Towards Effective Recommendation of Social Data across Social Networking Sites

Yuan Wang[1], Jie Zhang[2], and Julita Vassileva[1]

[1] Department of Computer Science, University of Saskatchewan, Canada
{yuw193,jiv}@cs.usask.ca
[2] School of Computer Engineering, Nanyang Technological University, Singapore
zhangj@ntu.edu.sg

Abstract. Users of Social Networking Sites (SNSs) like Facebook, MySpace, LinkedIn, or Twitter, are often overwhelmed by the huge amount of social data (friends' updates and other activities). We propose using machine learning techniques to learn preferences of users and generate personalized recommendations. We apply four different machine learning techniques on previously rated activities and friends to generate personalized recommendations for activities that may be interesting to each user. We also use different non-textual and textual features to represent activities. The evaluation results show that good performance can be achieved when both non-textual and textual features are used, thus helping users deal with cognitive overload.

1 Introduction

Social Networking Sites (SNSs) have changed how people communicate: nowadays, people prefercommunication via SNSs over emails [1]. With the explosion of SNSs, it is also common that a user may engage with multiple SNSs. These users of multiple SNSs see a great number of status updates and other kinds of social data generated by their network friends everyday. This causes a significant information overload to users. One way to deal with information overload is by providing recommendations for interesting social activities, which allows the user to focus her attention more effectively.

In this paper, we present an approach for recommending social data in a dashboard application called "SocConnect", developed in our previous work and described in [2], that integrates social data from different SNSs (e.g. Facebook, Twitter), and also allows users to rate friends and/or their activities as favourite, neutral or disliked. We compare several machine learning techniques that can be used to learn users' preferences of activities to provide personalized recommendations of activities that are interesting to them. In the machine learning process, we use several different non-textual and textual features to represent social activities. The evaluation results show that some of the machine learning techniques achieve good performance (above 80% of correct recommendations on average). Both non-textual and textual features should be used for representing activities.

D. Dicheva and D. Dochev (Eds.): AIMSA 2010, LNAI 6304, pp. 61–70, 2010.

2 Related Work

2.1 Recommender Systems

There is a lot of research in the area of recommender systems dating back from the mid 1990ies. There are two main types of recommender systems: content-based (or feature-based) and collaborative (social). Content-based recommenders analyze features of the content in the set and match them to features of the user (e.g. preferences, interests), based on a user model developed by analyzing the previous actions of the user. Collaborative or social recommenders work by statistically correlating users based on their previous choices. Based on the assumption that people who have behaved similarly in the past will continue to do so, these recommenders suggest content, rated highly by a user, to similar users who have not seen the content yet. Collaborative (social) recommender systems are widely used to recommend movies, books, or other shopping items in e-commerce sites. More recently, recommender systems have been applied in SNSs, but there are still relatively few academic works in this area. SoNARS [3] recommends Facebook groups. It takes a hybrid approach, combining results from collaborative filtering and content-based algorithms. Dave Briccetti developed a Twitter desktop client application called TalkingPuffin (talkingpuffin.org). It allows users to remove "noise" (uninteresting updates) by manually muting users, retweets from specific users or certain applications. Many existing SNSs use social network analysis to recommend friends to users. This, however, does not help in dealing with information overload, on the contrary. So our research focuses on recommending status updates. Status update is different from items like movies, books, or shopping goods in two ways: first, the number of status updates arrive in large volumes, and are only relevant for very short time; second, a status update is more personal and aimed at a small audience. Due to these two features, a collaborative recommendation approach is not a good solution: collaborative filtering works well for a large group of similar users and requires previous ratings.

We focus on status updates recommendation that is content-based. It use machine learning techniques to make predictions based on the user's previous choices and generate personalized recommendations.

2.2 Text Recommendation

Our research shares similarity with text recommendation in the field of Information Retrieval and Personal Information Management, since each status update can be considered as one document. Text recommendation usually has four steps [4]: (1) recognizing user interest and document value; (2) representing user interest; (3) identifying other documents of potential interest; and (4) notifying the user - possibly through visualization. Our work follows these four steps.

Vector space is the most common method for modelling document value. A vector space represents a document or documents by the terms occurring in the document with a weight for each term. The weight represents the importance of the term in the given document. The most common two ways to calculate

the weight are Term Frequency (TF) and Term Frequency - Inverse Document Frequency (TF-IDF).

