**WINTER TRAINING PROJECT REPORT ON**

Data Mining To Produce Novel and Serendipitous Friend Recommendations In A Social Bookmarking System

**Under guidance of Mr. Rahul Katarya**



**UNDERTAKEN BY:**

Anmol Mahajan (2K13/IT/014)

Ashish Pandey (2K13/IT/022)

Bhrigu Srivastava (2K13/IT/027)

Divyesh Kumar (2K13/IT/034)

**Department Of Computer Science and Engineering,**

**DELHI TECHNOLOGICAL UNIVERSITY**

***Abstract:***

As the world and the internet continues to change rapidly, the extent and complexity of many web sites change along with it. It becomes difficult and time exhausting for the users of these websites to find the information they are looking for. Some web sites provide personal information to the users by letting them choose from a set of predefined topics of interest. Users do not always know their interests in advance and their interests may change with time which would require them to change their selection at regular intervals. **Recommender systems** provide personalized information by studying the user’s interests from traces of communication with that user.

This report is based on behavioral data mining and suggests a friend recommender system that works in the social bookmarking application domain. Experimental results display how this type of mining is able to produce precise friend recommendations, thus giving users a way to know bookmarked resources that are both new and serendipitous. Using this approach, the effect of the “interaction overload” and the “over-specialization” problems is minimized.

The report is subdivided into various sections. In section 2, related work, we summarize the theoretical approaches and the associated terms used in the paper. Section 3 explains our proposed work for the given problem of recommending novel and serendipitous resources to the users in a social bookmarking system. In section 4, we focus on the actual implementation of our system in the form of algorithms. Section 5 explains the framework that our implementation of the bookmarking system uses. In section 6, we deal with the results obtained by applying the proposed approaches to a real dataset, Delicious and try to analyze how they yield better results.

In our proposed work, we considered two types of similarity measures simultaneously, for calculating tag based similarity. It gave us a tighter bound in selecting friends for recommendation. Pearson's similarity method and Euclidean distance similarity methods were used, i.e., these similarities values are supposed to be higher than a threshold value, alpha, for a friend to be recommended. It produced accurate results, and recommended novel and serendipitous resources.

For example, the percentage improvements in metrics results (for alpha = beta = 0.1) are:

Precision: +25%; Novelty: +28.1%; Serendipity: +27.5%

***Key Terms:*** Social bookmarking, Friend recommendation, Behavioral data mining,

Novelty, Serendipity

1. **INTRODUCTION:**

The growing popularity of IT and communication in recent years strongly encourages the development of e-services in many application domains, such as e-commerce, e-business, e-learning and e-government. In the intervening time, the rapid growth of web information has caused gradually severe information excess problems whereby users are not able to find the exact information to meet their needs in a proper manner by using the current Internet search functions. The amount of information available on the web usually includes a lot of irrelevant contents so that users have to spend a lot of time of their search navigation to search for more interesting contents. One of the possible solutions is **Recommender Systems**. **These** are information filtering systems which provide users with information, they might be interested in. Determined by recommendation algorithms, RS systems provide a way to consumers by selecting products they will probably like and might buy based on their browsing, searching, purchasing, and preferences. Designed to help retailers boost sales, recommenders are a rapid growing business. The supposed “information explosion” in a variety of web-based applications in e-commerce, e-learning and e-tourism has flagged way for recommender systems that provide recommendations for new, movies, books, videos, resources and real estate. These systems incline to provide “personalized” suggestions because they trail each user’s behavior- pages viewed, purchases, and ratings- to come up with recommendations. All such background data of users, and input data, such as item features and user ratings are used to generate a recommendation. These two are then joined using models and algorithms thus, generating a recommendation.

***Social interaction overload problem:***  Social interaction overload is a problem related to the very large amount of users and items that each user can interact with. This leads to the problem of shortage of attention, not allowing users to watch or focus on items that are interesting to her/him. In the last couple of years, the research on recommendation has taken to the development of a new class of systems, named social recommender systems (Ricci et al. 2011). These systems come across the social interaction overload problem, by recommending users or items that the users might be interested in. The filtering is done by means of different metrics. For example, Chang et al. (2014) just highlighted that the social similarity amongst users in a social network can be successfully accepted.

