



ELSEVIER

Contents lists available at SciVerse ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

SocConnect: A personalized social network aggregator and recommender

Jie Zhang^{a,*}, Yuan Wang^b, Julita Vassileva^b

^a School of Computer Engineering, Nanyang Technological University, Singapore

^b Department of Computer Science, University of Saskatchewan, Canada

ARTICLE INFO

Article history:

Received 30 September 2011

Received in revised form 5 June 2012

Accepted 18 July 2012

Available online 6 October 2012

Keywords:

Social networking site

Social network aggregator

Personalized recommendation of activities

ABSTRACT

Users of Social Networking Sites (SNSs) like Facebook, LinkedIn or Twitter, are facing two problems: (1) it is difficult for them to keep track of their social friendships and friends' social activities scattered across different SNSs; and (2) they are often overwhelmed by the huge amount of social data (friends' updates and other activities). To address these two problems, we propose a user-centric system called "SocConnect" (Social Connect) for aggregating social data from different SNSs and allowing users to create personalized social and semantic contexts for their social data. Users can blend and group friends on different SNSs, and rate the friends and their activities as favourite, neutral or disliked. SocConnect then provides personalized recommendation of friends' activities that may be interesting to each user, using machine learning techniques. A prototype is also implemented to demonstrate these functionalities of SocConnect. Evaluation on real users confirms that users generally like the proposed functionalities of our system, and machine learning can be effectively applied to provide personalized recommendation of friends' activities and help users deal with cognitive overload.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

The advent of web 2.0 technology especially Social Networking Sites (SNSs), has changed the way people communicate. Clara Shih, in her book "The Facebook Era" (Shih, 2009), observes that social media such as Facebook (facebook.com) have transformed the socio-cultural landscape – people's behaviour, attitudes, interactions, and relationships. People spend more time on SNSs than ever, and prefer communication via SNSs over emails (Chisari, 2009). Every successful SNS has its unique features. Facebook allows a large number of third party applications to build on its APIs. Twitter (twitter.com) offers micro-blogging and an asymmetric following relation between users. MySpace (myspace.com) has a large user community interested in music. LinkedIn (linkedin.com) focuses on career and professional networking. Despite the diversity of SNSs and the fact that social media enriches people's lives, current SNSs have several significant limitations (Erétéo, Buffa, Gandon, Leitzelman, & Limpens, 2009), two of which motivate our work.

1.1. The "Walled Garden" problem

In the context of SNS, the "walled garden" problem is about the SNSs companies such as Facebook or Twitter having control over user's data. With the explosion of the number of SNSs, it is also common that one user engages with multiple SNSs. In July 2009, Anderson Analytics conducted an online survey¹ over 11,000 SNS users. The results show a high overlap of user

* Corresponding author.

E-mail addresses: zhangj@ntu.edu.sg (J. Zhang), yuw193@mail.usask.ca (Y. Wang), jiv@cs.usask.ca (J. Vassileva).

¹ http://www.readwriteweb.com/archives/who_uses_social_networks_and_what_are_they_like_part_1.php.

populations of Facebook, Twitter and LinkedIn. User-generated content, users' online activities, and their friendships are scattered over different SNSs. It becomes increasingly inconvenient for users to manage their social data and constantly check several SNSs to keep track of all recent updates. Even worse, people may have different accounts on the same SNS.

1.2. The “Network Overload” problem

Another problem of SNSs is information overload. The users of multiple SNSs see a large amount of social data generated by their network friends everyday. In this work, “social data” denotes status updates, posts of photos, links, likes, retweets, i.e. all new items that appear in the stream of updates in a SNS. The innovation of SNS has constantly increased the richness of social data. This causes significant information overload to users. Christian Kreutz in his blog described this specified kind of information overload as “network overload”.² Network overload is caused by two reasons: first, there is too much new social data appearing constantly on SNSs; second, social data often does not have explicit context. The first reason is fairly intuitive, but the second one needs some explanations.

SNSs generate huge amount of social data. However, lots of the data do not have explicit context. For example, the way the word “friend” is used in Facebook does not reflect the true meaning of the word in colloquial English. On Facebook, a user’s “friends” may include co-workers, college mates, and people whom the user barely knows but was too polite to decline their invitation. It is thus important to have a way of distinguishing these people. Users and their friends on different social networking sites may also have different kinds of relationships. For example, Facebook friends are mostly people whom the user already knows (Lampe, Ellison, & Steinfield, 2006), but users may have not met most of their Twitter friends in person. Without explicit context, it becomes very difficult to handle the huge amount of social data and too complex for users to make sense of the data. The contexts may include the type of social bound (the provenance, closeness, symmetry, etc.), the type of relationships (family, colleagues and friends in personal life), the common interests they share, the closeness of friendships, and the location of friends.

The “network overload” problem becomes more serious when the social data of the user is aggregated across different SNSs into one place by a social aggregator application. A social network aggregator is the application pulls together content from multiple social network service into a single location. The number of updates will increase significantly in this case. One way to deal with information overload is by providing recommendations for interesting social updates, which allows the user to focus her attention more effectively.

In this paper, we propose a system called “SocConnect” (short for social connect) which attempts to address these two problems, “walled garden” and “network overload”. SocConnect provides functionality to integrate social data across SNSs, and to allow users to organize their aggregated social data. The users can define social contexts of their social data. The added context can then help users to browse their social data. Moreover, SocConnect learns the users’ preferences using machine learning techniques and recommends new unread social data to them based on their preferences. As the evaluation of the effectiveness of our system, we collect data from real users to show the good performance on personalized recommendations of social data, and that users generally like the proposed functionalities of our system.

The rest of the paper is organized as follows. Section 2 provides a summary of related work on social network aggregators and recommendation. Section 3 presents the proposed schema used by SocConnect to integrate social data across SNSs, and the main functionalities of SocConnect. Section 5 describes a prototype of SocConnect to demonstrate its functionalities. Section 6 evaluates the effectiveness of the personalized recommendation functionality and the usability of all the functionalities. Finally, Section 7 summarizes the contributions of our current work and proposes some future directions.

2. Related work

SocConnect is a social network aggregator and recommender. Here, we discuss important requirements for integrating social data across different SNSs and compare with other existing social network aggregators. We also survey the state-of-art recommender systems and clearly point out that our recommendation is content-based and shares similarity with text recommendation.

2.1. Social network aggregators

One important requirement for integrating social data across different social networking sites is a unified ontology to represent social data (Chisari, 2009). SNSs have their own syntaxes and terms for representing social data. The academic and open web communities have put great effort to develop standard ontologies for the representation of social data. There are several major standards, including FOAF (foaf-project.org), XFN (gmpg.org/xfn/), GUMO (Heckmann, Schwarzkopf, Mori, Dengler, & Kroner, 2007) and Activity Stream (activitystrea.ms).³ These standards have solid foundations; some of them have already been adopted by social networking sites and other IT companies. For example, the activity stream has been recently embraced by both Facebook and MySpace.

