Association for Information Systems
AIS Electronic Library (AISeL)

MCIS 2010 Proceedings

Mediterranean Conference on Information Systems (MCIS)

9-2010

SIMILARITY ENHACEMENT IN TIME-AWARE RECOMMENDER SYSTEMS

Roee Anuar
Tel-Aviv University, Israel, roeeanua@post.tau.ac.il

Yossi Buckchin
Tel-Aviv University, Israel, bukchin@eng.tau.ac.il

Oded Maimon
Tel-Aviv University, Israel, maimon@eng.tau.ac.il

SIMILARITY ENHACEMENT IN TIME-AWARE RECOMMENDER SYSTEMS

Anuar, Roee, Tel-Aviv University, roeeanua@post.tau.ac.il; Buckchin, Yossi, Tel Aviv University, bukchin@eng.tau.ac.il; Maimon, Oded, Tel Aviv University, maimon@eng.tau.ac.il;

Abstract

Time-aware recommender systems (TARS) are systems that take into account a time factor - the age of the user data. There are three approaches for using a time factor: (1) the user data may be given different weights by their age, (2) it may be treated as a step in a biological process and (3) it may be compared in different time frames to find a significant pattern. This research deals with the latter approach.

When dividing the data into several time frames, matching users becomes more difficult - similarity between users that was once identified in the total time frame may disappear when trying to match between them in smaller time frames.

The user matching problem is largely affected by the sparsity problem, which is well known in the recommender system literature. Sparsity occurs where the actual interactions between users and data items is much smaller in comparison to the entire collection of possible interactions. The sparsity grows as the data is split into several time frames for comparison. As sparsity grows, matching similar users in different time frames becomes harder, increasing the need for finding relevant neighboring users.

Our research suggests a flexible solution for dealing with the similarity limitation of current methods. To overcome the similarity problem, we suggest dividing items into multiple features. Using these features we extract several user interests, which can be compared among users. This comparison results in more user matches than in current TARS.

Keywords: Recommender Systems, Sparsity, Similarity, Time, Patterns.

1 INTRODUCTION

The term "Recommender Systems" (RS) was introduced by Resnick and Varian (1997), describing an application of information filtering (see Figure 1), used to suggest data items to people that are likely to be interested in them. The new term was introduced as a substitute to “Collaborative Filtering” (CF) in order to signify the difference between collaboration, which refers to the method, and recommendation, which refers to the result of the process. According to Montaner et al. (2003), there are three main filtering technique approaches to issue a recommendation: (1) Personalized (Demography-Based), suggesting items that correspond with the user's profile, (2) Item driven (Content-Based), suggesting similar items to the items that the user was previously interested in, (3) Collaborative, suggesting items used by other users with similar interests as the current user .The three methods may be used together and form a hybrid recommender system.

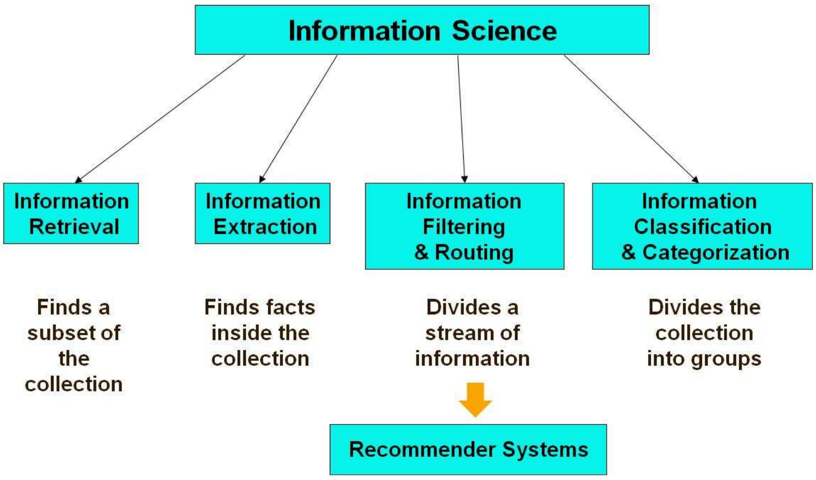

Figure 1. Recommender Systems domain

Time-aware recommender systems, as explained later in this article, offer an extra perspective when analyzing the users’ behavior. These rather new approaches, take the time factor into consideration when issuing a recommendation. In this paper we focus on a specific set of time-aware recommender systems, which use sequential patterns over time.

