Identifying Interests of Web Users for Effective Recommendations

Rakesh Rawat, Richi Nayak, Yuefeng Li

***Abstract*—Search log data is multi dimensional data consisting of number of searches of multiple users with many searched parameters. This data can be used to identify a user's interest in an item or object being searched. Identifying highest interests of a Web user from his search log data is a complex process. Based on a user's previous searches, most recommendation methods employ two-dimensional models to find relevant items. Such items are then recommended to a user. Two-dimensional data models, when used to mine knowledge from such multi dimensional data may not be able to give good mappings of user and his searches. The major problem with such models is that they are unable to find the latent relationships that exist between different searched dimensions. In this research work, we utilize tensors to model the various searches made by a user. Such high dimensional data model is then used to extract the relationship between various dimensions, and find the prominent searched components. To achieve this, we have used popular tensor decomposition methods like PARAFAC, Tucker and HOSVD. All experiments and evaluation is done on real datasets, which clearly show the effectiveness of tensor models in finding prominent searched components in comparison to other widely used two-dimensional data models. Such top rated searched components are then given as recommendation to users.**

***Index Terms*— Decomposition, Recommendation, Tensor, Web Log Data.**

I. Introduction

When searching online, a user has fixed goals, which are fulfilled, if the concerned website returns good quality and relevant information as needed and liked by the user. Since, most of the user's make erratic and random searches, identifying individual user preferences from such log data becomes difficult. Identification of individual user interests is crucial for any Web based personalization [1] system. However, identification of user behaviour and interests is a complex process. It involves various co-relations between searched parameters. Recognizing such interests of users can solve the information overload problem by recommending items/objects that match highly with a user's interests. When using explicit data (registration data, data in subscription forms or ratings given to items by user) for modelling user behaviour, the biggest problem is that such data may be too old to identify a user's current needs as a user's interest may have changed since such information was last provided by the user. On the other hand, implicit data obtained from server logs is more reliable, as it reflects a user's actual needs expressed by queries. However, the problem with such data is its multi dimensionality. Each user's data consists of many searched query parameters, date-time, operating system used, browser used and various other details. Finding relationships between multiple searches and searched query components is a complex process. Traditional methods use two-dimensional data modelling techniques to mine information from such datasets consisting of users-items relationships [1]. The other noteworthy factor is that interest vectors would be compared using a distance measure such as Euclidean distance or cosine similarity, however, previous research [2] has shown that distance measures used for clustering or comparisons may reflect strange properties in high dimensional space and might not be as useful as they seem.

In this research, we propose to use TSM (Tensor Space Models) which are higher dimensional data modelling methods, to effectively mine user's information, consisting of user's highest interests in each dimension. Once user information is stored, this information is then used for making recommendations to a user. Empirical evaluations have been done on real search log data from a car sales website and employed methods are compared with traditional vector and matrix methods. Results clearly outline the effectiveness of such methods in identifying a user's behaviour more accurately.

II. Related work

Multi dimensional data is becoming a norm in many scientific and engineering fields. Applicability of tensors in fields where the data have multi dimensional properties have become popular due to its multi dimensional data modeling abilities and inferences capabilities [3],[4]. The use of tensor modelling in data mining and Web mining applications is gaining momentum. In comparison, tensors have been used extensively in chemometrics [4]. Some prominent work related to Web data mining is discussed briefly in this paper. One such methodology, as proposed by [5], uses search from click stream data to personalize Web search. The click through data of Web users is converted into a 3rd order tensor

Manuscript received November 30, 2010. This research has been funded by CRC (Co-operative Research Centre), Australia and Queensland University of Technology, Brisbane, Australia under the CRC Smart Services Personalization Project 2009-10.

Rakesh Rawat is with Faculty of Science and Technology, Queensland University of Technology, Brisbane, Australia. (Corresponding Author. Phone: +61-07-31389339, e-mail: r.rawat@qut.edu.au).

Richi Nayak is with Faculty of Science and Technology, Queensland University of Technology, Brisbane, Australia. (e-mail: r.nayak@qut.edu.au).