***Serendipity/Over-specialization problem:*** Recommender systems are usually heavily dependent on the user profile as a result of which they the user always gets recommendations for items very comparable to those that she/he already considered and never receives suggestions for unexpected, surprising, and new items. This limit of recommender systems, known in the literature as “serendipity problem” or “over-specialization problem”, deteriorates the user experience by not giving the users the opportunity to explore and discover new items and to improve their knowledge (D). It is known that both the content-based recommender systems (E) and the collaborative filtering approaches (F) are affected by the problem of serendipity. In Shani and Gunawardana (2011), the authors highlight that: – exactness of a recommender system is important but it is not sufficient to evaluate it. Novelty and serendipity are two metrics that are gaining ever increasing importance in the performance evaluation of a recommender system. Novelty refers to the number of recommended items the user did not know about, while serendipity deals with how surprising the successful recommendations are. Serendipity can be thought of as a way to expand recommendations and to allow users to discover new items that they did not know they needed. Lops et al. (2011) highlight that the main difference between a novel recommendation and a serendipitous one is that a recommendation is novel when the user might have autonomously discovered the recommended item, while a recommendation is serendipitous when the user receives a recommendation that she/he might not have discovered.

1. **RELATED WORK:**

This section presents related work on user recommendation in the social sphere. These systems can be classified into several categories, based on the source of data used to build the recommendations:

***2.1)*** Systems based on the analysis of social graphs, which explore the set of people connected to the target user, in order to produce recommendations. These systems recommend either the closest users in the graph, like friends of friends and followers of followers (the “People you may know” feature offered by Facebook (Ratiu 2008) is the most widely known example of this approach), or recommend the users that have the highest probability to be crossed in a random walk of the social graph (the main reference for this type of systems is the “Who to follow” recommendation in Twitter (Gupta et al. 2013)).

***2.2)*** Systems that analyze the interactions of the users with the content of the system (tags, likes, posts, etc.). In order to exploit the user interests, these systems usually build a user profile by giving a structured form to content, thanks to the use of metrics like TFIDF (Term Frequency - Inverse Document Frequency).

***2.3)*** An example of this class of systems is presented in Chen et al. (2009)(G). 3. Hybrid systems, which consider both the social graph and the interactions of the users with the content (an example is represented by Hannon et al. (2010)).

***2.4)*** Systems based on the analysis of social graphs:

Barbieri et al. (2014) (H) recently presented an approach to predict links between users with a stochastic topic model. The model also represents whether a connection is “topical” or “social” and produces an explanation of the type of recommendation produced. Gupta et al. (2013) presented Twitter’s user recommendation service, which is based on shared interests, common connections, and other related factors. The proposed system builds a graph in which the vertices represent the users and the directed edges represent the “follow” relationship; this graph is processed with an open source in-memory graph processing engine, called Cassovary. Finally, recommendations are built by means of a user recommendation algorithm for directed graphs, based on SALSA (Stochastic Approach for Link-Structure Analysis). Liben-Nowell and Kleinberg (2003) studied the user recommendation problem as a link prediction problem. They develop several approaches, based on metrics that analyze the proximity of the nodes in a social network, to infer the probability of new connections among users. Experiments show that the network topology is a good tool to predict future interactions.

***2.5)*** Systems based on the collaborations with the content

In Quercia and Capra (2009) (I), a recommender system based on collocation (i.e., the position of a user) is presented. It uses short-range technologies of mobile phones, to infer the collocation and other correlated information, which are the base for the recommendations. In Brzozowski and Romero (2011), a study about what cues in a user’s profile, behavior, and network are the most effective to recommend people to follow, is presented. In Arru et al. (2013), Arru et al suggests a user recommender system for Twitter, based on signal processing techniques. A signal is determined by the number of times a given theory occurs in a predefined period for a user. The well-thought-out approach defines a pattern-based similarity function among users and makes use of a time dimension in the demonstration of a user profile.