² <http://www.crisscrossed.net/2009/10/15/network-overload-the-burden-to-deal-with-too-many-social-network-sites/>.

³ More details about Activity Stream will be given in Section 3.1.

Another important concern in integrating social data is to keep the context of the data (Erétéo et al., 2009). The contexts may include the type of social bond (the semantics) of relationships (family, colleagues and friends in personal life), the common interests they share, the closeness of friendships, and the location of friends. Therefore, the ontology should also be able to allow users to express the context of social data. There are two solutions for the expression of contexts. One common way is a top-down approach that pre-defines sets of vocabularies to describe different types of social contexts. However, social contexts contain too many dimensions and too many possible variables along each dimension, of which only a few may be relevant to any given user. The process of selecting the relevant value in each dimension from a pre-defined ontology would be too hard for a user. The second solution is to let users themselves express social contexts by, for example grouping or rating their social data. This solution is more flexible and feasible, and we use it in our work.

There have been some attempts to create personal portals that aggregate a user's accounts on different social networking sites, for example, the Seismic Desktop (seismic.com), power.com, the social web browser Flock (flock.com), and TweetDeck (tweetdeck.com). They allow the user to view her pages and status updates on different social networking sites in one place. In this way, the users do not have to login to many different sites to view the updates of their friends. However, these applications do not allow users to blend or group their friends from different places. They provide just a single-login interface in which users can switch between different tabs, one for each social networking site. Android (android.com) allows to integrate social networking contacts.

Bojars, Passant, Breslin, and Decker (2008) have been working on the SIOC project (Semantically-Interlinked Online Communities). This project shares similar focus with our work: social network portability and semantic web technologies. They propose the SIOC ontology, which mainly focuses on users, implicit friendship, and social contents (primarily photos and discussions) in online communities such as online forums and Weblogs where contexts of social data are not so different. In contrast, we focus mainly on developing a user-centric system for integrating users' social data (including explicit friendship) on different social networking sites, and that allows users to organize their social data and to create their personal contexts for the social data. We also provide personalized recommendation of friends' activities that are interesting to users.

2.2. Recommendation

There is a lot of research in the area of recommender systems dating back from the mid 1990s. There are two main types of recommender systems: content-based (or feature-based) (Chen, Nairn, Nelson, Bernstein, & Chi, 2010) and collaborative (social) (Resnick, Lacovou, Suchak, Bergstrom, & Riedl, 1994). 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 (Carmagnola, Vernerio, & Grillo, 2009) 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. 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 uses machine learning techniques to make predictions based on the user's previous choices and generate personalized recommendations. We also specifically address the challenge of providing recommendations across different domains (i.e. SNSs) after social data is integrated into our SocConnect from these domains.

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 (Claypool et al., 2000): (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.

The common models of representing text documents are Vector Space Model (VSM) (Salton, Wong, & Yang, 1975), Standard Boolean Model (BIR) (Lancaster & Fayen, 1973), and Probabilistic Model (van Rijsbergen, 1979). Among them, vector space is the most widely used one 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:

$$TF_i = \frac{N_i}{\sum_i N_i} \quad (1)$$

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:

$$TF - IDF_i = TF_i \times \lg \frac{|A|}{|A_i|} \quad (2)$$

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

3. SocConnect

In this section, we first present a schema for representing integrated social data across SNSs. We then describe in details the proposed functionalities of our SocConnect system and the implementation of the functionalities.

3.1. A schema to integrate social data across SNSs

To represent heterogeneous social data across SNSs, a unified schema is required. As described in Section 2, a variety of standards and ontologies serve this purpose, such as FOAF, activityStream, and the SIOC project. However, any single one of them cannot fully meet the requirements of SocConnect. FOAF's scope is about the users and the relations among them. And, activityStream focuses on describing user's online activities. The scope of the SIOC project is mainly on blogs and forums. Therefore, we develop an adapted schema based on FOAF and activityStream. The philosophy behind activityStream is that the essential elements of SNSs include actors and their activities. Every user is an actor; every movement of an actor is an activity, such as adding a new friend, publishing a new blog article, and commenting on others' articles. Each activity has a type, such as Twitter update and retweet, and sharing a link or a Facebook photo. The type of an activity represents the feature of this activity. In addition, social data is inherently "URI-based"; almost every piece of social data has an URI (Unique Resource Identifier). For example, each Facebook user has his or her own facebook homepage as an URI, each Twitter update has a permanent address (such as <http://twitter.com/username/status/9993890828>), and each Flickr (www.flickr.com) photo has its URL. This makes social data easy to be interlinked. The design of our schema for representing social data takes advantage of this feature.

The proposed schema is presented in Fig. 1. There are six entities in the ontology: SNS account (SNSAcc), integrated account (person), activity, tag, group, and rating. SNSAcc represents a user account on a SNS. Each SNSAcc has a source which is a SNS, such as Facebook, Twitter, and MySpace. The profile of a SNSAcc indicates what kinds of data are collected by the SNS. For example, Facebook keeps lots of information about each user. On another hand, Twitter only stores very simple user information. Person represents a user who holds one or more SNS accounts. For example, a user on Facebook also can have a Twitter account. Activity represents generic information about activities appearing on SNSs. Activities can have different

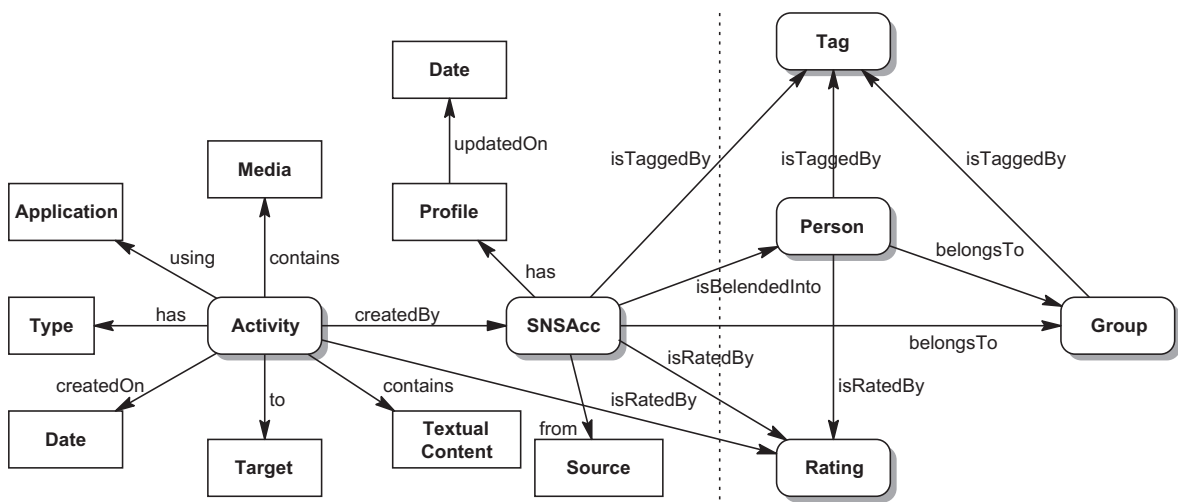


Fig. 1. The schema for integrating social data.