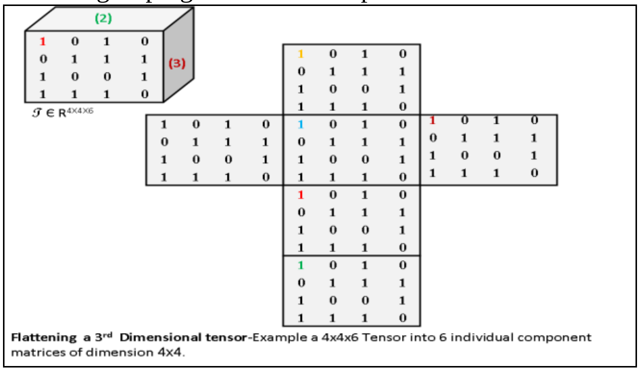
Yuefeng Li is with Faculty of Science and Technology, Queensland University of Technology, Brisbane, Australia. (e-mail: y2.li@qut.edu.au).

consisting of users, query and pages as the three dimensions, and a tensor decomposition approach based on generalization of the matrix SVD (Singular Value Decomposition) is proposed to decompose such tensors. In another work [6], that applies Tucker3 decomposition to analyze user behavior in chat rooms has been proposed. In this work, the three dimensional data from chartroom activities such as users, keywords and time windows is analyzed. The researchers found that tensor decomposition is appropriate for such data due to the number of components in each dimension. Additionally, using tensor decomposition rather than two-dimensional methods the researchers found that interaction pattern and the latent relationships that exists between various dimensions in such kind of datasets is advantageous to mine using tensors. Recently, a probabilistic latent variable model called as pTucker was proposed by [7]. It has the ability to learn rich dependency structure from partially observed multi way array data. Here the core tensor is integrated out and missing values are handled in a principled manner. TSM using HOSVD (Higher Order Singular Value Decomposition) for dimension reduction, have been used for recommending personalized music [8] and tags [9]. Researchers [10] have used TSM based tag recommendation model which uses tensor factors by multiplying the three features matrices with core matrix each consisting of user, items and tags. A recent work of TSM clustering used for clustering similar blogs is proposed by [11]. Unlike these previously discussed methods, we have used tensors to model individual user behavior consisting of more than three dimensions, and then have used this model consisting of user's top rated interests for making recommendations. To measure the quality of recommendations made, efficiency of our methodology is tested with real searches (after the model is created) made by users. The recommendations made by each method TSM, vector and matrix methods are compared to the actual searches made by the users. If the top *n* (here we have taken *n* as top 3, top 5, top 10 and top 15 recommendations) recommendations are similar to the actual searches made by a user the recommendations is considered to be accurate.

III. Proposed Method

There are many methods and representation styles used for representing tensors, however, we have followed the conventional notation that is adopted by many previous researchers like [4], [5], [6]. Scalars are denoted by lowercase letters, e.g., c. All vectors are represented by boldface lowercase letters e.g., $\boldsymbol{v}$. The i^{th} entry of $\boldsymbol{v}$ is denoted by $\boldsymbol{v}_i$. Matrices are denoted by boldface capital letters, e.g., **A**. The j^{th} column of **A** is denoted by a_j and element (i, j) by a_{ij}. Tensors are denoted by boldface Euler script letters, e.g., $\mathcal{T}$ Element (i, j, k) of a 3rd-order tensor $\mathcal{T}$ is denoted by t_{ijk} .

A vector is a one dimensional data array and a matrix is a two dimensional data array consisting of some arbitrary values for each row and column entries. These values in a matrix can be referenced by two digit index e.g. $A_{i,j}$, *i* for row and *j* for the column entry position of each element in **A**. Quite similarly a tensor is a multi-dimensional data array which has $1 \ldots n$ dimensions. The order of a tensor is the number of dimensions, also known as ways or modes. E.g. the tensor $\mathcal{T} \in \mathbb{R}^{M_1 \times M_2 \times \ldots M_n}$ has dimensions from $1..n$ Vectors and matrices can be thought of as tensors of order one and two respectively. All vectors are tensors, but not all tensors are vectors. Matricizing is an important operation of tensor flattening. The following figures 1, 2 further help in the visualization of how matricizing is done. Given a third order tensor $\mathcal{T} \in \mathbb{R}^{4 \times 4 \times 6}$ the matricizing can also be done based on grouping individual component matrices.



Flattening a 3rd Dimensional tensor-Example a 4x4x6 Tensor into 6 individual component matrices of dimension 4x4.

Figure 1. View of Tensor with various component matrices.