The rest of the paper is organized as follows: Section 2 reviews several systems, and suggests a taxonomy of recommender systems, section 3 presents the methodology of extracting the interests, section 4 presents a comparison of user matching by similarity and section 5 summarizes the article.

2 LITERATURE REVIEW

Several literature reviews have been conducted during the last few years, concerning the development, techniques, implementation and classification of recommender systems. Some of these reviews suggest a taxonomy-based classification of existing recommender systems. Schafer et al. (2001) reviews existing e-commerce systems using recommendations, and makes the distinction between different RS based on several dimensions, including the input, approach and output of the RS. While Schafer et al. (2001) suggests a taxonomy solely for recommendations based on preference data (known interests), Montaner et al. (2003) includes in his taxonomy recommendations based on other data, including the user profile adaptation technique.

Montaner's dimensions are more inclusive than Schafer's, but are not independent as well. Montaner adds an important dimension of recommender systems – the adaptation of the user's profile that enables dealing with drift in interests.

To view a complete, yet independent taxonomy by dimension, we present a hierarchical view of the taxonomy (see Figure 2). The taxonomy is closer to Schafer et al's presentation, but includes more than preference data alone. The taxonomy nodes hold a reference to an example of a paper or site using the method indicated in the node.

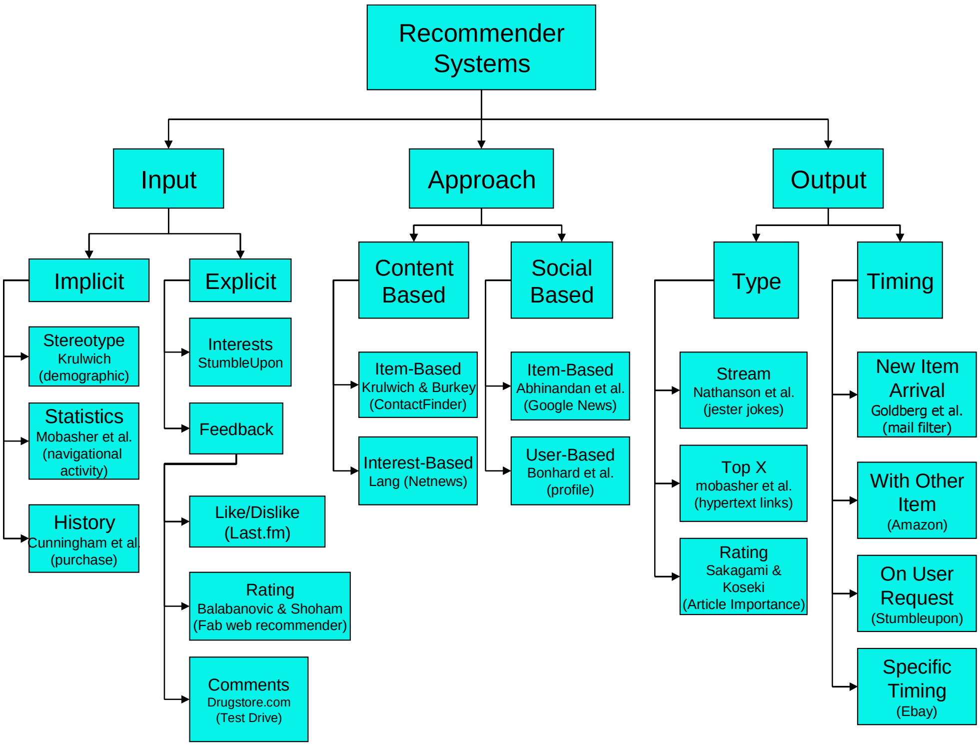

Figure 2. *Recommender Systems taxonomy*

Taxonomy details:

- Inputs – The data used for creating the recommendation. This data divides into two groups:
  - Implicit – In our perspective, implicit inputs include all the inputs that are not supplied directly by the user:
    - Stereotype – This is an assumption (classification) made on the user, being part of a defined group.
    - History – This method takes as input, the history of items that the user interacted with
    - Statistics – Statistics are used to decipher user's interests, based on his behavior. These statistics may sometimes indicate interests that even the user is not aware of.
  - Explicit – These inputs include all inputs that are supplied explicitly by the user:
    - Interests – An explicit list of interests that the user indicates in his profile.
    - Feedback – An explicit opinion given by the user in one of the following forms:
      - Rating – A score given by the user to the item.
      - Like/Dislike –A simple kind of rating and may be viewed as a 1/-1 rating for an item.
      - Comments – This feedback is the least used, since it requires a sentiment understanding of the written feedback (Pang et al. 2002).
- Approach – Explains the general method used when computing the recommendation. It does not explain the exact technique used, since these techniques may be used in either approach. For an

extensive classification of techniques, refer to Hanani et al. (2001) and Canfora and Cerulo (2004). Systems that use both content-based and social-based approaches are called hybrid RS.

- Content-Based – Relies on user-item matching, to identify the items that may interest the user:
  - Item-Based – In the content-based context, the item-based recommendation relies on the similarity between items.
  - Interest-Based – In the content-based context, the interests of the user are considered when finding similar users, instead of the items themselves.
- Social Based – Relies on user-user matching, to identify users that share the same interests, and may imply on new interests to one another.
  - Item-Based – In the social-based context, the item-based recommendation relies on the similarity between users, based items they've all been interested in.
  - User-Based – In the social-based context, similarity between user profiles is used to obtain the recommendation.

- Output – This dimension considers how the recommendation is given and when:
  - Type – The type of recommendation explains how the recommendation is delivered:
    - Stream – The user receives one item at a time, as in a stream of information.
    - Top X – The user is presented with the top X items that best match his taste.
    - Rating – An approximation of the rating that the user would have given the item is computed and presented.
  - Timing –When is the recommendation delivered to the user?
    - New Item Arrival – When a new item (or a group of items) is introduced into the collection, the recommender system decides if to recommend this item to a user or not.
    - With other Item – Recommended item is added to items that the user is observing.
    - On User Request – The user requests an immediate recommendation.
    - Specific Timing – A daily/weekly/monthly… recommendation.

Next, we elaborate on the adaptability aspect from Montaner's work and extend it to reflect time-aware and unaware systems.

2.1 Time-aware recommender systems

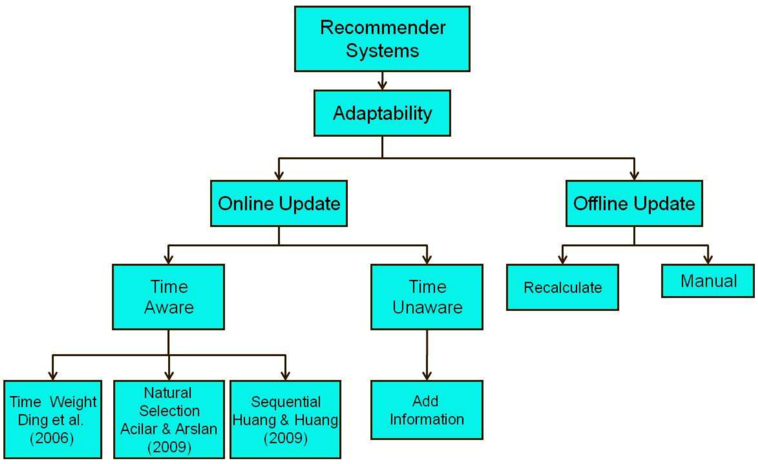

Figure 3. Recommender Systems taxonomy

Figure 3 presents the taxonomy breakdown for adaptability in recommender systems.

The adaptability aspect in recommender systems may be summarized by these approaches:

- Offline Update – In this approach, the adaptability is not an integral part of the system. The system may be updated in two ways:

    - Manual – Manual interference is introduced to the system, to change the recommendation. For example, users may be required to update their interests every once in a while. As the users updates their interests, new recommendations may be issued for them based on their new interests.
    - Recalculate – In this approach, the recommendation model is recalculated every given time, adding new and discarding old information.
- Online update – In this approach, the adaptability is an integral part of the system.
    - Time Unaware – although these systems consider new data as an integral part of their model, they do not treat old and new data differently. We have observed a single approach in general for this behavior:
        - Adding new information – In this approach, new data is added to the old one, thus adding data on the customer. There is no reference to the age of the data, and all data from all time periods is accounted in the same manner when issuing a recommendation.
    - Time Aware – In this approach, the age of the data is considered when issuing a recommendation. This approach includes three methods of adaptation:
        - Time Weight – A weight is attached to each sample to increase or decrease its influence on the recommendation. With time, older observations may receive less influence in issuing a new recommendation, until they are no longer used. The moving average is a specific implementation of this approach. Ding and Li (2005) and Ding et al. (2006) use this approach to ensure that newer items will receive more weight when issuing a recommendation.
        - Natural selection – Biological based algorithms are used to evolve, mutate and reproduce the best recommendation. Other, less productive recommendations disappear as time goes by. Both Cayzer and Aickelin (2002) and Acilar and Arslan (2009) suggest a collaborative filtering method based on artificial immune network, to group users.
        - Sequential – Recommendation is issued based on patterns in time. For example, Huang and Huang (2009) assign customers into groups, according to their purchase sequence.

2.2 Sequential analysis

In sequential analysis, the similarity between users is based on findings matching items in time patterns among them.

Min and Han (2005) suggest a recommender system based on a time variant pattern. In their work, they map the change in user behavior, by measuring the distance between the clusters a user belonged to in different time points. Afterwards, they match similar users by matching the users' ratings of items in different time points and predict the user's ratings for other items.

Cho et al. (2005) followed the customer purchase sequence over several time periods. In the first step each transaction is allocated into a product group to map the purchase behavior of customers. In the second phase, similar purchase patterns are grouped into clusters. In the third step, based on the change in the group number a user belonged to in each time period, new dynamic profiles are built. From these dynamic profiles, sequential rules are discovered. These rules are later used to predict the next group for a customer. Based on the predicted groups, a recommendation is issued.

Huang and Huang (2009), assign customers into groups, according to their purchase sequence. To solve the sparsity problem created when trying to match specific sequences, they also use the hierarchical product taxonomy, and build the purchase sequence based on the product group instead of the product itself. After building the customer groups, sequential pattern mining is used to find the purchase sequence for each customer and each group. To issue a recommendation, they match a customer pattern versus all purchase patterns of groups, and find the most probable (supported) product categories that the user will buy from. Based on these groups, they find the most probable (frequently bought) items that the user will purchase and issue a recommendation.

As seen above, all sequential recommender systems aggregate the items into item categories or clusters to solve the sparsity problem that arises when looking at user activities over different time frames. These single aggregations may lose valuable data about the specific item that the user was interested in, and as such have many limitations on the recommendation.

2.3 Previous work

The interest extraction approach presented here is based upon a method first used for rating scientific papers in the NHECD project (Maimon et al. 2009, Anuar et al. 2010). In short, the rating method used incoming citations and citation "strength" to calculate normalized rating scores for each paper in the NHECD corpus. Using the same data structures, we extend the rating capabilities so that using simple SQL statements, the user interests may be extracted. This work differs from our previous one (Anuar et al. 2010) in our direct reference to the user similarity problem that emerges when splitting the data into several time frames, instead of the sparsity problem.

3 DEFINITIONS AND METHODS

Let us consider a recommender system in which each user u, and each item i, have several attributes. The interactions between users and items are recorded by saving the user id, the item id, and the time of the interaction between them. In this context, the users do not rate their interaction with the items.

To enhance the similarity, we suggest using the entire available data for users and items, instead of a single aggregation as mentioned earlier. Using all of the available features of items and users, our goal is to find the users' interests. Comparing between interests instead of items will lead to enhanced similarity in different time frames.

3.1 General definitions

To solve the problem, we use a relational DB approach using the following definitions:

$\mathrm{SVIA_i} \subseteq \{\mathrm{IA_t}: \mathrm{VIA_{t1}}, \mathrm{VIA_{t2}} \dots, \mathrm{VIA_{tk_t}}\} \forall \mathrm{t} = 1..\mathrm{N}$ is a subset of N possible attributes of item i, where $\mathrm{IA_t}$ is an attribute t of items in the system and $\mathrm{VIA_{tw}}$ is the $\mathrm{w'}$th possible value of the item attribute t. For example: $\mathrm{SVIA_{124}}$= {Name: Gold Particles Toxicology; Keyword: Toxicology, Nano-particles; Author: John Smith, Jane White; Referenced Articles: 1713, 19054, 7242}, this is the representation of an article written by John Smith and Jane white, in the topic of gold particles toxicology, it references paper 1713, paper 19054 and paper 724. A DB representation of this example is described on Table 1.