1. **PROPOSED WORK:**

The aim of our work is to build a friend recommender system in the social bookmarking domain, and recommend novel resources to the user, design and develop to face the social interaction and serendipity problems. These resources should generate interest in the user and should be diverse from those that already appear in his user profile. At the same time, the accuracy of the friend recommendations is a fundamental property.

Therefore, this work aims for:

* Recommend friends with high accuracy
* The resources of the recommended friends are novel and serendipitous, *ie,* the resources are not already considered by the target user and are also diverse enough from those available in his user profile

More specifically, we define the problem statement as:

**Problem:** We are given a social bookmarking system, defined as a tuple:

Q = {U, R, T , A, C}, where:

* U , R, and T are sets of users, resources, and tags respectively,
* A is a ternary relation between the sets of users, resources and tags, i.e., A ⊆ U × R × T , whose elements are the tag assignments of a user for a resource;
* C is a binary relation between the users, i.e., C ⊆ U × U , whose elements express the connection between two users.

Our aim is to define a function , which allows the system to derive if, given two users u ∈ U and m ∈ U , there is a connection c ∈ C among them, *ie,* the two users were each other’s friends already in the given dataset.

Our solution is based on *behavioral data mining*, i.e., an analysis of the user interaction with the data, learning from the fields of interests of the already specified data (tagged bookmarks) of the users,  in order to select and recommend the users with the same interests.

There are two types of behaviour, in which the use of tags and resources in a social tagging system is related to . Therefore, using tag based similarity and user based similarity, and aggregating them would blur the information on how similar two users are for each type of behavior. This could lead to potentially inaccurate friend recommendations, and thus result in dissatisfied users, like two users that use the same tags to describe completely distinct and unrelated types of resources. Therefore, our recommendation method recommend friends to users considering both the parameters. First we calculate the tag-based similarities, and then the user-based similarities of each user. After the calculations, our system chooses a set of users to recommend to the target user by selecting:

* the ones that have a tag-based user similarity, calculated by Pearson’s similarity method,  higher than a threshold value α (i.e., ts1 > α);
* the ones that have a tag-based user similarity, calculated by Euclidean distance similarity method,  higher than a threshold value α (i.e., ts2 > α);
* the ones that have a user interest (at least one of the two computed) higher than a threshold value β (i.e., ui > β).

The choice of considering two types of similarity measures for tag based similarity gives us a tighter bound, in recommending the friends, which allows us to recommend a friend with novel and serendipitous resources.

**Novelty and Serendipity**

Novelty and serendipity are two metrics used in the evaluation of recommender systems. Precisely, novelty measures how many recommendations include items that the user did not know about while serendipity measures how surprising the successful recommendations are. Serendipity can be seen as a way to introduce diversification and ranges in the recommendation, in order to allow users to discover new items (that they did not know they were interested in and to improve their knowledge. A novel recommendation might not be serendipitous, while a serendipitous recommendation, by definition, is always novel.

**Definition:** A resource r ∈ R (set of resources) can be considered novel for a user u if and only if r  ∉ R(u), *ie,* it was not already selected by the user. We define the set of novel resources for the user u as as N(u).

When a user is recommended as a friend, we can determine if a resource he bookmarked is serendipitous for the targeted user, by computing how distant it is from the behavior of the user. So, the distance between a recommended resource and the resources already bookmarked by the target user is based on the tags used to classify the resources.

In order to define and find serendipitous resources, we first need to define a set of tags T(r) for a particular resource r, and similarity between two resources sim(r1,r2)

**Definition:**  From the ternary sets (u,r,t), let T (r) = {t ∈ T | ∃(u, r, t) ∈ A} be the set of tags used for a given resource r.