1	0	1	0	1	0	1	0	1	0	1	0
0	1	1	1	0	1	1	1	0	1	1	1
1	0	0	1	1	0	0	1	1	0	0	1
1	1	1	0	1	1	1	0	1	1	1	0
1	0	1	0	1	0	1	0	1	0	1	0
0	1	1	1	0	1	1	1	0	1	1	1
1	0	0	1	1	0	0	1	1	0	0	1
1	1	1	0	1	1	1	0	1	1	1	0

Figure 2. An Example of a Tensor Flattened as a single matrix.

The searches made by a user reflect his interest. A tensor model consisting of the searched components as searched by a user is built. The major objective behind constructing individual user models is finding most relevant features in each dimension as searched by a user. The three common steps undertaken for modelling each user's individual search behavior represented as a tensor are 1) Model Construction (Building various tensor models from processed data), 2) Model Decomposition (finding prominent features and finding latent relationships between different features). 3) Finding relevant features for each dimension and saving such features as top *n* items. Each of these three steps is discussed below in detail.

Step 1, Model Construction: Prior to creating the tensor model, the data is preprocessed. Pre processing includes removing unwanted attributes, removing missing records and identifying duplicate records or features from the datasets. Once, this is achieved, the searches made by each user are grouped as per sessions. For each user, his session search data is analyzed. All unique features appearing in sessions are extracted to represent as modes into the tensor model. A tensor is created with all such features. The overall size of each dimension is the number of distinct objects referenced in a dimension. As an example for a car website, if there are 80 different models in the database, then the model dimension has 80 possible values. For a user u_j, if the user has searched for 3 different makes of a car, 8 different models of a car, 2 different body types categories, and 2 categories of search type (new as well as used car) and 4 different price ranges then each such distinct value of searched dimension (make, model, body type, search type and cost type) and denoted as (3,8,2,2,4) are the mode values to be fed into the tensor model. For each search of a user the five dimensions (make, model, body type, search type and cost type) values are identified and such distinct values for each search are counted. The term frequency of each search is counted, where two different searches are considered same if all the five searched dimensions are same. Thus, term frequency value for all the searches of a user are populated in the tensor. As an example, the term frequency t_{ijklmn} is an entry value at the *i, j, k, l, m* and *n* modes, where *i* represents the *Make*, *j* the *Model*, *k* the *Bodytype*, *l* the search type, *m* the cost ranges. The structure of individual user tensor created, consisting of 5 dimensions is as follows:

$$\mathcal{T} \in \mathbb{R}^{Make \times Model \times Bodytype \times Search\ Type \times CostType} \quad (1)$$

Input: Processed Web log data of each user.
Output: Tensor $\mathcal{T} \in \mathbb{R}^{M_1 \times M_2 \times M_3 \times .. M_n}$
$s_c = (q_1, q_2, ..., q_n)$ //search query components.
Begin
1. $u_j = [(q_1, q_2, ..., q_n)_1, ..(q_1, q_2, ..., q_n)_m]$.//For a user read search query components individually for each interest vector.
2. $\forall iu_j^{\,k} .....iu_j^{\,l}, if\{(q_n)_k = (q_n)_l\}, or\{(q_1...q_n)_k = (q_1...q_n)_l\}$.
//Count frequency denoted as *f* of his interest vectors. Interest vectors iu^k and iu^l are considered as same interest only when all searched parameters are same.
3. Create an empty sparse tensor $\mathcal{T}$, and populate it with frequency *f* and mode values as.
$\{\mathcal{T}(q_1, ..., q_n) = f\}$.
End

Figure 3. Algorithm for constructing Individual Users TSM from Web log data.

Step 2, Decomposition: It is data reduction method where the most commonly found components are clearly distinguishable from the not so important ones. In multidimensional data modeling, the decomposition process enables to find the most prominent components (i.e. tensor entries and modes) as well as the hidden relationships that may exist between different components. The overall influence and correlations of factors in each dimension is then represented by a component matrix, whose columns are the factors determined by the model. The matrix constructed summarizes the structure in each dimension. The two most well-known and commonly used multi way models are Tucker [13] models and the PARAFAC [12] model, which is also called CANDECOMP (Canonical Decomposition). CANDECOMP was proposed independently but is considered equivalent to PARAFAC. A new tensor decomposition model based on matrix SVD is also proposed by [14]. It is called as HOSVD (Higher order singular value decomposition).We have used these three popular and widely used PARAFAC [12], Tucker [13] and HOSVD [14] tensor decomposition techniques to decompose the constructed individual user models. Each of these techniques has been discussed in detail by [4], [15].