TF is simply counting how many times each term occurs in the given document, defined as:

$$\mathrm{TF}_i = \frac{N_i}{\sum_i N_i} \tag{1}$$

TF-IDF takes into account not only the importance of the term in the given document but also the general importance of the term across all documents, based on the number of documents containing this term. It can be defined as:

$$\text{TF-IDF}_i = \mathrm{TF}_i \times \lg \frac{|A|}{|A_i|} \tag{2}$$

where $|A|$ is the total number of documents, and $|A_i|$ is the number of documents containing the term.

3 Personalized Recommendations in SocConnect

To relieve the information overload, SocConnect provides personalized recommendations of activities to individual users according to a prediction generated using their ratings on previous social data. Thus, our approach is content-based recommendation, rather than collaborative. In this section, we propose a list of potential non-textual and textual features for representing each activity and we present several machine learning techniques that were used to predict users' preferences on activities from the social networks Twitter and Facebook.

3.1 Learning User Preferences on Activities

Users directly express their preferences on activities and friends by using the function of rating activities as "favourite" or "disliked". The users' ratings of their friends are also used in predicting users' interests in activities posted by these friends. Based on the ratings, SocConnect can learn users' preferences and predict whether they will be interested in new similar activities from friends. Machine learning techniques are often used for learning and prediction. SocConnect applies the classic techniques of Decision Trees, Support Vector Machine [5], Bayesian Networks, and Radial Basis Functions [**?**]. In brief, Decision Tree learning is one of the most widely used techniques to produce discrete prediction about whether a user will find an activity interesting. It classifies an instance into multiple categories. Bayesian Belief Networks is a commonly used Bayesian learning technique. The method of Radial Basis Functions belongs to the category of instance-based learning to predict a real-valued function. Support Vector Machines have shown promising performance in binary classification problems. A performance analysis of these techniques (as implemented in Weka) on learning users' preferences on their social network activities will be presented in Section 4.

3.2 Features for Representing Activities

All machine learning techniques listed above require a set of features describing the data. We identify both non-textual and textual features that are potentially useful for learning.

Non-textual Features. Table 1 summarizes a list of relevant non-textual features and some of their possible values. Each activity has an actor (creator). SocConnect allows a user to rate friends as "favourite" or "disliked". Using these two features, we will be able to learn whether a user tends to be always interested in some particular friends' activities or activities from a particular type of friends (i.e. favourite or disliked friends). Each activity has a type. We also take into account the SNS sources of activity, such as Facebook and Twitter, since often users have a particular purpose for which they predominantly use a given SNS, e.g. Facebook for fun, Twitter for work-related updates. From this feature, we can find out whether a user is only interested in activities from particular SNS sources. Different applications used to generate those activities are also useful to consider. For example, if a user's friend plays "MafiaWars" on Facebook but the user does not, the status updates generated from the "MafiaWars" application may be annoying to the user.

Table 1. Non-Textual Features of Activities for Learning

Non-Textual Features	A Set of Possible Values
Actor	actor's SNS account ID
Actor Type	favourite; neutral; disliked
Activity Type	upload album; share link; upload a photo; status upload; use application; upload video; reply; twitter retweet; etc
Source	Facebook; Twitter; etc
Application	foursquare; FarmVille; etc

The above non-textual features of activities can be obtained through the APIs offered by SNSs. In our work, we also consider the textual content of activities, even though many activities, such as video uploads, do not have any textual content. The purpose of having these features is to investigate whether text analysis will contribute to the personalized recommendation of social activities.

Textual Features. In the text analysis part, we first remove the stop words and URL links in each activity. Two vector spaces are then calculated for each activity, one is using TF and another one is using TF-IDF. The reason of using both algorithms is to investigate whether the commonality (IDF value) of terms plays a role in the data mining process in the context of analysis social data.

Having the vector spaces for each activity and given training data containing a set of activities rated by a user as favourite, neutral or disliked, we sum up

the weight values for each term in all the favourite, neutral and disliked activities, respectively. The results are three vectors over the training data, for the favourite, neutral and disliked activity sets respectively. Each vector consists of the total weight of each term in all activities of the corresponding set (either favourite, neutral or disliked activity set). We then calculate the cosine similarity between a vector representing each activity and the three vectors representing the favourite, neutral and disliked activity sets, denoted as S_F, S_N and S_D, respectively. Each of these similarity values can represent a textual feature for activities.