Given the above definition, we use **Jaccard Index** to define the similarity between two resources as:

where:

* T(ri) ⋂ T(rj) = set of common tags used to bookmark both resources ri, rj
* T(ri) ⋃ T(rj)= set containing all the tags used to bookmark either of the resources ri, rj

To understand better the computation of the similarity between two resources sim(ri , rj ), we can represent each resource r as a set of indices or a k-dimensional binary vector t = {t1 , t2 , ..., tk }, where k is the number of tags used in the system and each value ti of the vector is computed as follows:

ti = 1, if ti ∈ T(r); 0 otherwise

We can also define a resource r ∈ R as serendipitous for a user u, as:

**Definition:** For a user u, a resource, ri is serendipitous iff ri ∉ R(u) for all ru ∈ R(u),  sim(ri , ru ) < 0.5. We named B(u) the set of serendipitous resources for the user u.

Since the Jaccard index takes values [0,1], the choice of 0.5 was taken as threshold. So, by considering similarities lower than 0.5, we filter only highly distinct resources.

1. **ALGORITHM :**

**Algorithm 1: Friend Recommender system:**

*Input:*  B = The social bookmarking system

*Output:* The result set containing recommended user pairs

**Algorithm 1** explains the main friend recommender system which takes the social bookmarking system as the input. An empty result set S is initialised. For a given target user ut ∈ U , our system recommends users which have high value of tag-based similarity and a high percentage of common resources. The system follows the following five steps which are described in the five sections below.

***4.1 Tag-based user profiling:***

For given tag assignments of each user, we try to build a user profile which is based on how frequent the tags are used by the user. Given the sets defined in the problem, we first try to consider the tag assignments of a user u as:

The users are profiled according to their use of tags , by taking into account the relative frequency of each tags as follows:

Equation 1 estimates the major role of a tag tj ∈ T in the profile of the user u ∈ U , by defining the relative frequency as the how many times the tag tj was used and dividing by the total number of tag assignments of the user u.

A vector vu = { vu1 , vu2 , vu3 , … , vuk} is used to implement the tag-based user profile where each element vuj is the relative frequency previously defined and k is the number of tags in the system.

***4.2 Resource-based user profiling:***

For a given tag assignment of every user, we try to build a user profile based on what resources are bookmarked by the user. A user profile can be made considering the fact whether he/she has bookmarked a resource (i.e. he/she have an interest in it):

Equation 2 estimates the user interest for the user u in a resource rj with a binary value that is equal to 1 in case rj was bookmarked by and 0 otherwise.

A binary vector is used to implement the resource-based profile for a  user u ∈ U which represents the tagged resources for each user, where n is the number of resources in the system.

***4.3 Tag-based similarity computation***

Based on the tag-based user profile, this first metric is computed among a target user ut and the other users.  Pearson’s correlation and Euclidean distance are used to derive the similarity.

The following algorithm demonstrate how the tag-based similarity is calculated:

**Algorithm 2:** Tag Similarity calculation

*Input: Two users u1 and u2*

*Output: Pair of Tag Similarities - Pearson and Euclidean between two users*

In **Algorithm 2,** the tag based similarity between the two users is calculated, considering the tags used by both the users. **Pearson’s Correlation Coefficient** it is the most effective for the similarity assessment among users. Along with this, **Euclidean Distance Similarity** was used to give it a tighter bound.

For users u and m, let Vu and Vm represent, respectively, the mean of the tag frequencies of user u and user m, and Tum represents the set of tags used by both users

u and m.

                                                 (4)

This metric compares the frequencies of all the tags used by the considered users. The Pearson’s similarity values range from 1.0, which indicates complete similarity, to −1.0, which indicates complete dissimilarity. The Euclidean SImilarity ranges from 0 to 1.

***4.4 User-interest computation***

It is the second computed metric which measures the interest of a user towards another user and it can be depicted by the percentage of common resources among them.