However, just to refresh the memory how multi-dimensional decomposition is achieved, we discuss PARAFAC briefly. PARAFAC is a generalization of PCA (Principal Component Analysis) to higher order arrays. Given a tensor of rank 3, as $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, a R-component PARAFAC model can be represented as

$$x_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr} + E \qquad (2)$$

where a_i, b_i, c_i are the i^{th} column of component matrices $\mathbf{A} \in \mathbb{R}^{I \times R}, \mathbf{B} \in \mathbb{R}^{J \times R}$ and $\mathbf{C} \in \mathbb{R}^{K \times R}$ respectively and $E \in \mathbb{R}^{I \times J \times K}$ is the three way array containing residuals. x_{ijk} represents an entry of a three way array of $\mathcal{X}$ and in the i^{th} row, j^{th} column and k^{th} tube. Thus in our case when the user's tensor (equation 1) is decomposed using [16], the various matrices formed are as shown in the figure 4 below. In figure 4, $\mathbf{M_1, M_2 .. M_n}$ are the various component matrices formed after the decomposition of the tensor, and *R* is the desired best rank tensor approximation, which is set as 1, 2 and 3 in all our experiments.

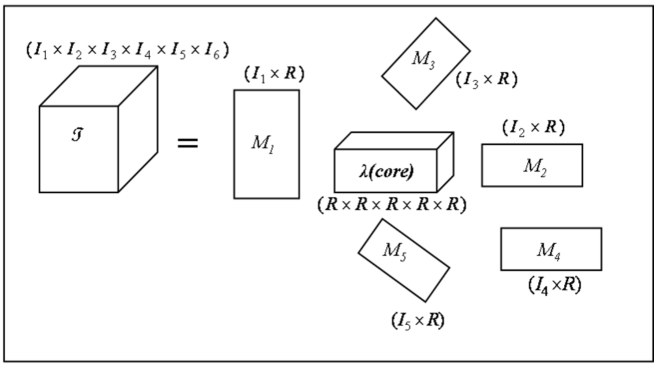


Figure 4. PARAFAC Decomposed tensor of users-searches, gives component matrices as shown

In case when the tensor rank decomposition (denoted as *R*), $R = 1$, such highest values can be found out easily, but in case when $R > 1$, then average of such row values is taken. For each dimension, only the top *n* aggregate values are saved in the user profile. Such top *n* dimension values are then used when making recommendations.

Step 3: To create individual user profiles based on tensors, we utilize the number of independent searches made by a user. Frequency of similar searches consisting of searched parameters (like in our case, the particular car make, model, bodytype, cost and search type of a car) are found out. Once the individual user model is created and decomposed (step 1 and 2), the top n values in each dimension are taken as the dimension values to be saved in the user profile. The complte algorithm for constructing user's tensor from his processed search data is explained in figure 3. Thus for each matrix from $\mathbf{M_1, M_2 .. M_n}$, (figure 4) we find top n values and save such values in the user profile table with complete details of objects as retrived from the database. As an example for the dimension car model, we have taken 3 highest decomposed values in each dimension for saving in the user profile (Table 1). We can say that highest PARAFAC decompostion values for car model and denoted as (Mode Value, Rank of Decomposition) are mode value with (5,1) =0.9806. Further, we can deduce that the specific user shows highest interest in a Mercedes-Benz car, as this car ranks highest in the specified dimension.

TABLE 1: PROMINENT DIMENSION (MODELS) VALUES IDENTIFIED FOR A USER.

Dimension=Car Models		
Highest PARAFAC Values	**Corresponding Values shown as (Id, Make,Model, Doors, Body type Year, Price)**	**Ranking**
(5,1) = 0.9806	4452638, Mercedes-Benz, 300, 4D, SEDAN, 1987, 8750.00	1
(8,1) = 0.1961	2851202, Alfa Romeo, 147, 5D, Hatchback, 2001, 13990.00	2
(286,1),= 0.0004	4398655, Toyota, Camry, 4D, Sedan,1988, 12999.00	3

Similarly, the prominent features in each component matrices for car make, body type, search type and cost type of a user are found out. These top n feature values with scores are stored in the user profile model. Once when a user makes new searches, this user profile information can be used to recommend him interesting items.