We can also use one combined textual feature C for an activity. Two ways can be used to generate a value for this feature. One way is to use the difference between the two similarity values, $C = S_F - S_D$. Another way is to map the difference into the three interest levels, favourite, neutral and disliked, as follows:

$$C = \begin{cases} \text{favourite} & \text{if } 0.33 < S_F - S_D \leq 1 \\ \text{neutral} & \text{if} -0.33 \leq S_F - S_D \leq 0.33 \\ \text{disliked} & \text{if } -1 \leq S_F - S_D < -0.33 \end{cases} \quad (3)$$

In summary, we can have four potential textual features for representing activities, including S_F, S_N, S_D and the combined one C, as listed in Table 2. Note that the combined feature C can have a continuous value ($S_F - S_D$) or a discrete one (mapped interest levels). Also note that the values of each feature summarized in Table 2 can be calculated based on either TF or TF-IDF. The performance of the different features and the different ways of calculating feature values will be evaluated and compared in Section 4.

Table 2. Textual Features of Activities for Learning

Textual Features	Possible Values
S_F	$\in$ [0, 1]
S_N	$\in$ [0, 1]
S_D	$\in$ [0, 1]
C	$S_F - S_D \in$ [-1, 1]; or Mapped interest levels: $\in$ {favourite, neutral or disliked}

After learning from a user-annotated list of activities from his or her friends, each of which is represented by a set of the feature values, a learning algorithm is able to predict whether a new activity from a friend will be considered as "favourite", "neutral" or "disliked" by the user.

4 Evaluation

We carried out experiments to evaluate 1) the performance of the four machine learning techniques for learning user preferences on social activities and 2) the performance of personalized recommendations when different features are used

to represent social activities. Social data streams from ten subjects were used in the evaluation. Five of the subjects are from Saskatoon, Canada, and the other five are from New Jersey, USA. Half of them are students and the other half are workers. Six of the subjects are experienced users of Facebook and Twitter. For each of these subjects, we collected from Facebook and Twitter 200 recent activities of their friends. The other four subjects are relatively new users of Facebook and Twitter. For each of them, we collected around 100 recent activities of friends. Thus, in total, we collected around 1,600 user activities. We asked all subjects to rate their friends and activities. On average, they rated 38% of their friends as favourite or disliked friends and 45% of the activities as favourite or disliked. Thus, the data sample is quite diverse. A 10-fold cross validation was performed on the collected data from each subject, and the average performances of the machine learning techniques over the activities of all subjects are reported in the following sections.

4.1 Performance When Using only Non-textual Features

We first used only the set of non-textual features summarized in Table 1. Fig. 1 shows the performance of the four machine learning techniques. Although the performance difference among these techniques is not significant, support vector machine (SVM) provides the best performance, and it correctly classifies 69.9% of instances in the testing data. RBF performs the worst (68.4%). The performance of Decision Tree and that of Bayesian Belief Networks are about the same, which is around 69.5%. So, these machine learning techniques generally do not show good performance when only the non-textual features are used for representing activities.

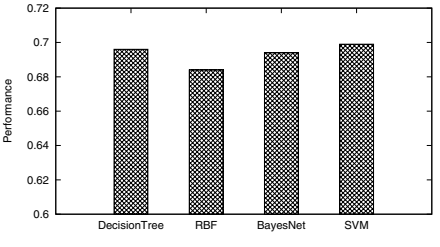

Fig. 1. Performance when only Non-Textual Features are Used

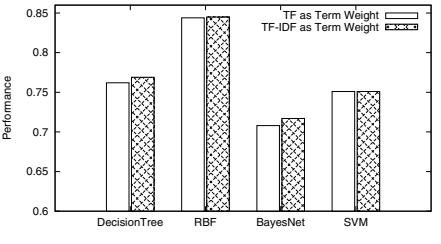

Fig. 2. Performance when Three Textual Features are Used

4.2 Performance When Using Only Textual Features

We then evaluated the performance of personalized recommendations on social activities when only the textual features summarized in Table 2 are used. In this set of experiments, we first tested the performance when the combined feature C is used. All the four machine learning techniques perform the same and achieve 64.9% of correct prediction. In addition, there is no difference when

TF or TF-IDF is used as term weight. Using this feature alone shows even worse performance than using the non-textual features.