INDEX	ATTRIBUTE	VALUE
124	Name	Gold Particles Toxicology
124	Keyword	Toxicology
124	Keyword	Nano-particles
124	Author	John Smith
124	Author	Jane White
124	Referenced Articles	1713
124	Referenced Articles	19054
124	Referenced Articles	7242

Table 1. Item attribute values set in DB representation

An interaction record is added at the time the user interacted with an item:
$\mathrm{SVNA_j} \subseteq \{\mathrm{u}; \mathrm{i}; \mathrm{T}\} \forall \mathrm{t} = 1..\mathrm{Q}$ is the interaction j for user u with item i at a specific time T.

Example: $\{131; 124; 17/12/2009\}$ = The interaction of user 131 with item 124 on 17/12/2009.

3.2 Extracting the users' interests

To extract the user interests in a specific time period T_l, we will first find the common behavior of all users. We first perform a Cartesian product (DB inner join) of the item and the interactions in this time periods, resulting in $CB_0(T_l)$ - the interaction breakdown. This breakdown is defined as:

$$CB_0(T_l) = \{SVNA_i \bowtie SVIA_j \; \forall T \text{ in } T_l\}$$

Using the interaction breakdown, we may count he number of times, each attribute was observed in each attribute value. This results in an aggregation of the interaction breakdown, $CB_{1t}(T_l)$. The attribute value summation is done for each item attribute and item attribute-value:

$$CB_{1t}(T_l) =_{IA,VIA} \mathcal{G}_{count(1)} \left(CB_0(T_l)\right)$$

To calculate the final common behavior, we divide the aggregated table $CB_{1t}(T_l)$, by the number of interaction in T_l. This will result in the common behavior - $CB(T_l)$:

$$CB(T_l) = \frac{CB_{1t}(T_l)}{\mathcal{G}_{count(1)} \left(SVNA_i \; \forall T \; in \; T_l\right)}$$

Repeating the same steps above for the interactions of a specific user, we receive $UB_u(T_l)$ - the user u behavior in T_l. Subtracting the common behavior from the user behavior, we receive the user interest behavior $UIB_u(T_l)$:

$$UIB_u(T_l) = UB_u(T_l) - CB(T_l)$$

3.3 Interest extraction example

Given the item set and interaction set seen in Figure 4, we'll follow an example to extract the user interests.

Items

Item Id	Item Attribute	Item Attribute Value
1	Keyword	Nanotoxicology
1	Keyword	Taxonomy
1	Author	David Norton
2	Keyword	Nanotoxicology
2	Keyword	Gold
2	Author	John Smith
2	Author	Jane white
3	Keyword	Taxonomy
3	Author	John Smith
3	Author	Tom Flank

Interactions

User Id	Item Id	Date
1	1	12/3/2009
1	2	12/3/2009
2	2	12/4/2009

Figure 4. Table Representation of Model

The first step is to join the interactions and items and recieve an interaction breakdown, as seen in Figure 5(a).

Interaction Breakdown

User Id	Item Id	Item Attribute	Item Attribute Value
1	1	Keyword	Nanotoxicology
1	1	Keyword	Taxonomy
1	1	Author	David Norton
1	2	Keyword	Nanotoxicology
1	2	Keyword	Gold
1	2	Author	John Smith
1	2	Author	Jane white
2	2	Keyword	Nanotoxicology
2	2	Keyword	Gold
2	2	Author	John Smith
2	2	Author	Jane white

(a)

Attribute Value Summation

Item Attribute	Item Attribute Value	Occurences	Normalized
Keyword	Nanotoxicology	3	1
Keyword	Taxonomy	1	0.333
Keyword	Gold	2	0.667
Author	David Norton	1	0.333
Author	John Smith	2	0.667
Author	Jane white	2	0.667

(b)

Figure 5. Interaction breakdown, Attribute value summation & Common behavior

Next we use the interaction breakdown and aggregate the data by item attribute and item attribute-value, resulting in the attribute value summation (see "occurrences" column in Figure 5(b)).

Next we divide the summation over the number of occurrences, for each observed item attribute value. (see "normalized" column in Figure 5(b)).