The following algorithm shows how the user similarity is calculated between two users:

**Algorithm 3**: User Interest calculation

*Input: Two user ids u1 and u2*

*Output: User Interest*

User\_Interest(u1,u2):

1. numerator = len( users[u1].bookmarks ⋂ users[u2].bookmarks )
2. denominator = len ( users[u1].bookmarks )
3. ui = numerator/denominator
4. return ui

In **Algorithm 3,** the level of interest of a user u1 in a user u2 is estimated as the number of resources common between both the users, divided by the number of resources bookmarked by the first user u1. This means that the interest of user u2 in user u1 depends on the number of resources bookmarked by user u2 (i.e., when calculating ui(u1, u2), the denominator would be |R(m)|).

***4.5 Recommendations selection***

The use of tags and resources in a social tagging system is related to two different types of behavior in a tagging system. Therefore, to aggregate the tag-based similarity and the interests of  user  into a single score would not bring out the information about the similarity of  two users for each type of behavior. This can lead to very poor results, in this case, friend recommendations. So we need our algorithm to filter the users by taking into account both the types of behavior. Once we compute the tag-based similarity and the interest of user , we choose a user set to recommend to the target user by the considering the ones that have a tag-based user similarity (for both the types of similarity metric) higher than a value called threshold value α (i.e., ts > α) and have a user interest (at least one of the two computed) higher than a threshold value β (i.e., ui >β).

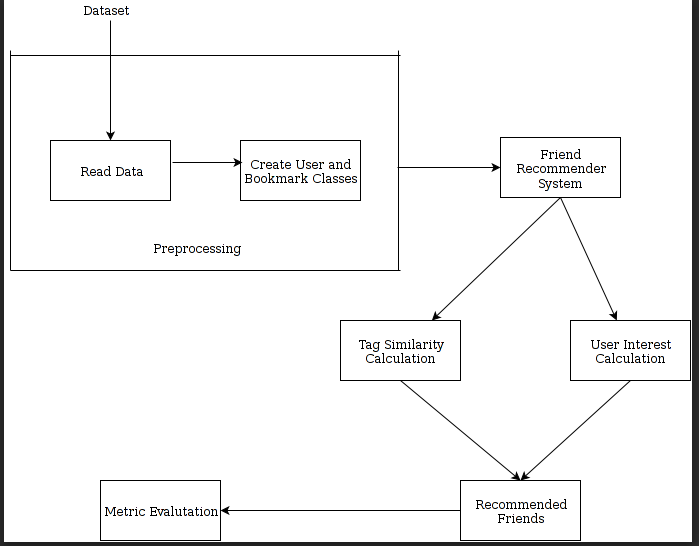
\*Definition 3: For a given target user ut , the candidate set of users to recommend S(ut ) can be defined as:

                (6)

where for both the pearson correlation and euclidean distance similarity.

Equation 6 shows that the  system creates a recommendation between two users if there is a similarity on both types of behavior.

1. **FRAMEWORK:**



1. **EXPERIMENTAL EVALUATIONS**

The dataset used for the evaluation of this approach was the Delicious Dataset which had about 1867 user, 53,000 tags and 68,000 bookmarks.

These are noted as -

* U for the user set
* R for the bookmark set
* T for the tag set
* C for the user relations set
* A for the tags previously assigned

As per the framework, we created user and bookmark classes to preprocess the data and minimize the data-file handling operations which are particularly costly in Python. We trimmed the dataset, removing the cold-start users (with less than 5 bookmarks or tags) to prevent them from wrongly affecting the precision reading.

The metric evaluation is carried out on the set of recommendations produced by the Friend Recommender System algorithm which forms the backbone of this approach.

**6.1 Metrics**

Since the objective of the paper is to measure the serendipity-precision trade-off, we considered the following metrics:

1. Precision
2. User-Satisfaction
3. Serendipity
4. Novelty

**6.1.1 Precision**

In order to measure the accuracy of the system, we evaluate the effectiveness of the recommendations (i.e., which recommended friends are actually friends with the target user), by measuring its precision.