We then tested the performance when the other three textual features (S_F, S_N and S_D) are used. The results are plotted in Fig. 2 when TF and TF-IDF are calculated for term weight respectively. We can see that now RBF performs the best (84.5% of correct prediction). RBF is known as generally showing good performance when the values of features are continuous, as it predicts a real-valued function. Decision Tree is the second best and has the performance of 76.9%. SVM is better than Bayesian Belief Network in this case. We can also see that there is still no much performance difference between TF and TF-IDF. From the evaluation results presented in this section, it is also clear that the performance when the three textual features are used is significantly better than that when the combined textual feature C is used and also better than the performance when non-textual features are used.

4.3 Using Both Non-textual and Textual Features

We further evaluated the performance of personalized recommendations on social activities when non-textual and textual features are both taken into account. We first use the combined feature C and the non-textual features. As described in Table 2, four different ways can be used to calculate the value for the feature C of an activity, listed as follows:

- TF+noMap: weight of term is calculated using TF and feature value is calculated by $S_F - S_D$;
- TF+Map: weight of term is calculated using TF and feature value is calculated by mapping $S_F - S_D$ to one of the three interest levels;
- TF-IDF+noMap: weight of term is calculated using TF-IDF and feature value is calculated by $S_F - S_D$;
- TF-IDF+Map: weight of term is calculated using TF-IDF and feature value is calculated by mapping $S_F - S_D$ to interest levels.

The performance of each method is summarized in Table 3. We can see that the methods without mapping to interest levels produce the better performance than those with mapping. There is no much difference between "TF-IDF+noMap" and "TF+noMap" or between "TF-IDF+Map" and "TF+Map". Thus, calculating term weight using TF-IDF does not provide much contribution to the personalized recommendation of social data. The performance when using both the combined feature C and the non-textual features (up to 79.4%) is much better

Table 3. Performance when Using C and Non-Textual Features

Methods	DecTree	RBF	BayesNet	SVM
TF+noMap	0.777	0.793	0.773	0.764
TF+Map	0.712	0.704	0.711	0.716
TF-IDF+noMap	0.780	0.794	0.761	0.749
TF-IDF+Map	0.718	0.698	0.713	0.718

than that using each alone (up to 69.9% with non-textual features and 64.9% with only the combined feature C).

We then use the combination of the three textual features (S_F, S_N and S_D) and the non-textual features. The results are plotted in Fig. 3 when TF and TF-IDF are calculated for term weight respectively. Again, there is no much performance difference between TF and TF-IDF. RBF performs the best (81.4%). Decision Tree and SVM perform similarly (around 80%). Bayesian Belief Network is the worst in this case (around 75.2%).

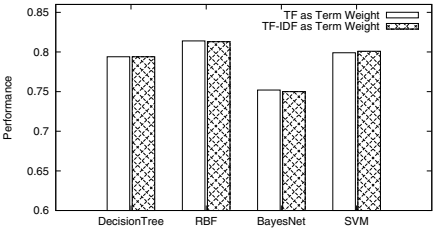

Fig. 3. Using S_F, S_N, S_D and Non-Textual Features

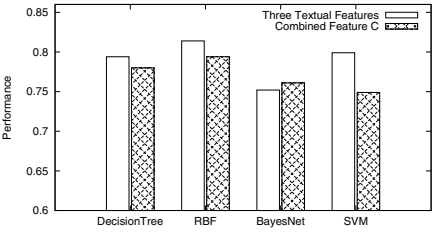

Fig. 4. Performance Comparison between Textual Features

We compare the performance between different textual features when the textual features are integrated with the non-textual features. In this comparison, we choose the best performance of the combined feature C. The result obtained is similar as that when only textual features are used, as shown in Fig. 4. In most of the cases, the three textual features provide better results than the combined feature. Bayesian Belief Network is the exception. The result concludes that it is generally better to use the three features separately instead of combining them.