IV. EVALUATION METRICS

Dataset: Real car sales web log data from a popular car sales website [1] in Australia is taken for evaluation of experiments. A portion of the dataset consisting of 20 users, over a month's time was randomly selected, where one of users was a frequent visitor (user1, with 700 searches) and rest were users, where each one had made different number of searches. Each of these users had made at least 4 searches. The mean number of searches for these remaining users was 56, with minimum number of search being 4. The number of searches made by these users are shown in figure 5.

To evaluate the quality of top-n recommendations given by each method we used the following metrics. Let S_n be the actual searches made by a user U_n, which are taken after the user model is created (figure 6), and let R_n^m be the top-n recommendations given by various methods to U_n, where $n \geq 3$ and $n \leq 15, m \in \{\{3\},\{5\},\{10\},\{15\}\}$. We are considering top 3, 5, 10 and 15 recommendations. Precision (Pr_n) and recall (Re_n) for each user U_n are evaluated as

$$Pr_n = \frac{|R_n^m \cap S_n|}{|R_n^m \cap S_n| + |(R_n^m - S_n)|}, \quad Re_n = \frac{|R_n^m \cap S_n|}{|R_n^m \cap S_n| + |S_n - R_n^m|} \qquad (3)$$

The various methods used for evaluation are recommending highly searched items (Frequency based), associations (Finding associations of relevant make-model of a car for a user's searches), singular value decomposition (SVD), principal component analysis (PCA), non negative matrix factorization (NNMF) and various tensor decomposition techniques like PARAFAC, HOSVD and Tucker. We identified two popular dimensions like make and model of a car and built various matrices of each user. These two dimensions have been taken as cars belonging to any category can be clearly identified from these two features. Similarly, for tensors, top n values of make and model after decomposition are taken for evaluations.

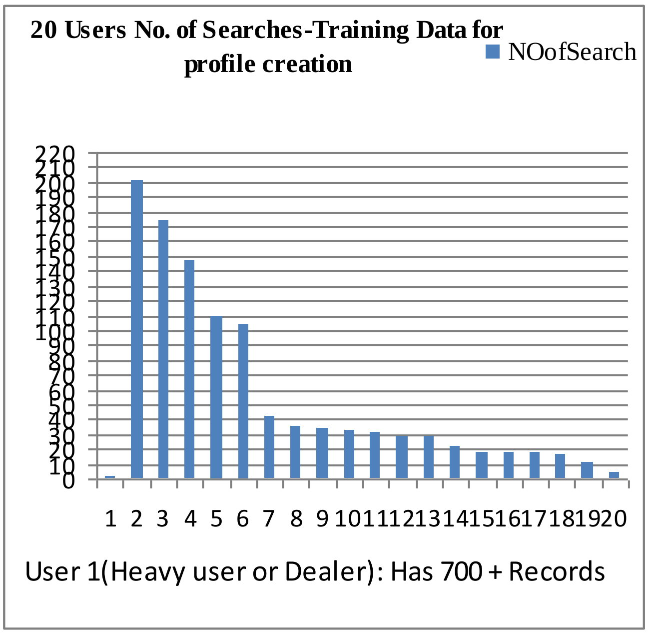


Figure 5. Number of User searches Used for identifying user's interests.

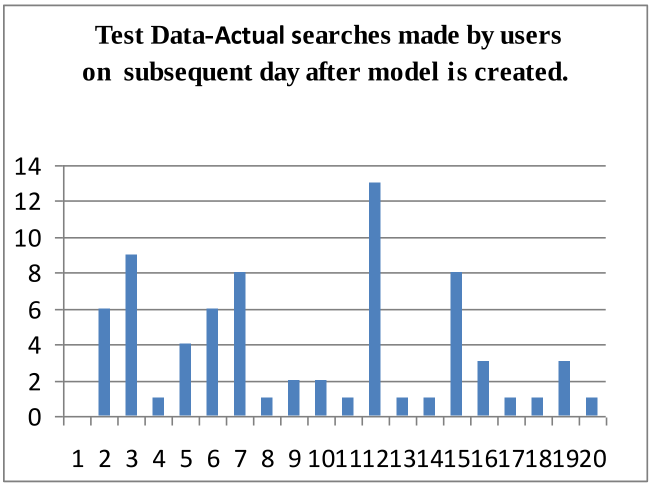


Figure 6. Users searches against which recommendations are measured.