types, such as user status updates, events like a new friend added by the user, or a new third party application used by the user. Activities may be generated by different applications, and contain some form of media such as video and textual content. An activity may be generated towards a particular user (called target). Tag represents a user-generated label. Tags are used to represent contextual information of social data (Heckmann et al., 2007). Group represents a user-defined group for keeping friends together. A member of a group can be a SNSAccount or a Person. Rating represents a user-generated interest level (favourite, neutral or disliked). Ratings are used to represent user preferences on social data.

The six entities are interlinked among each other. Each SNSAcc has a set of activities belonging to the user's SNS account. A person may have a set of SNSAccs and a number of activities associated with each SNSAcc. A group may contain a number of persons and SNSAccs as its members. One SNSAcc can belong to multiple persons or groups, and one person can also belong to more than one groups. The domain objects of SNSAcc, person and group can have a set of tags. The domain objects of SNSAcc, activity and person may have a set of ratings. The activity class is the core of this domain. Each activity has a SNSAcc as its actor. Activities of users or their friends incrementally fill social networks with contents. SNSs are essential sources of activity streams. Users and their friends are the actors of the activities.

3.2. Functionalities

Based on the schema presented in the previous section, SocConnect aggregates social data from different SNSs, by proposing four categories of functionalities: (1) connecting different SNSs and loading users' social data; (2) allowing users to manage their friends and assign context to their social data; (3) browsing social data; and (4) personalized recommendation of social data. The first three functional categories are proposed to address the problem of "walled garden", and provide a way for users to aggregate and organize their social data from different SNSs. The fourth category of functionalities is used to address the problem of "network overload" whose details will be given in Section 4.

3.2.1. Loading social data

SocConnect uses authentication methods provided by different SNSs and invokes their APIs to retrieve users' friends information and their activities on these sites. There are three authentications methods used by current SNSs: basic authentication, OAuth, and custom authentication. Basic authentication asks SNS users to provide their SNS usernames and passwords to external applications, e.g. SocConnect. Basic authentication is easy to implement. OAuth is an open protocol about how to request and handle user authentication between systems. Custom authentication is a special authentication method that only works for one SNS. SNSs often provide multiple authentication methods. For example, Twitter provides both basic authentication and OAuth; Facebook provides both OAuth and custom authentication. SocConnect uses both basic and custom authentications. After authentication, SocConnect invokes APIs provided by SNSs to retrieve raw data (in XML or JSON) from SNSs, and then translates the data using the schema described in Section 3.1.

3.2.2. Managing friends

The second functional category, "managing friends" contains two functions: blending friends and grouping friends. In most cases, there is some level of overlap between the sets of a user's friends on different SNSs. This function allows the user to merge the different accounts of a friend across SNSs, to create a single "person". It is a unique feature of SocConnect (Yeung, Liccardi, Lu, Seneviratne, & Berners-Lee, 2009). The friend can have different user accounts on different sites, but the user knows that they refer to the same person (something that no data mining algorithms can find out accurately). It is up to the user to create the mapping between her friend's accounts across different sites and assign an integrated account to represent the same friend. In this way, the user can have an integrated view of all activities of this friend, despite which social networking sites the activities come from. Compared to the other social network aggregators that only present social data at the same place, SocConnect provides users with the possibility to integrate scattered social data.

The second function in the "managing friends" category is to group friends. Users can put their friends, both individual SNS accounts and blended "person" accounts, into groups. This function allows users to express the contexts of friendships, which are the shared characteristics or interests between friends.

When a user blends a friend's SNSAccs, SocConnect creates an instance of the Person class, and adds these SNSAccs into the instance. The activities associated with each account link to the Person instance. When a user defines a group and adds SNSAccs and persons into the group, SocConnect creates an instance of Group, and adds these accounts and persons into the instance. The activities associated with SNSAccs and persons link to the Group instance.

3.2.3. Browsing social data

The third functional category, "browsing social data" also has two functions. Social data can be browsed according to tags provided by users. Users can tag friends (both individual social network site accounts and integrated accounts), groups, and individual social updates. After tagging, the tags will be added into the instance of the SNSAccount, Person or Activity class respectively. Users can then browse social data based on these tags. Tagging allows the user to add richer context description to their friends, in addition to that achieved by grouping.

Another function is to allow users to browse social data based on groups. Users can view the activities of the members in the groups which they are interested in. Note that the function of browsing social data by tags and that by groups are different, and both are necessary. Normally, the number of groups created by a user is not expected to be very large. Otherwise,

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.

it will become difficult for the user to manage all the groups. Thus, grouping friends should be normally used to create large groups, such as a group for classmates from a same university. Tagging friends provides a flexible way for the user to view activities of only a few friends for whom the user does not want to create a separate group. Thus, it should be normally used to create small or short-term “groups”, such as a group for this Friday’s party or a particular trip, in an indirect manner.

3.2.4. Personalized recommendation of social data

To relieve the network overload, SocConnect provides personalized recommendations of activities to individual users according to a prediction generated using their ratings on previous social data. Users are allowed to rate social data. When a user rates a friend or one of the friend’s activities, the rating will be added into the instance of the SNSAccount, Person or Activity class respectively⁴. When new social data is retrieved, users will be provided with the recommendations about whether the new social data is interesting to them. In this way, non-interesting data can be filtered out. The detailed implementation of this function is described in the next section.

4. Personalized recommendations in SocConnect

Our approach of personalized recommendation in SocConnect is content-based rather than collaborative. In this section, we propose a list of potential non-textual and textual features for representing each activity and present several machine learning techniques used to predict users’ preferences on activities from the social networking sites of Twitter and Facebook.

4.1. 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 (i.e. favour) 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 (Platt, 1999), Bayesian Networks, and Radial Basis Functions (Mitchell, 1997). 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 6.

4.2. 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.

4.2.1. 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.

⁴ An alternative way of implementation is to define a rating profile and specify the type (Person, Activity) that each rating is part of.

