Recommender Systems Using Harmonic Analysis

Gilbert Badaro
American University of Beirut
Beirut, Lebanon
ggb05@aub.edu.lb

Hazem Hajj
American University of Beirut
Beirut, Lebanon
hh63@aub.edu.lb

Ali Haddad
Yale University
Connecticut, USA
ali.haddad@yale.edu

Wassim El-Hajj
American University of Beirut
Beirut, Lebanon
we07@aub.edu.lb

Khaled Bashir Shaban
Qatar University
Doha, Qatar
khaled.shaban@qu.edu.qa

***Abstract*— Recommender systems provide recommendations on variety of personal activities or relevant items of interest. They can play a significant role for E-commerce and in daily personal decisions. However, existing recommender systems still face challenges in dealing with sparse data and still achieving high accuracy and reasonable performance. The issue with missing rating leads to inaccuracies when trying to match items or users for rating prediction. In this paper, we propose to address these challenges with the use of Harmonic Analysis. The paper extends on our previous work, and provides a comprehensive coverage of the method with additional experiments. The method provides a novel multiresolution approach to the user-item matrix and extracts the interplay between users and items at multiple resolution levels. New affinity matrices are defined to measure similarities among users, among items, and across items and users. Furthermore, the similarities are assessed at multiple levels of granularity allowing individual and group level similarities. These affinity matrices thus produce multiresolution groupings of items and users, and in turn lead to higher accuracy in matching similar context for ratings, and more accurate prediction of new ratings. The evaluation of the system shows superiority of the solution compared to state of the art solutions for user-based collaborative filtering and item-based collaborative filtering.**

Keywords— Recommender Systme; Sparse Matrix; Partition Tree; Multiresolution Analysis; Haar Basis; Coupled Geometry

I. INTRODUCTION

Recommender systems continue to capture attention due to their potentials in helping people with their daily-life decisions such as: what book to read, what movie to watch, what music to listen to, where to eat, which links to visit on the web, etc. [1]. Moreover, recommender systems are becoming more of a necessity since the amount of information available to the user is overwhelming and thus, an automated system can assist the users by providing only what is relevant to their needs [2]. It is also a helpful tool for businessmen in the marketing field.

Several approaches have been proposed in the literature for recommender systems. Traditional methods rely on the ratings provided by the users on previously rated items. These methods can be classified into: collaborative techniques, content-based techniques, hybrid models, and preference-based methods [3]. However, existing recommender systems are still facing challenges when dealing with sparse data, and in achieving high accuracy.

In this paper, these challenges are addressed by using a new approach based on harmonic analysis [4]. This paper extends our previous work on recommender systems [5], and provides a comprehensive coverage of the approach with additional experiments showing consistently the superiority of the method.

The rest of the paper is organized as follows: Section II provides a literature review on conventional and state-of-the-art techniques for recommender systems. Section III presents the proposed approach which extends harmonic analysis for user-item matrix. Section IV reports evaluation and comparison results of the proposed approach with conventional and state of the art methods. Section V concludes the paper by summarizing findings and outlining future work.

II. LITERATURE REVIEW

Recommender systems can be classified into four types of approaches: collaborative filtering techniques [6-11], content-based systems [3], hybrid recommender systems [12-15], and preference-based systems [16-19].

Collaborative filtering techniques can be divided into user-based and item-based collaborative filtering. In the user-based case, similarity between two users is computed while in the item-based case, similarity between two items is computed. Collaborative filtering can be memory based or model-based. Content-based techniques look at the content of the items and try to retrieve features specific for a certain type of items. Textual features are usually used as features for content-based systems. Hybrid models were developed to overcome the limitations of collaborative filtering and content-based techniques. Briefly, these models combine the outputs of collaborative filtering and content-based methods using for example linear combinations of predicted ratings or voting schemes. Preference based methods are newly introduced approaches for recommender systems that identify abstract features and relations based on the user profile which could include user's age, gender and location.

Recommender systems are being utilized in different scenarios such as recommending social events based on the user geographical information [20], helping users in selecting their travel packages [21], recommending web pages [22], solving patent maintenance related problems [23] and last but not least for recommending collaborators in scientific research across different domains [24].

Privacy issues [25], evaluation metrics [26] and scalability of recommender systems are still big challenges for recommender systems. Along those lines, existing recommender

978-1-4799-4274-9/14 $31.00 © 2014 IEEE
DOI 10.1109/ICDMW.2014.28

IEEE computer society

systems still face challenges in dealing with sparse data and achieving high accuracy. The issue with missing rating leads to inaccuracies when trying to match items or users for rating prediction. In this paper, we propose to address this issue of sparse user-item matrices, and to achieve higher accuracies than existing systems.