V. RESULTS

The average precision, recall and F-Score for all methods are shown in table 2 and 3 respectively. The comparative result between NNMF and tensors is shown in table 4 and

[1] Due to privacy issues we are unable to specify details about the website.

summarized results of matrix methods and tensor methods are shown in table 5. In figure 7 we show comparative results of all methods with the tensor methods. The top 3, 5, 10 and 15 recommendations for each method and for each user is evaluated and then compared with the user's actual searches, where such actual number of searches are shown in figure 6. The numbers of searches made by User 1 are not shown in figure where, user 1 had made 48 searches.

In the tables 2, 3 and 4 NNMF-1, NNMF-2 and NNMF-3 refers to non negative matrix factorization of rank 1, 2 and 3 respectively. Similarly each Parafac 1, 2, 3, Tucker 1, 2, 3 and HOSVD 1, 2, 3 refer to PARAFAC, Tucker and HOSVD decomposition of rank 1, 2 and 3.

TABLE 2. AVERAGE PR FOR ALL METHODS

Method	Top3		Top5		Top10		Top15	
	Pr	Re	Pr	Re	Pr	Re	Pr	Re
Frequency	0.30	0.62	0.21	0.62	0.14	0.54	0.13	0.40
Association	0.37	0.63	0.24	0.61	0.16	0.59	0.10	0.36
SVD	0.35	0.54	0.18	0.55	0.13	0.59	0.09	0.54
PCA	0.27	0.36	0.12	0.29	0.07	0.31	0.06	0.32
NNMF-1	0.35	0.54	0.17	0.51	0.13	0.59	0.09	0.54
NNMF-2	0.38	0.63	0.19	0.58	0.14	0.57	0.11	0.37
NNMF-3	0.37	0.67	0.18	0.63	0.11	0.62	0.10	0.48
Parafac1	0.41	0.63	0.23	0.64	0.16	0.60	0.20	0.58
Parafac2	0.40	0.53	0.22	0.54	0.15	0.52	0.13	0.33
Parafac3	0.38	0.62	0.23	0.68	0.20	0.69	0.22	0.51
Tucker1	0.38	0.61	0.20	0.57	0.13	0.50	0.11	0.45
Tucker2	0.37	0.57	0.19	0.51	0.12	0.49	0.09	0.46
Tucker3	0.42	0.64	0.21	0.57	0.13	0.56	0.10	0.54
HOSVD1	0.38	0.61	0.20	0.55	0.13	0.54	0.11	0.54
HOSVD2	0.38	0.62	0.19	0.55	0.12	0.53	0.09	0.51
HOSVD3	0.39	0.63	0.21	0.57	0.15	0.60	0.12	0.62

TABLE 3. AVERAGE F-SCORE OF ALL USERS.

Method	Top3	Top5	Top10	Top15
Frequency	0.40	0.31	0.22	0.20
Association	0.47	0.34	0.25	0.16
SVD	0.42	0.27	0.21	0.15
PCA	0.31	0.17	0.11	0.10
NNMF-1	0.42	0.26	0.21	0.15
NNMF-2	0.47	0.29	0.22	0.17
NNMF-3	0.48	0.28	0.19	0.17
Parafac1	0.50	0.34	0.25	0.30
Parafac2	0.46	0.31	0.23	0.19
Parafac3	0.47	0.34	0.31	0.31
Tucker1	0.47	0.30	0.21	0.18
Tucker2	0.45	0.28	0.19	0.15
Tucker3	0.51	0.31	0.21	0.17
HOSVD1	0.47	0.29	0.21	0.18
HOSVD2	0.47	0.28	0.20	0.15
HOSVD3	0.48	0.31	0.24	0.20

TABLE 4. AVERAGE SUMMARY OF RESULTS OF TSM AND NNMF.

Methods	Top 3	Top 5	Top 10	Top 15
NNMF 1-3	0.46	0.28	0.21	0.16
TSM (Parafac 1-3, Tucker 1-3, Hosvd 1-3)	0.48	0.31	0.23	0.21
% Improvement	4.35 %	10.71 %	9.52 %	31.25%

TABLE 5: AVERAGE SUMMARY OF F-SCORE RESULTS OF MATRIX METHODS AND TSM BASED METHODS.