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: $\in \{\text{favourite, neutral or disliked}\}$

The above non-textual features of activities can be obtained through the APIs offered by social networking sites. 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.

4.2.2. Textual features

In the text analysis part, we first remove the stop words and URL links⁵ in each activity. Two vector spaces are then calculated for each of collected activity, one is using TF and another one is using TF-IDF.⁶ 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.

We then sum up the weight values for each term in all the favourite, neutral and disliked activities in training data, respectively, based on the calculated vector spaces of each activity. 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 values⁷ 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 6.

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.

4.3. Heuristic to supplement learning and prediction

We assign an approximate weight to the new activity as follows:

$$w = \begin{cases} 0.5 & \text{if predicted as favourite;} \\ 0 & \text{if predicted as neutral;} \\ -0.5 & \text{if predicted as disliked.} \end{cases} \quad (4)$$

These predictions are based on the features of each activity. We also present how the social context, expressed by the user by grouping friends in SocConnect, influences the recommendations.

As described earlier, SocConnect allows users to create groups and add friends into the groups. A group implies the existence of some commonalities among the members of the group or some activities that group members have been doing together. The group information provides an indirect indication about users' preferences on activities. For example, if many

⁵ Some URL links may contain useful word information. We will try to take them into consideration in our future work, to check whether this will bring improvement.

⁶ For future work, we will calculate IDF values based on a larger collection of activities for IDF to more accurately work as a feature of term informativeness.

⁷ The reason why we do not use the detailed TF and TF-IDF values for each term as features is to reduce the number of features. Learning with a large number of features generally requires a lot of training data.

activities of members in a given groups are considered as favourite by a user, the activities of the other friends classified by the user in this group will likely also be interesting to the user. Based on this heuristic, we extend the results of machine learning, by adjusting the weight of an activity. More specifically, for a friend in a group, if the number of favourite activities of other group members is larger than that of disliked activities, the weight of each activity from this friend will be increased. Otherwise, the weight will be decreased. Formally, suppose that the number of liked (marked as “favourite”) activities of other group members in the group is F , and the number of disliked activities from them is D , then the weight of an activity from the friend will be updated as follows:

$$w = w + 0.5 \times \frac{F - D}{F + D} \quad (5)$$

Note that in extreme cases where every activity of the other group members is considered favourite, the weight of the friend’s activity will be increased by 0.5. On another hand, if every activity of the other group members is considered disliked, the weight of the friend’s activity will be decreased by 0.5. Also note that w stays the same if every activity of the other group members is considered neutral by the user ($F + D = 0$). For a friend who belongs to several groups, the effect of the heuristic on the weight of the friend’s activity will be averaged over these groups.

This extension brings two extra levels of user interests in activities, namely “very favourite” and “very disliked”. This expands the range of levels of distinction for user interests from 3 to 5 levels, which has been commonly used in many popular rating systems, such as Amazon (amazon.com) and TripAdvisor (tripadvisor.com). The mapping between the interest levels of users in activities and the numerical weight for the activities is summarized in Table 3.

5. Demonstration of SocConnect

We provide several screenshots to demonstrate the user interface of SocConnect. This interface is an early prototype implementing the main functionalities rather than the ultimate interface for SocConnect. We use Facebook and Twitter for the purpose of demonstration. Suppose that a user Jane has accounts on both Facebook and Twitter. SocConnect retrieves Jane’s social data on these two sites. The social data of her friends can then be managed, browsed and filtered by her SocConnect dashboard based on her personal needs or interests. We step through an example to show more specifically what Jane can do with the application. The social networking site accounts of the actual users in the screenshots are blacked out to protect their privacy.

Jane can use SocConnect to blend her friends who have social networking site accounts on both Facebook and Twitter. As shown in Fig. 2, there are three lists in the upper part. The left list contains Jane’s friends on Twitter and the middle one contains her friends on Facebook. Jane drags her friend Linda’s Twitter account “LindaTwit” from the left list and Linda’s Facebook account “LindaFace” from the middle list to the lower list. By clicking the “Blend” button shown in the bottom of the figure, Linda’s accounts in the lower list are joined into a “blended” person. Jane gives a name “Linda” for the blended person. The third list in the upper-right part of the screen shows the list of all Jane’s “blended” persons. Linda will be added to the list.

Jane can also use SocConnect to group her friends together. As shown in Fig. 3, the interface for this function is similar to the interface for blending friends. To add members into a group, Jane can drag her friends’ accounts from the three lists in the upper part of the figure and drop them into the list in the lower part. She drags her friends in New Jersey into the lower list, including John and Bob from the Twitter list and Amy from the Facebook list. She also drags the blended person Linda into this list from the list of blended persons. She gives the name “friends@NJ” to the group and clicks the button of “Create a new group” in the bottom of the screen. A new group is then created for Jane, and the list of Jane’s groups is shown in the right most list in the lower-right part of the screen. A user can also put her friends in different groups, e.g. John can be both a member of Jane’s “friends@NJ” group and her “friends@SK” group.

The function of grouping friends provides a flexible way for users to organize their friends by contexts. It also allows users to browse only social data from the members of a particular group. For example, Jane can check news from friends@NJ by clicking the group name listed in the right most list called “Groups” in Fig. 4. The members in this group will appear in the middle list, and the updates from these members will appear in the left most list.

To allow for more expressive representation of context information, users can add tags to their friends and groups. They can choose any of these tags as a keyword, and the application will display the social data that relates to the tag. As shown in Fig. 4, Jane can add a tag to her friend John by clicking the button “tag” beside John’s icon. A separate window pops up as

Table 3
Interest Level, activity weight and colour presentation.

Interest level	Activity weight	Colour
Very favourite	$0.6 \leq w \leq 1$	Persimmon
Favourite	$0.2 \leq w < 0.6$	Tawny
Neutral	$-0.2 \leq w < 0.2$	Maroon
Disliked	$-0.6 \leq w < -0.2$	Burgundy
Very disliked	$-1 \leq w < -0.6$	Thyrian purple