III. Proposed Multiresolution Aproach for Rating Prediction

In this section, we provide the details for the proposed approach to address the sparse matrix problem in a user-item matrix and increase the accuracy of rating prediction. The proposed method is based on a new formulation for harmonic analysis with application to user-item matrix. The idea is to derive a multiresolution transformation of the user-item matrix similar to a wavelet transform. This transformation captures inherently the interplay between users and items at multiple levels of granularity, which enables similarity evaluation at both individual and group levels. The intuition is that similar users are likely to have similar items and similar items are likely to have similar users. While previously published hybrid recommender solutions covered aspects of the interplay between users and items, none provided the simultaneous evaluation of interplay measures at multiple resolutions, which should render rating estimation more accurate. Following the transformation, the most dominant coefficients in the new transformed space are used to reconstruct a matrix similar to the original user-item matrix, but with estimations of previously missing ratings. Hence, the issue of sparse user-item matrix is solved.

A. Overview of Approach

The overall approach can be represented by 8 steps as shown in Fig. 1.

Step 1: In this step, the similarities among users and items are computed separately using correlation measures. The similarity measures are represented by so called affinity matrices. For user affinity, similarity is measured between rows of the user-item matrix. For item affinity, similarity is measured between columns. The computation of each affinity matrix is further detailed in section III.B.

Steps 2-4: These steps constitute an iterative process that converges to two multiresolution representation of the user-item matrix. The process initially starts by deriving a multiresolution partition tree for items by clustering similar groups of items (columns) together at different granularity levels. The details for constructing the partition tree are provided in section III.C. In step 3, the interplay and similarity between users (rows) and items (columns) are measured by computing similarity between users (rows) based on the similarities across rows of the items' partition tree, i.e. rows across the multiresolution levels of the items' partition tree. This measure of interplay is called dual affinity. Similarly, the interplay and similarity between items (columns) and users (rows) are measured by computing similarity between items (columns) based on the similarities across columns of the users' partition tree derived in step 4, i.e. columns across the multiresolution levels of the users' partition tree. This iterative approach in steps 2-4 is repeated until the partition trees converge with very little change from one iteration to the next. The results of this convergence are two multiresolution partition trees capturing the interplay: one for the users and one for the items. These steps are further detailed in section III.D.

Steps 5-6: Through these two steps, the user-item matrix is transformed to the new multiresolution space with the use of the orthonormal Haar-like bases constructed from the partition trees. The product space spanned by the tensor product of the Haar-like bases can then be used to represent the original rating matrix in the new space. The orthonormal representation is constructed to represent the original user-item matrix in the new space of the multiresolution partition trees. Given a partition tree, T_Y, created from the rows of a user-item matrix M, we compute an orthonormal basis that spans the set of step functions that are constant on the nodes of T_X at a given level of the tree. As an example, columns of M are interpreted as step functions that are constant on the nodes at the finest level. Following the same process, a Haar-like basis can be constructed for the partition tree T_Y in relation to the columns Y. This part of the algorithm is detailed in section III.E.

Step 7: Similar to a Wavelet transform, this new transform can be used to efficiently compress and denoise the user-item matrix. As a result, in step 7 of the approach, the dominating coefficients in the transformed space are selected to provide the efficient representation. This step is included in section III.F.

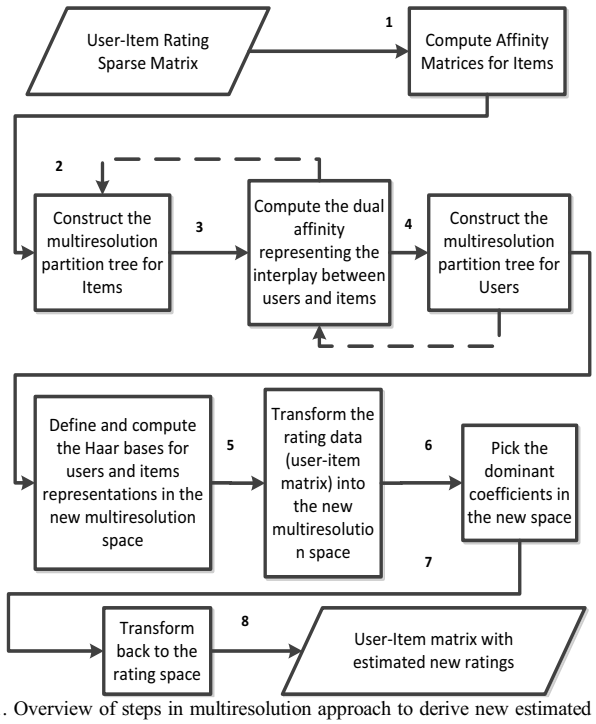

Fig. 1. Overview of steps in multiresolution approach to derive new estimated ratings.