We repeat the same steps to build user 1's behavior and calculate his interest behavior. First we select only the interactions of user 1, and by joining them with the item attributes, we receive the interaction breakdown for user 1 (see Figure 6. User 1 *Interaction Breakdown & Interests Behavior*. Next we repeat the summation over all item attribute values, for user 1 interaction breakdown alone (see "occurrences" column in Figure 6. User 1 *Interaction Breakdown & Interests Behavior*. The user behavior is calculated by dividing the item attribute value summation by the number of interactions of user 1 (see "normalized" column in Figure 6. User 1 *Interaction Breakdown & Interests Behavior*. Subtracting the common behavior from the user behavior results in the user interest behavior, as seen in Figure 6. User *1 Interaction Breakdown & Interests Behavior*.

User 1 Interation Breakdown

User Id	Item Id	Item Attribute	Item Attribute Value
1	1	Keyword	Nanotoxicology
1	1	Keyword	Taxonomy
1	1	Author	David Norton
1	2	Keyword	Nanotoxicology
1	2	Keyword	Gold
1	2	Author	John Smith
1	2	Author	Jane white

(a)

UB

Item Attribute	Item Attribute Value	Occurences	Normalized
Keyword	Nanotoxicology	2	1
Keyword	Taxonomy	1	0.5
Keyword	Gold	1	0.5
Author	David Norton	1	0.5
Author	John Smith	1	0.5
Author	Jane white	1	0.5

(b)

UIB

Item Attribute	Item Attribute Value	UIB
Keyword	Nanotoxicology	0
Keyword	Taxonomy	0.167
Keyword	Gold	-0.167
Author	David Norton	0.167
Author	John Smith	-0.167
Author	Jane white	-0.167

(c)

Figure 6. User 1 Interaction Breakdown & Interests Behavior

As seen from the figure, after considering the behavior of the entire population, user 1 shows an interest in articles with the "Taxonomy" keyword and articles with "David Norton" as author. The values in the UIB will always vary from -1 and 1. A value of -1 suggests no interest in the specific attribute value, and a value of 1 suggests a total interest in it. In this example we do not look into the interactions between interests. Using the UIB, we can find similar users by matching similar interests by each item attribute. In this example we may find similar users by their interests in specific keywords, or by specific authors.

4 EXPERIMENTAL RESULTS

Regularly in recommender systems, the similarity is calculated by using cosine similarity between the users (Adomavicius et al. 2005). The cosine similarity measure is used to compare between two user vectors, resulting in a similarity score between -1 and 1. We use cosine similarity on two approaches (1) between movies each user have seen, and (2) between each user's extracted interests. Our similarity enhancement algorithm was tested on the MovieLens dataset. In our context, the matrix considered is the matrix of movies, seen by each user in the repository. Since each user viewed only a small fraction of the movies, this matrix is sparsely populated.

On our first comparison, we used the full available data (as a one time frame) to compare between the approaches. First, we used our algorithm to extract the UIB based on the available genres for each movie. Next, we measure the cosine similarity for each two different users. Next, for each user we calculate the similarity for every other customer. Next, for each user and similarity threshold, we count how many similar users may be found. By incrementally decreasing the distance threshold, we find the number of neighbouring users as a function of the similarity threshold them. We repeat the same process for the user-movie instead of the user-interest vectors.

Figure 7 presents the results of the average number of users that fall within different levels of cosine similarity threshold. As seen from the figure, for the two approaches, the number of similar users grows as we decrease the threshold. However, when the similarity threshold is relatively high, the number of similar users is larger when comparing by interest than by movies. When the similarity threshold drops to 0.2 and below, the number of similar user is larger when comparing by movies than by interests. This suggests that users clustered by interests are located in tighter groups.