Methods	Top 3	Top 5	Top 10	Top 15
PCA,SVD, NNMF 1-3	0.40	0.24	0.18	0.14
TSM	0.48	0.31	0.23	0.21
% Improvement	20%	29.18%	27.78%	50%

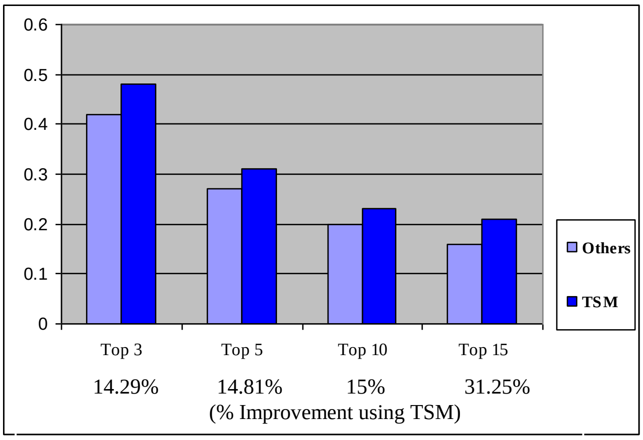


Figure 7. Average Summary of F-Score Results of All methods and TSM based methods.

VI. DISCUSSION

Clearly as can be seen from the results (Tables 2, 3, 4, 5 and figure 7) tensor based user modelling and subsequent recommendation out performs the recommendations given by two-dimensional vector and matrix based models. Two-dimensional models suffer from rotational freedom and thus are unable to find latent relationships between items. Tensors based methods give superior co-relations of items-items and thus are able to find highly relevant components. Since the data is sparse, and contains a lot of noise, NNMF performs quite well (Table 4), where two-dimensional methods are considered. Due to this ability of NNMF, it is often considered analogues to tensors, where two-dimensional data models are considered. Overall, when average F-Score of tensor and three matrix based methods (SVD, PCA and NNMF) are compared (Table 5), TSM results are far superior to the matrix methods. In case, when F-Scores of all methods like recommendation using association rule mining, frequency based recommendation (recommending highly searched items by a user), and all recommendations made by matrix and tensor methods are compared, it can be clearly seen as in figure 7, that TSM methods out performs all other methods, and the quality of recommendations is far superior than recommendations made by such methods. All results clearly outline the performance of TSM methods with their ability to identify top rated interests of a user from such complex multi dimensional Web log data.

VII. CONCLUSION

User behaviour modelling based on multiple searched attributes is a complex problem. Various methods from vectors to matrices are currently used to find prominent features as searched by a user. However due to the multi dimensionality of Web log data, such information is prone to loose latent relationships that exists between features, when such data is modelled as a two dimensional data. In order to map item-item relationships in a better way and to avoid loosing the latent relationships that exist between different searched components, there is a need to model such data using some high dimensional data analysis techniques like tensors. This research focuses on using tensors to mine knowledge from such data for effective user behaviour modelling. However, one major drawback of building individual tensor model for each user is the overhead in space and time. Time is not a big issue as such models can be build offline, but space and computational costs versus quality of recommendations is an important consideration, which has to be carefully analyzed when employing such methods for user behaviour modelling used for identifying interesting patterns from his Web log search data.

REFERENCES

[1] Mobasher, B., "Data Mining for Web Personalization," in The Adaptive Web. vol. 4321, A. K. P. Brusilovsky, and W. Nejdl (Eds.), Ed., 2007, pp. 90-135.

[2] Skillicorn, D., Understanding complex datasets: data mining with matrix decompositions: Chapman & Hall/CRC, 2007.

[3] Kleinberg, J., "Authoritative sources in a hyperlinked environment.," in Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, January 1998., pp. pages 668–677.

[4] Kolda, T. G. and Bader, B. W., "Tensor decompositions and applications," Technical Report SAND2007-6702, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, , November 2007.

[5] Sun, J. T., Zeng, H. J., Liu, H., Lu, Y., and Chen, Z., "CubeSVD: a novel approach to personalized Web search," in International World Wide Web Conference Committee(IW3C2), Chiba, Japan., May 10-14,2005, pp. 382-390.