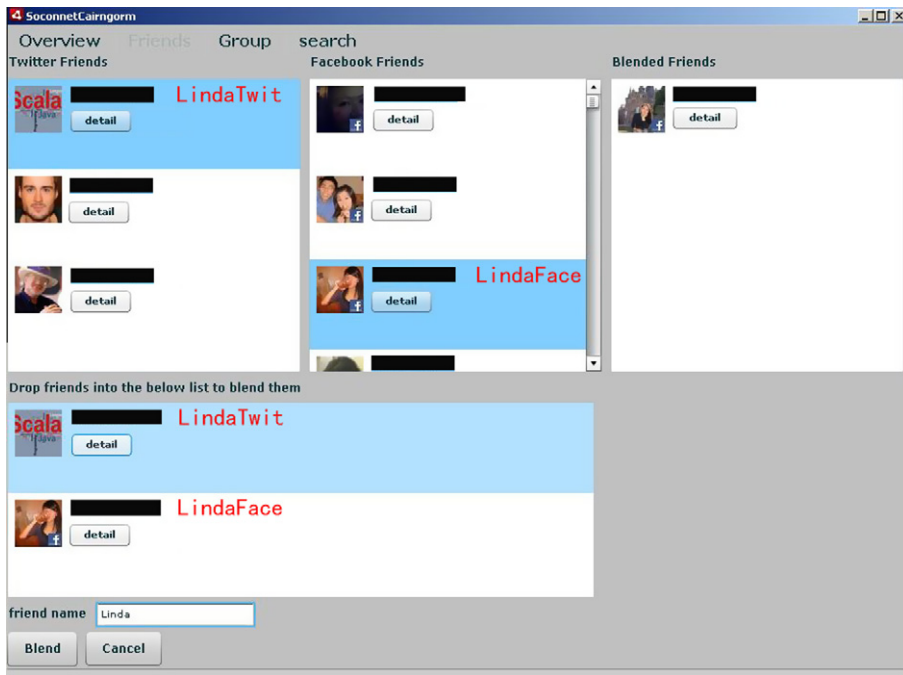


Fig. 2. Blending friends.

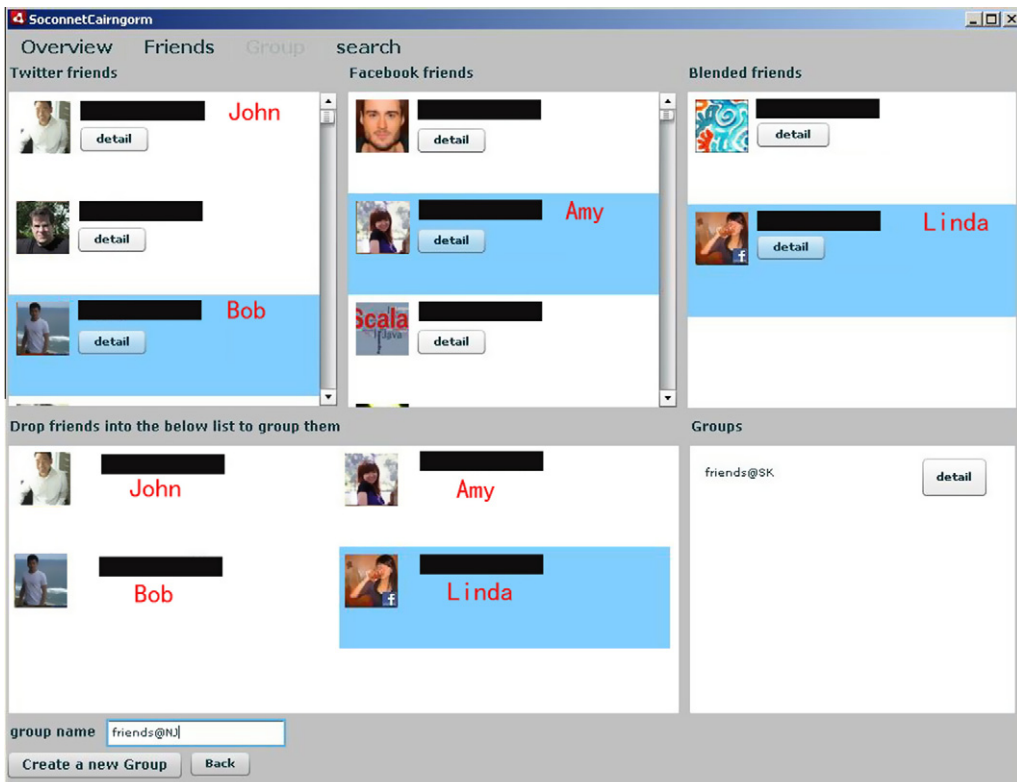


Fig. 3. Grouping friends.

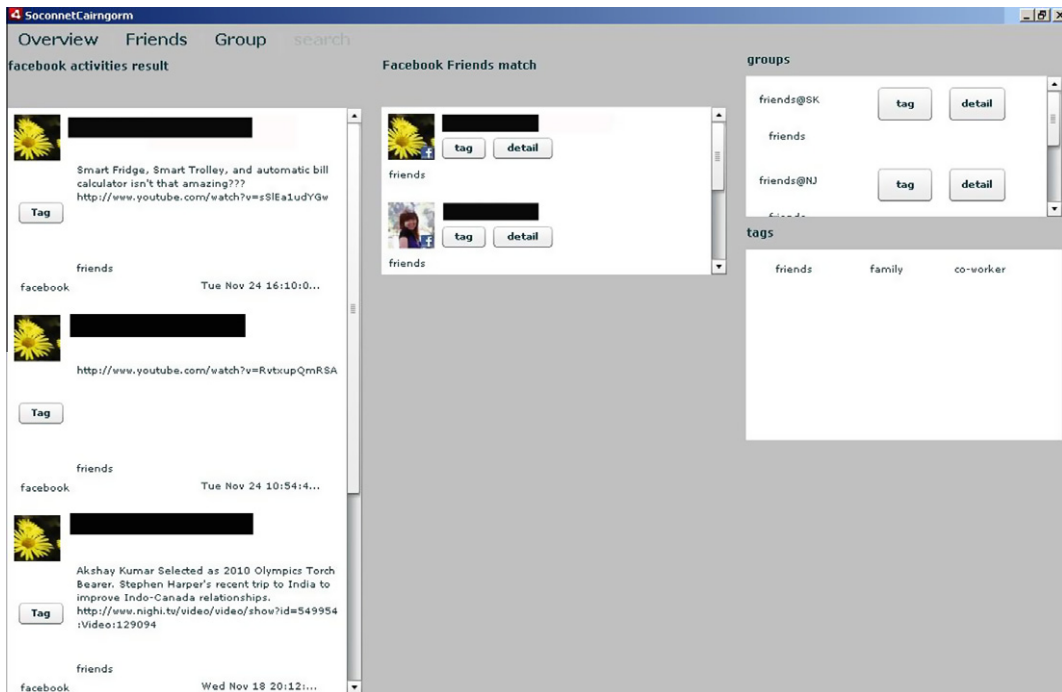


Fig. 4. Browsing social data.