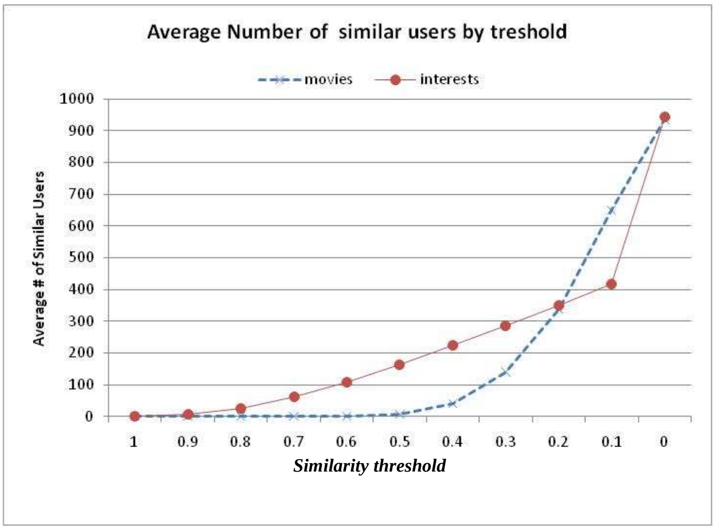

Figure 7 Average number of neighbours by distance, in one time frame.

On our second comparison, we used only one half of the data. This data includes all the interactions between users and movies that occurred before the median interaction occurred, resulting in a smaller time frame. We repeated the same test as explained above. Figure 8 presents the results when comparing between users in a smaller time frame.

As seen from both tests, using the interest vectors to find similar users, results in finding more neighbouring users than when using the movie vector. This implies that when trying to find neighbouring users, using the user-interest matrix will lead to more results than using the user-movie matrix.

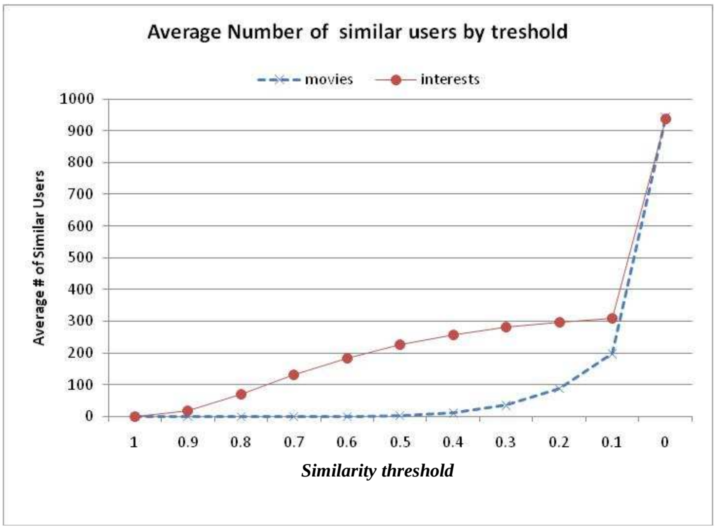

Figure 8 Average number of neighbours by distance, in a half time frame.

5 CONCLUSIONS AND FURTHER RESEARCH

This paper aims at two goals. First, we survey and organize TARS into a taxonomy. Moreover, our taxonomy of RS presents independent dimensions of RS properties and is the only visual breakdown we have encountered in the literature. Second, we suggest a new approach for finding similar users, based on user interests. The extraction of interests offers a strong tool when dealing with sparse data for finding user similarities.

As seen from the experimental results, the usage of user interests results in more user matches than traditional similarity matching. By reviewing the number of neighbouring users by a similarity threshold (as seen from figure 7), it seems that when using the interest vector, there are tighter clusters of users.

Although we show that more user matches may be found, it is not guaranteed that matching more users will yield better recommendations. To solve this problem, our future research will focus on two topics: (1) Usage of the user behavior vector as an estimator of the probability for a user to like a specific feature, and (2) Measurement of recommendation in a scale of entropy. Using an entropy measure, we may find the certainty of our recommendation, and compare it to traditional measures as RMSE.

References

Abhinandan, D., Mayur, D. and Ashutosh G. (2007) Google News Personalization: Scalable Online Collaborative Filtering. WWW 2007 / Track: Industrial Practice and Experience. May.

Acilar, A. M., Arslan, A. (2009). A collaborative filtering method based on artificial immune network. Expert Systems with Applications, Vol. 36, Pages: 8324- 8332.

Adomavicius, G. and Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions On Knowledge And Data Engineering. Vol. 17(6).

Anuar, R., Browarnik, A., Bukchin, Y., Korenstein, R., Maimon, O., Shapira, B.(2010) Rating in NHECD. In Nano Impact Net Conference, Lausanne, March 10th, 2010.

Anuar, R., Bukchin, Y., Maimon, O. (2010). Sparsity Reduction in Time Aware Recommender systems. In 16th Industrial Engineering and Management Conference, March 24th 2010.