4.4 More Analysis

To further analyze the obtained evaluation results, we also plot the performance of personalized recommendations when using only non-textual features, when using only textual features of S_F, S_N and S_D, and when using both, respectively in Fig. 5. We can see that in general, the best performance of the machine learning algorithms is produced when both non-textual and textual features are used. Thus, both non-textual and textual features contribute to the personalized recommendations of social activities. Note that RBF is exceptional. Its performance when using both non-textual and textual features is worse than that when using only textual features. Integrating discrete values of non-textual features degrades its performance. We analyzed the evaluation results using two factor ANOVA (analysis of variance) test with replication with 0.05 p-value, and the analysis shows that the difference between the performance of the combined approach and the other two approaches (textual and non-textual) is statistically significant. The ANOVA analysis did not show significant difference in the performance of the four tested machine learning algorithms. The combined text

and non-text features approach yielded significantly better results with all four algorithms.

Using Weka's feature selection function, we can see which features are more important for individual users. We summarize in Fig. 6 the number of subjects for whom each feature was the most important one in the prediction. In this experiment, non-textual features and the three textual features (S_F, S_N and S_D) are used because they produce the best performance for most of the machine learning algorithms.

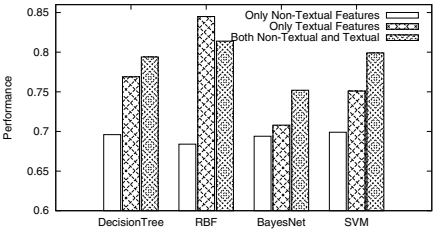

Fig. 5. Performance Comparison for Different Features

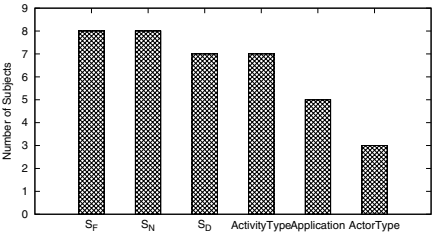

Fig. 6. The Most Important Features for Learning

For most of the users, the three textual features are important. This implies that most of the users are interested in the textual content of their friends' activities. "Activity Type" is also important for most of the users. For half of the users, "Application" is important. "Actor Type" is important for three users. The source of activities (i.e. whether they come from Twitter or Facebook) turns out to be not important. This interesting difference represents the diversity of social networking users' criteria in judging whether an activity is interesting to them, reflected in their ratings. Some users mainly care about the textual content of activities. Some users care about the type of their friends' activities. Some users care more about the applications that generate the activities, which are usually the games they are playing. And, some users care about their close friends' activities. The implication is that learning the user type would be useful in selecting the best suitable set of features for personalized recommendation of activities. We leave this for future work.

4.5 Conclusions from the Evaluation Results

Several important conclusions can be drawn from the evaluation results presented in the previous sections: a) both non-textual and textual features contribute to the personalized recommendation of social activities; the combination of textual and non-textual features performs significantly better than only textual or only non-textual features across all four algorithms; b) the best performance (84.5%) is produced by RBF using only the textual data, indicating that good performance can be achieved for the personalized recommendation of social

activities; c) calculating term weight using TF-IDF does not show much advantage for textual features; and d) learning user types would be useful for further improving the performance of the personalized recommendations of activities.

5 Contribution and Future Work

Our work shows that it is possible to generate effective recommendations of social data using machine learning. Moreover, we found that both textual and non-textual features have to be taken into account and the results then the results would be comparably good for four machine learning algorithms. For future work, we are interested in exploring more deeply the relative importance of different features of SNS activities, to further improve the performance of the personalized recommendation. Other features that may be worth looking at include the targeted friends in activities (e.g in comments, responses, likes). Our immediate next step will be to conduct user studies to evaluate the quality of recommendations from the user point of view.

References

1. Chisari, M.: The future of social networking. In: Proceedings of the W3C Workshop on the Future of Social Networking (2009)
2. Wang, Y., Zhang, J., Vassileva, J.: SocConnect: A user-centric approach for social networking sites integration. In: Proceedings of the International Conference on Intelligent User Interface (IUI)Workshop on User Data Interoperability in the Social Web. (2010)
3. Carmagnola, F., Vernero, F., Grillo, P.: Sonars: A social networks-based algorithm for social recommender systems. In: Proceddings of the 17th International Conference on User Modeling, Adaptation, and Personalization (2009)
4. Claypool, M., Le, P., Waseda, M., Brown, D.: Implicit interest indicators. In: Intelligent User Interfaces, pp. 33–40. ACM Press, New York (2000)
5. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. In: Schoelkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods: Support Vector Learning. MIT Press, Cambridge (1999)
6. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)