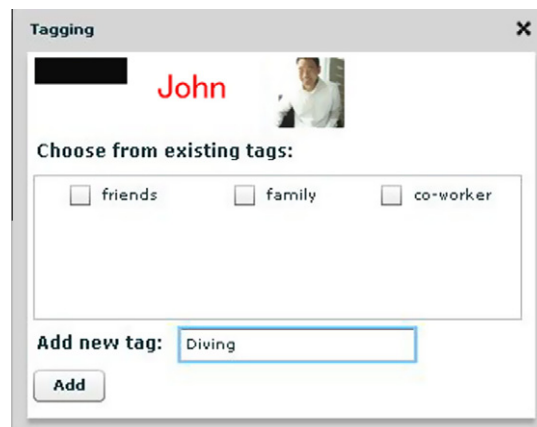


Fig. 5. Tag a friend.

shown in Fig. 5. Jane can choose an existing tag from the list of tags or add her own tag. In this case, Jane adds her own tag “Diving” to John and clicks the “Add” button (Fig. 5). The list of all Jane’s tags is shown in the right most list marked by “Tags” in Fig. 4. Jane can view the activities of all her friends that relate to diving by clicking the tag “Diving”. All her friends who are tagged by “Diving” will appear in the middle list, and the updates from these friends will appear in the left most list.

The recommendations for the activities that the user may find interesting are integrated in the display of the activities in the activity stream that the user views in the interface of SocConnect (see Fig. 6). Colours in a spectrum that allows people with the most common type of colour-blindness (red-green) to distinguish,⁸ is used to represent if an activity is recommended or unrecommended according to the predicted interest level calculated for the activity (Table 3). In this way the recommendation is unobtrusive, and can be easily ignored, but in the same time, it is intuitively clear for the user since it uses the metaphor “hot” item (displayed in bright orange background, yellow text) and “cold” item (dark purple background, blue text). The met-

⁸ Images can be tested for appearance with simulated colour blindness at: <http://www.colblindor.com/coblis-color-blindness-simulator/>.



Fig. 6. An example of visualization.

aphor allows representing a spectrum of recommendations with a larger number of values than 5, but we have picked 5 colours to represent transitions from hot through neutral (earth colour) to cold.

We have tested a visualization of items with different levels of interestingness using this metaphor with users in previous work (Webster & Vassileva, 2006) and it was shown to work very well in quickly focussing user attention to the recommended items, while still allowing them to explore all items. This kind of recommendation visualization has been successfully deployed in the Comtella-D system in four classes with over hundred students for 2 years. That is why we decided to use it in SocConnect.

6. Evaluation

6.1. Performance of personalized recommendation

We first 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 1600 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. In our experiments, a machine learning technique predicts whether an activity from a subject in the testing data is “favourite”, “neutral” or “disliked”. And, the performance of the machine learning techniques is averaged over all subjects, and the averaged results are reported in the following sections. Note that the baseline that would always predict “neutral” for each activity is 55% in our case. We will see that the performance of our learning algorithm is much better than the baseline.

6.1.1. Performance when using only non-textual features

We first used only the set of non-textual features summarized in Table 1. Fig. 7 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.

6.1.2. 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. 8 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.

6.1.3. 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 4. 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

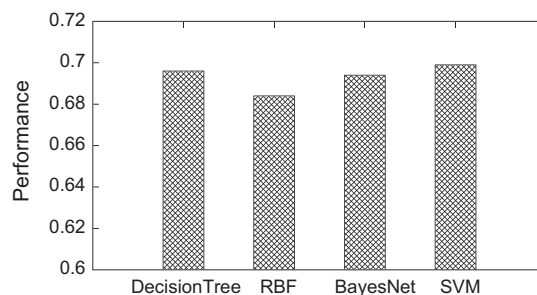


Fig. 7. Performance when only non-textual features are used.

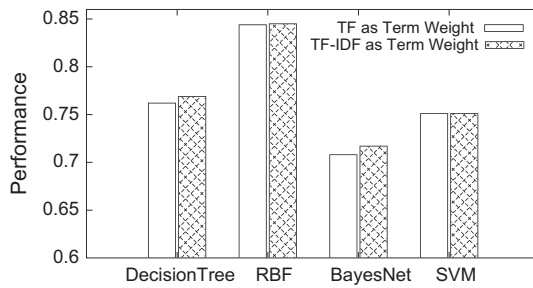


Fig. 8. Performance when three textual features are used.

Table 4

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

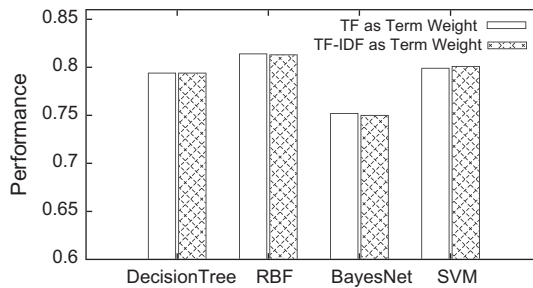


Fig. 9. Using S_F, S_M, S_D and non-textual features.

feature C and the non-textual features (up to 79.4%) is much better 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. 9 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%).

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. 10. In most of the cases, the three textual features provide

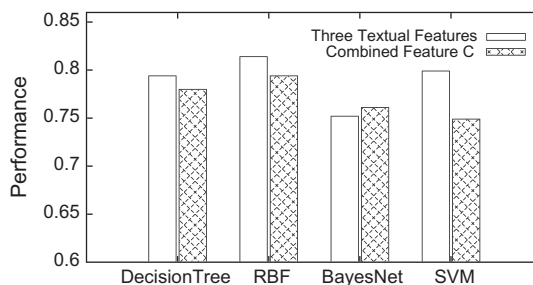


Fig. 10. Performance comparison between textual features.

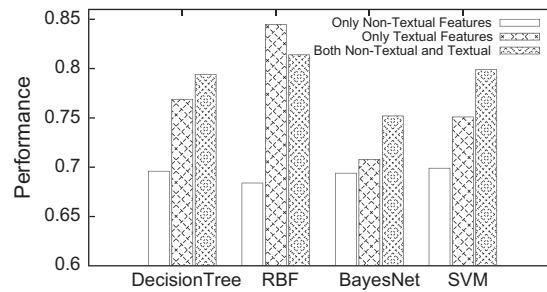


Fig. 11. Performance comparison for different features.

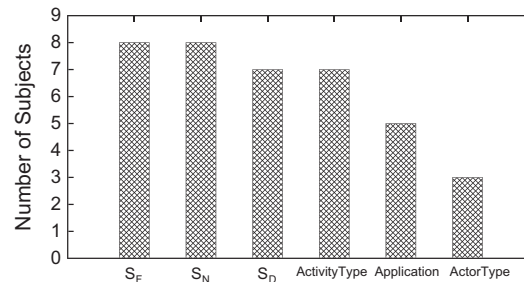


Fig. 12. The most important features for learning.

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.