[6] Evrim Acar, S. A. C. a., Mukkai S. Krishnamoorthy, and B ̈ulent Yener, "Modeling and Multiway Analysis of Chatroom Tensors," in Intelligence and Security Informatics. vol. 3495/2005: Springer Berlin / Heidelberg, 2005, pp. 256-268.

[7] Chu, W. and Ghahramani, Z., "Probabilistic models for incomplete multi-dimensional arrays," in Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS), Clearwater Beach, Florida, USA., 2009.

[8] Ruxanda, P. S. M. and Manolopoulos, A. N. Y., "Ternary Semantic Analysis of Social Tags for Personalized Mucis Recommendation," in Proceedings of the 9th International Conference on Music Information Retrieval, 2008, p. 219.

[9] Symeonidis, P., Nanopoulos, A., and Manolopoulos, Y., "Tag recommendations based on tensor dimensionality reduction," in RecSys '08', Lausanne, Switzerland, 2008, pp. 43-50.

[10] Rendle, S. and Marinho, B., "Learning optimal ranking with tensor factorization for tag recommendation," in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, Paris, France 2009, pp. 727-736.

[11] Mirzal, A., "Weblog Clustering in Multilinear Algebra Perspective," International Journal of Information Technology, vol. Volume 15, 2009.

[12] Harshman, R. A., "Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multi-modal factor analysis," UCLA working papers in phonetics, vol. 16, p. 1, 1970.

[13] Tucker, L. R., "Some mathematical notes on three-mode factor analysis," Psychometrika, vol. 31, pp. 279-311, 1966.

[14] De Lathauwer, L., De Moor, B., and Vandewalle, J., "A Multilinear Singular Value Decomposition," SIAM JOURNAL ON MATRIX ANALYSIS AND APPLICATIONS, vol. 21, pp. 1253-1278, 2000.

[15] Faloutsos, C., Kolda, T. G., and Sun, J., "Mining large graphs and streams using matrix and tensor tools," in Proceedings of the 2007 ACM SIGMOD international conference on Management of data, Beijing , China, 2007, pp. 1174-1174.

[16] Bader, B. W. and Kolda, T. G., "Efficient MATLAB computations with sparse and factored tensors," SIAM Journal on Scientific Computing, vol. 30, pp. 205-231, 2007.

ABOUT AUTHORS

Rakesh Rawat is presently pursing his PhD degree in computer science at the Queensland University of Technology, Brisbane Australia. Prior to his PhD degree he has worked in India (Uttranchal) as a Data Manager in the World Bank funded project IDSP (Integrated Disease Surveillance Project). His area of interests include data mining, pattern recognition, Web mining, Database management systems, Multi dimensional data modeling methods, e-health systems and Semantic Web technologies.

Dr. Richi Nayak is a senior lecturer in the faculty of science and technology at the Queensland University of Technology, Brisbane Australia. Her research interests include areas such as data mining, information retrieval and Web intelligence. Dr. Nayak has published over 87 refereed research articles in quality journals, conferences and books since 1999. She has been invited to serve as a program committee member in various conferences (PAKDD07, AusDM06, IAI07, KES06, PAKDD06, AMIT06, IRMA05, CITSA05 and UWSI06). She is editorial advisory reviewer board member of the International Journal of Knowledge-Based & Intelligent Engineering Systems (KES), International Journal of Cases on Electronic Commerce (IJCEC), and Information Resource Management Journal (IRMA). Dr. Nayak has developed various cutting edge techniques to mine XML and Web data, and also has successfully applied her data mining proficiency in a number of application domains such as Software Engineering, e-commerce, m-commerce, Web services, social networking and many other engineering fields.

Dr. Yuefeng Li is a professor in the faculty of science and technology at the Queensland University of Technology, Brisbane Australia. Dr Li has published more than 83 refereed papers and numerous journal articles and has co-authored a book. Dr. Li has been awarded an ARC discovery grant, an ATN grant and a QUT grant. He is a member of IEEE computer society and WIC (Web Intelligence Consortium). He served as the co-chair of various International conferences (WI/IAT04, AMT05 and AMT06). He is also an Associate Editor of International Journal of Pattern Recognition and Artificial Intelligence and IEEE Intelligent Informatics Bulletin. Dr. Yuefeng Li's expertise includes Web Intelligence and Data Mining. He has contributed an innovative and promising technique called as ontology-based Web mining aimed at solving Web-based problems.