**Definition:** Let R be the total recommendations produced by FRS, such that, R = ∪S(ut), ∀ut ∈ U. This set represents the positive outcomes, that is, the sum of the true positive and the false positive recommendations. Let Rc be the amount of correct recommendations, i.e., Rc ⊆ W = {(u, m) | (u, m) ∈ R ∧(u, m) ∈ C}. So, Rc is the subset of recommendations that exist in the dataset already. So this can be dubbed as ‘true recommendations’. Given the previously defined two sets, R and Rc, we can measure the precision of our recommender system as:

**6.1.2 User Satisfaction**

This metric brings to light another way to look at the accuracy of the recommendations produced by the approach.

Precision calculation acts on the tuple of user-user relation,i.e, the number of user tuples already existing in the dataset.

User satisfaction is more atomic, in that, it takes in account the number of users for which correct recommendations were produced.

So, precision acts on the relational domain whereas user satisfaction acts on the bookmark domain.

**Definition:** Let X ⊆ U be the subset of users for which a recommendation was produced and Y ⊆ U be the subset of users for which a correct recommendation was produced.

User satisfaction can be measure as the ratio of Y and X, i.e.,

(7)

**6.1.3 Novelty**

The recommendations produced by the approach are a form of mining,in that, the friends recommended to a user by the frs are mined for their bookmarks and those are suggested to that user.

The novelty for the recommendations  produced can be computed as follows:

So, the novelty is calculated as ratio of novel resources to recommended resources. Novelty ranges from 0-1, obviously.

**Definition:**  A resource is said to be novel if it is not in the user’s resource list already. That is, A resource r ∈ R can be considered novel for a user u ∈ U iff r   ∈ R(u). We define as N(u) the set of novel resources for the user u.

**6.1.4 Serendipity**

The serendipity for the recommendations is computed as:

(8)

So, serendipity is calculated as the ratio of serendipitous recommendations to total recommendations produced by. It has a range

**Definition:** A resource is said to be serendipitous for a

sim(ru, ri) can be calculated as ratio of number of elements in the intersection of their tags to the number of elements in their union.

**Here we only consider the precise recommendations as non-precise recs can be novel and serendipitous but also worthless**

1. **GRAPHS AND DISCUSSION**

Fig.1 Satisfaction Vs Precision

Here, we analyze the relation of percentage of satisfied users with the precision. So, we have calculated the percentage of satisfied users for each value of precision from the previous equation.

From the above graph, we see that the percentage of satisfied users grows as the precision grows. We can also say that the more the similar the users were, the higher the precision was, and thus the more similar users are, there is a higher chance that the recommended users are satisfied (similar in terms of both tag-based similarity and user-interest similarity).

This is an interesting property of the recommender system shown, and depicts the impact of of precision on individual users.

Fig. 2

Fig. 3

Now, we will analyze the trend of Novelty and Serendipity with Precision. Thus we have calculated the values of novelty and serendipity using their given formulas, for each value of precision.

The results from graph highlight that novelty and serendipity is inversely proportional to the precision of the recommendations made. This means that the number of serendipitous and novel recommendations decrease as the precision of the recommendations made by our recommender system increases. However, the rate of fall of novelty and serendipity is quite low, hence we can say that our recommender system is capable of producing novel and serendipitous recommendations with high precision also.

1. **COMPARISONS**

Fig. 4

In fig 4, we compare the two approaches i.e. the original and the improved approach in terms of the precision of the recommendations produced by them. It is clear from the above representation that the improved approach provides results with higher precision as compared to the original approach. In the improved approach, we intelligently choose the similarity relations to determine the similarity between two users and add an additional bound by using both the Pearson correlation and Euclidean distance similarity measures simultaneously, whereas in the original approach the bound was determined only by the Pearson correlation. Thus the quality of recommendations is improved which is reflected by the increase in the precision values.