6.1.4. 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. 11. 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 textual and non-textual features approach yielded significantly better results with all four algorithms. In the real user evaluation in Section 6.2, we use RBF with the combined textual and non-textual features, as it produces the best performance among all the machine learning techniques when the combined textual and non-textual features are used.

Using Weka's feature selection function, we can see which features are more important for individual users. We summarize in Fig. 12 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.

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.

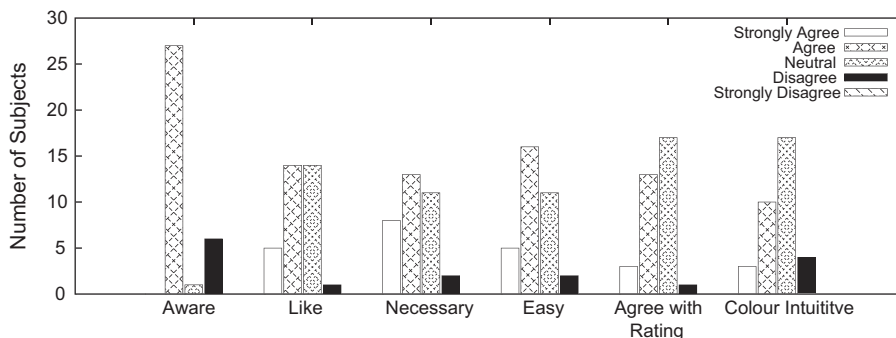


Fig. 13. User feedback on the function of personalized recommendation.

6.2. Usability of SocConnect's recommendation related functionalities

We also evaluated SocConnect's functionalities in real use over a period of time. We recruited Facebook and Twitter users to download SocConnect on their computers after signing consent to participate in the study. They used SocConnect in an uncontrolled environment for two weeks.⁹ The SocConnect server kept the logs of user interaction history with SocConnect, such as the login, blending, grouping, tagging, and rating actions. After the two weeks of usage, the participants were asked to fill in a user satisfaction survey. The survey has three sections, including basic information, functionality feedback, and general questions. The basic information section collects the participants' contact information and Twitter and Facebook usernames. The functionality feedback section is organized into several sub-sections, each of which collects participants' feedback on SocConnect's functionalities (blending friends, grouping friends, tagging friends, searching by tags, recommendation, and rating updates) for the criteria of "Aware of the functionality", "Like the functionality", "The functionality is necessary", and "The functionality is easy to use". For these questions, the answers are Likert-scale with five options, from "strongly agree" to "strongly disagree". For the functionalities "recommendation" and "rating", there are a few additional questions. For the recommendation functionality, the participants were asked how much they agree with the recommendation results, and how intuitive they find the visualization colour. For the rating functionality, they were asked how much they are willing to rate updates in order to gain better recommendation results, how easy it is to decide to rate one activity, and whether the participant has a consistent rating criterion. The general question section contains several questions measuring the participants' overall satisfaction with SocConnect.

Invitations were sent through email and the study was advertised on Twitter and Facebook. Forty-three participants responded to the advertisements and installed the application. Thirty-four participants answered the survey. Most of them were active users, who accessed SocConnect several times over the two-week period, so they comprise a valid sample for the study. The results of the study are summarized in the following sections. Note that we only present and discuss the results for recommendation related functionalities.

6.2.1. The function of personalized recommendation

Fig. 13 shows the participants' feedback on the recommendation function of SocConnect. Six participants were not aware of this function, the largest proportion among all functions. Quite a lot of users did not know whether they liked the recommendation function. Most users did not know whether the recommendations are accurate, but only 1 user disagreed with the recommendations. Most of the users thought that the function is necessary and easy to use. The recommendation visualization seems not intuitive. Less than half of the participants found the colour intuitive. The problem may be that the highlighted updates were very likely to be buried among many neutral updates requiring the user to scroll a lot to find highlighted activities. Because users receive many updates, the recommended updates may be easily overlooked. The two participants, who rated the most, did not notice that SocConnect has generated recommendation for them. One participant suggested to separate the updates from the recommendations. Further, the recommendation function was not transparent enough for the users. The current implementation of SocConnect reminds users to rate more updates to receive recommendations. The recommendation algorithm requires at least ten ratings on ten different updates, before it can generate predictions for users. However, users had no idea how many ratings are required, and whether the recommendation function is already working for them. One participant stated that this non-transparency should be fixed in the future.

6.2.2. The function of rating activities

Fig. 14 presents the participants' feedback on the function of rating activities. While most of the participants were aware of the rating function, liked it, and thought that it was necessary and easy to use, only half of them (17) were willing to rate,

⁹ In future work, we will also conduct a comparative user study between the current version of SocConnect and a baseline system without some functionalities.

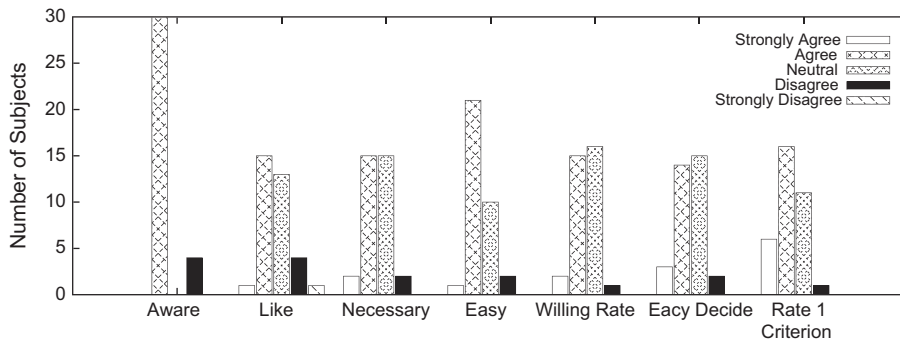


Fig. 14. User feedback on the function of rating activities.

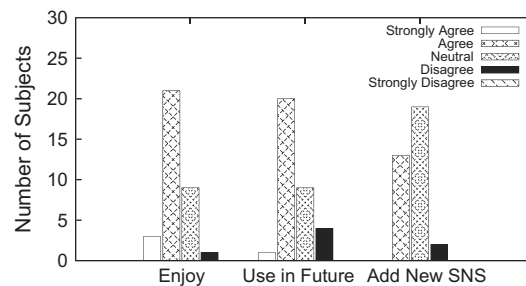


Fig. 15. Users' general feedback.

and thought it was easy to decide how to rate. This is normal because rating activities requires users' effects. Unless they strongly realize the benefit of recommendations generated based on their ratings, they will not be willing to do so. This suggests that in future work, we need to emphasize more on showing the benefits of recommendations and how users' ratings will improve the accuracy of recommendations. From the results, the good thing is that most of the users thought that they have a consistent criterion for rating activities, which makes the learning of user preferences easy and accurate. This partially explains the good performance of our personalized recommendation shown in Section 6.1.

