditional correlation analysis approaches fail to capture, also play important roles on users’ reposting behavior in microblogging sites. We proposed a dynamic approach to predict users’ reposting patterns, taking consideration of user follower/followee relationships. Experimental results validate the effectiveness of our model. There are still limitations that need to be studied further. For example, our proposed predictive method works in an iterative manner, which may result in some loss of performance.

We plan to extend our work to develop our predictive approach with model-free methods which make no assumptions about the patterns of users’ reposting behaviors and structure of the underlying statistical process.

ACKNOWLEDGMENT

This work was partially supported by the National Basic Research Program of China (No. 2012CB316400), the National Natural Science Foundation of China (No. 61222209, 61103063), and the Specialized Research Fund for the Doctoral Program of Higher Education (No. 20126102110043).

REFERENCES


[31] https://en.wikipedia.org/wiki/Principal_component_analysis