The percentage improvements in precision, for different values of alpha and beta are:

a = b = 0.1: +25%

a = b = 0.2: +36.5%

a = b = 0.3: +14%

Fig. 5

The percentage improvements in novelty, for different values of alpha and beta are:

a = b = 0.1: +28.1%

a = b = 0.2: +52.5%

a = b = 0.3: +66%

Fig. 6

In fig 5 and fig 6, we study how the novelty and serendipity vary in the two approaches we used in the paper. Clearly, an increase in the value of both the novelty and serendipity can be seen in the graphical representation in the case of the improved approach as compared to the original approach. Since only the optimal recommendations are produced, both the novelty and serendipity of the recommendations have also improved.

The percentage improvements in serendipity, four different values of alpha and beta are:

a = b = 0.1: +27.5%

a = b = 0.2: +54.3%

a = b = 0.3: +63%

1. **FLOWCHART:**

**https://lh3.googleusercontent.com/bv5hxXB4UBGnCSLdB89uYWd_daMtpQrwC9tUbaiUCZ3tJTG6ufTdCBRNlN5CoK9RqQZcr4WzT3OF67Lw2n1E4jpohtWFT7ENwPaD-Kk5OUhiIH9PxnoQ3T_3Lkx1tY20OaBciP0x**

1. **CONCLUSION AND FUTURE WORK:**

This paper presented a friend recommender system based on a form of behavioral data mining of the users in a social bookmarking system. By considering the regularity of the tags and which resources each user bookmarked, we selected only the users with similar profiles. An analysis of the user behavior in this domain highlighted that the amount of tags and resources in common between two users is limited with respect to the amount of tags and resources bookmarked by each user. The fact that, given a user, a large amount of resources was not considered by the others, allowed us to design and implement a friend recommender system whose intent was to suggest friends with a high accuracy and that allowed users to come across novel and serendipitous bookmarks. Since in the literature it is known that the definition of metrics to evaluate novelty and serendipity in a recommender system is an open research problem, we proposed new metrics that could be applied to our application domain and to the behavioral data mining used to build the recommendations. Experiments evaluated the accuracy in terms of precision and results highlighted the capability of our system to build recommendations with an increasing accuracy as the similarity among users grows. Moreover, we evaluated the capability of our system to suggest friends whose bookmarks are novel and serendipitous and results show that even when a system achieves a high accuracy, it is still capable of producing novel and serendipitous recommendations. Future work will be focused on adding a graph mining component to our system, in order to be able to produce recommendations also in the previously highlighted cases, in which users cannot receive recommendations.

To conclude, we have improved upon a friend recommender system based on the data inferred from the user interactions via bookmarks and tags. This ensures that although the recommendations are serendipitous, they are not irrelevant as the precision serendipity tradeoff has been improved a lot. Though this approach is limited for cold start users because this analysis takes in account the tag frequencies to calculate similarity. We've not taken the whole of the resources for a particular user, the recommender system paints a better picture of the user's preferences.  
Since the subject of measuring serendipity is very subjective and open to interpretation, we've used new metrics and cross-metric measures to get an accurate picture of the effectiveness of our approach. As can be seen from the evaluation section, the recommender system provides serendipitous recommendations without sacrificing too much precision and hence handles the relevance vs coverage conundrum,  
and serendipitous recommendations. This system can be extended to a graph like approach to compensate for the weakness of this method regarding cold start users. It can be used in a groundbreaking social application to give non-specialized recommendations based on interests rather than physical proximity of users in the graphical-social paradigm.

**11 ) REFERENCES:**

1. <http://www.delicious.com/>
2. <http://www.citeulike.org/>
3. <http://www.flickr.com/>
4. <http://link.springer.com/chapter/10.1007%2F978-0-387-85820-3_8> :

Shani and Gunawardana 2011

1. Lops et al. 2011
2. Ziegler et al. 2005
3. <http://onlinelibrary.wiley.com/doi/10.1111/j.1360-0443.2009.02750.x/abstract/>
4. <https://www.researchgate.net/publication/271134224_Barbieri_et_al_2014/>
5. <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5694070&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D5694070/>