6.2.3. General feedback

The general feedback is summarized in Fig. 15. Most users (24) enjoyed using it, only one user did not enjoy, and 20 users liked to use it in the future. Thirteen users suggested to add other SNSs in the future, such as Renren (renren.com) and LinkedIn where Renren is widely used in China. These results indicate generally positive attitude of users towards the SocConnect system including its recommendation related functionalities.

6.3. Summary of the evaluation results

Several important conclusions can be drawn from the evaluation results of personalized recommendation presented in Section 6.1: (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.

From the study results for the usability of SocConnect functions in Section 6.2, we can conclude that SocConnect provides users with a set of useful functions. Each functionality was found useful, necessary and easy to use by the majority of the participants. The general feedback on SocConnect was also quite positive. Because quite a lot of responses from the participants in the user study were "Neutral", we also performed the one-way ANOVA test of replication with 0.05 p -value on participants' feedback on the recommendation function, the function of rating activities, and the general feedback, respectively. The analysis results suggest that the differences between the numbers of positive responses and those of negative responses for the three tests (each for one aspect) are all statistically significant, as the p -values of the three tests are 0.1345, 0.1474 and 0.0512, respectively. Many participants found that the recommendation functionality was useful in general, necessary, easy to use, and accurate. However, quite a lot of participants also did not know whether the recommendations are accurate.

This may be due to the fact that participants did not provide a sufficient number of ratings to train the recommender. Yet, this result points out that it is necessary to make participants aware that they need to rate in order to receive accurate recommendations.

7. Contribution and future work

In this work, we proposed the SocConnect system to personalized aggregation and recommendation of social data from different social networking sites, to address the two important problems faced by SNS users, “walled garden” and “network overload”. SocConnect provides a set of functionalities, including blending and grouping friends, tagging friends and social activities, and the personalized recommendations of social activities. Results of our user study indicate strong support for the functionalities of SocConnect. Evaluation on real user data also confirms the sufficient performance for the personalized recommendations of social data. SocConnect thus provides effective social data aggregation and recommendation.

In summary, our work has the following major contributions: (1) SocConnect allows users to define their personal contexts of social data aggregated from different SNSs and to indicate their interest level (favourite, neutral or disliked) for social updates. To the best of our knowledge, no other social network aggregator allows such functionalities; (2) SocConnect provides personalized recommendation of social activities that may be interesting to individual users. There have been previous works on recommending friends and groups, but to the best of our knowledge, SocConnect is the first aggregator to provide content-based recommendations for social updates in SNSs; (3) Based on extensive evaluation, we suggest a particular machine learning method and a set of features for learning user preferences that can provide the best performance on personalized recommendation. This is useful for other researchers seeking to develop content-based recommender systems in SNSs.

For future work, we are interested in exploring more deeply the relative importance of different features of social networking activities, to further improve the performance of personalized recommendation of activities. Other features that may be worth looking at include textual content of activities and the targeted friends of friends in activities. One particular challenge for future work would be how to handle the evolution of activity types over time. We will also look into the sharing of ratings of activities among users of SocConnect. In this case, Collaborative Filtering will be used for predicting whether an activity is interesting to a user based on other users' ratings for the activity. And, Facebook's core concept, Open Graph that provides connections between users and with everything they care about, would also be helpful in providing recommendations of activities.

We also plan to add more social networking sites (e.g. LinkedIn, Renren, Flickr, etc.) into SocConnect and allow users to choose which ones they want to integrate. We can then conduct extensive evaluation on the recommendation performance of our system based on data collected from those social networking sites. Some social networking sites (e.g. Facebook) may also provide recommendations on status updates. We will extend SocConnect to integrate those recommendations and allow users to choose to view recommendations from social network sites or from SocConnect. We will also develop a mobile version and a web version for SocConnect. Users can then access SocConnect on mobile devices at any time and any place, allowing more flexibility.

References

- Shih, C. (2009). *The Facebook Era: Tapping online social networks to build better products, reach new audiences, and sell more stuff*. Prentice Hall PTR.
- Chisari, M. (2009). The future of social networking. In *Proceedings of the W3C workshop on the future of social networking*.
- Erétéo, G., Buffa, M., Gandon, F., Leitzelman, M., & Limpens, F. (2009). Leveraging social data with semantics. In *Proceedings of the W3C workshop on the future of social networking*.
- Lampe, C., Ellison, N., & Steinfield, C. (2006). A face(book) in the crowd: Social searching vs. social browsing. In *Proceedings of the ACM special interest group on computer-supported cooperative work*.
- Heckmann, D., Schwarzkopf, E., Mori, J., Dengler, D., & Kroner, A. (2007). The user model and context ontology gumo revisited for future web 2.0 extensions. In *Proceedings of the international workshop on contexts and ontologies: Representation and reasoning*.
- Bojars, U., Passant, A., Breslin, J., & Decker, S. (2008). Social network and data portability using semantic web technologies. In *Proceedings of the workshop on social aspects of the web*.
- Chen, J., Nairn, R., Nelson, L., Bernstein, M., & Chi, E. (2010). Short and tweet: Experiments on recommending content from information streams. In *CHI '10: Proceedings of the 28th international conference on human factors in computing systems* (pp. 1185–1194).
- Resnick, P., Lacombe, N., Suchak, M., Bergstrom, P., Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the ACM conference on computer supported cooperative work*.
- Carmagnola, F., Vernerio, F., & Grillo, P. (2009). Sonars: A social networks-based algorithm for social recommender systems. In *Proceedings of the 17th international conference on user modeling, adaptation, and personalization*.
- Claypool, M., Claypool, M., Le, P., Le, P., Waseda, M., Waseda, M., et al (2000). Implicit interest indicators. In *Intelligent user interfaces* (pp. 33–40). ACM Press.
- Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11), 613–620.
- Lancaster, F. W., & Fayen, E. G. (1973). *Information retrieval On-Line*. Los Angeles, California: Melville Publishing Co..
- van Rijsbergen, C. J. (1979). *Information Retrieval* (2nd ed.). London: Butterworths.
- Yeung, C. A., Liccardi, I., Lu, K., Seneviratne, O., & Berners-Lee, T. (2009). Decentralization: The future of online social networking. In *Proceedings of the W3C workshop on the future of social networking*.
- Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, & A. Smola (Eds.), *Advances in kernel methods: Support vector learning*. MIT Press.
- Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.
- Webster, A., & Vassileva, J. (2006). Personal relations in online communities. In *Adaptive hypermedia and adaptive web-based systems* (pp. 223–233). Dublin, Ireland: Springer LNCS.