Balabanović, M. and Shoham, Y. (1997) Fab: content-based, collaborative recommendation. Communications of the ACM. Mar, Vol. 40 (3), Pages: 66-72

Bonhard, P., Harries, C., McCarthy, J. and Sasse, M. A. (2006) Accounting for Taste: Using Profile Similarity to Improve Recommender Systems. CHI 2006 Proceedings • Social Computing 2. Apr.

Canfora, G. and Cerulo, L. (2004) A Taxonomy of Information Retrieval Models and Tools. Journal of Computing and Information Technology - CIT 12, Pages: 175–194.

Cayzer, S., Aickelin, U. (2002). A Recommender System based on the Immune Network. Proceedings of the 2002 Congress on Evolutionary Computation, CEC '02. Vol. 1, Pages: 807-812.

Cho, Y. B., Cho, Y. H., Kim, S. H. (2005). Mining changes in customer buying behavior for collaborative recommendations. Expert Systems with Applications, Vol. 28, Pages: 359–369

Cunningham P., Bergmann R., Schmitt S., Traphöner R., Breen S., Smyth B., Traphöner R., Breen S. and Smyth B. (2001) WEBSELL: Intelligent Sales Assistants for the World Wide Web. Proceedings of the Workshop Program at the 4th International Conference on Case-Based Reasoning, ICCBR-2001 Workshop 3: Case-Based Reasoning in Electronic Commerce. Pages: 104-109.

Ding, Y., Li, X. Time Weight Collaborative Filtering. (2005). Conference on Information and Knowledge Management, Proceedings of the 14th ACM international conference on Information and knowledge management, Pages: 485 – 492.

Ding, Y., Li, X., Orlowska, M. E. (2006). ACM International Conference Proceeding Series; Vol. 170. Proceedings of the 17th Australasian Database Conference, Vol. 49, Pages: 99 – 107.

Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992) Using collaborative filtering to weave an information tapestry. Communications of the ACM. Dec, Vol. 35(12)

Hanani, U., Shapira, B. and Shoval P. (2001) Information Filtering: Overview of Issues, Research and Systems. User Modeling and User-Adapted Interaction. Vol. 11, Pages 203-259.

Helft, M. (2007) A Way to Find Your Corner of the Internet Sky. New York Times. Oct

Huang, C. L., Huang, W. L. (2009). Handling sequential pattern decay: Developing a two-stage collaborative recommender system. Electronic Commerce Research and Applications, Vol. 8, Pages:117–129.

Krulwich, B.(1997) Lifestyle finder: Intelligent user profiling using large-scale demographic data. AI Magazine. Vol. 18 (2), Pages: 37-46.

Maimon, O., Browarnik, A., Saruk, E., Korenstein, R., Rossi, F., Krop, H., Broekhuizen, P., Wittig, H., Lemor, A. M. (2009) Creation of a critical and commented database on the health, safety and environmental impact of nanoparticles – Challenges and objectives. On EuroNanoForum 2009 in Prague, 2-5 June.

Min, S. H., Han, I. (2005). Detection of the customer time-variant pattern for improving recommender systems. Expert Systems with Applications, Vol. 28, Pages: 189–199.

Mobasher, B., Cooley R. and Srivastava J. (2000) Automatic personalization based on Web usage mining, Communications of the ACM. Vol 43(8), Pages: 142-151

Mobasher B., Dai H., Luo T. and Nakagawa M. (2001) Effective personalization based on association rule discovery from web usage data. Workshop on Web Information and Data Management, Proceedings of the 3rd international workshop on Web information and data management.

Montaner, M. Lopez, B. de la Rosa, JL. (2003). A taxonomy of recommender agents on the Internet. Artificial Intelligence Review, vol. 19(4), Pages: 285-330

Pang, B. Lee, L. Vaithyanathan S. (2002) Thumbs up? Sentiment classification using machine learning techniques. Proceedings of the ACL-02 conference on Empirical methods in natural language processing. Vol. 10, Pages 79-86.

Resnick, P. and Varian, H. Recommender Systems. (1997). Introduction to special section of Communications of the ACM, vol. 40(3).

Schafer, J. B., Konstan, J. and Riedl, J. (2001). Electronic Commerce Recommender Applications. Journal of Data Mining and Knowledge Discovery, Vol. 5, Pages: 115–152.