Step 8: This step involves a step similar to an inverse wavelet transform. The dominant coefficients selected from the previous step are transformed back to the original user-item space. Typically, a small percentage of the coefficients are used to reconstruct the new estimated user-item matrix with the desired capture for previously missing ratings. This step is further described in section III.F.

Since the user-item matrix is not smooth, in other words, the rating values are typically discrete, the described process (steps 1-8) of transformation and reconstruction can be iterated several times to get smoother and more accurate results. This is called a spin cycle procedure. For p iterations, the final estimated user-item matrix is calculated by averaging over the individual user-item matrices outcome from each iteration. The number of iterations can be chosen as a tradeoff between accuracy of convergence and computational complexity. For the experimentation in this paper, it was set to 5.

B. User-user and Item-item Affinity Matrices

Given a user-item rating matrix M of size $m*n$, m represents the number of users, and n represents the number of items. Let $X = M$, and $Y = M^T$. These two matrices X and Y are used in formulating item related affinity matrix W_X and user related affinity matrix W_Y. We define the item affinity matrix W_X as follows using similarity measures for all items i and j.

$$W_X(i,j) = \frac{\sum_{u \in X_{ij}} (r_{u,i} - \bar{r}_i) * (r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in X_{ij}} (r_{u,i} - \bar{r}_i)^2} * \sqrt{\sum_{u \in X_{ij}} (r_{u,j} - \bar{r}_j)^2}} \quad (1)$$

where X_{ij} represents all users who rated both items i and j, where $i \neq j$, *and* $1 < i, j < n$. n is the total number of items. $r_{u,i}$ and $r_{u,j}$ are the ratings provided by user u to items i and j respectively. $\bar{r}_i$ and $\bar{r}_j$ are the averages of all the ratings provided for items i and j respectively. Similarly one can compute the affinity matrix W_Y among users using (2).

$$W_Y(u,v) \frac{\sum_{i \in Y_{uv}} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in Y_{uv}} (r_{u,i} - \bar{r}_u)^2 \sum_{i \in Y_{uv}} (r_{v,i} - \bar{r}_v)^2}} \quad (2)$$

where Y_{uv} is the set of items that were simultaneously rated by users u and v. $r_{u,i}$ and $r_{v,i}$ are the ratings provided for item i by users u and v respectively, and where $u \neq v$, $1 < u, v < m$. m is the total number of users. $\bar{r}_u$ and $\bar{r}_v$ are the averages of all the ratings provided by users u and v respectively.

The computations of affinity can also be performed using other statistical measures indicating distance or similarity such as cosine measures or Pearson correlation. We use correlation for ease of illustration.

C. Multiresolution Partition Trees

Given a user-item matrix M, one can organize X, the set of items' ratings (columns), and Y, the set of users' ratings (rows) into two partition trees using the affinity matrices W_X and W_Y evaluated for the items and users separately. These partition trees, T_X and T_Y, give a hierarchical multiresolution grouping, called folders, of similar users and items respectively.

To illustrate the approach for partition trees of X and Y, consider the partition tree T_X for items X. Every node in the tree is a folder, and represents a cluster of items at the level of the node in the tree. A sample partition tree of three levels ($L = 3$) is shown in Fig. 2. The finest level corresponds to the leafs of the partition tree. At this finest level, the partition is composed of folders where each folder represents one column from the user-item matrix, i.e. a set of users' ratings for an item. At higher levels of the tree, starting from the finest level (leafs), the folders from the previous level are grouped to form a coarser partition at the next level up the tree. For a tree of depth L, X_l represents the set of nodes at any level, where $1 < l < L$. X_1 ($l=1$) represents the root of the tree, and X_L represents the set of nodes at the finest level. The full partition tree T_X is represented by $\{X_1 \cdots, X_L\}$. For each level l, where $1 \leqslant l \leqslant L$, the partition is composed of $n(l)$ mutually disjoint folders X_i^l where $1 \leqslant i \leqslant n(l)$.

The algorithm for the construction of the partition tree is shown in Fig. 3. At the finest level, each node contains the set of elements in a column from the item matrix X, and the set of nodes is equal to the number of columns. To get coarser representations in the tree, the clustering is done bottom-up. Starting from the finest level, nodes at the finer level are clustered to form the next level up in the tree. A notion of distance between nodes needs to be defined in order to group nodes. We choose the diffusion distance as defined in [26] since it was shown to help capture the overall geometry of the data for multi-scale analysis. The diffusion distance provides means of computing distances between data in a transformed space, similar to other transforms such as Principal Component Analysis (PCA). As a result, the diffusion distances can be computed by finding the distances between vectors in a new space defined by the eigenvectors of the affinity matrix W_X.

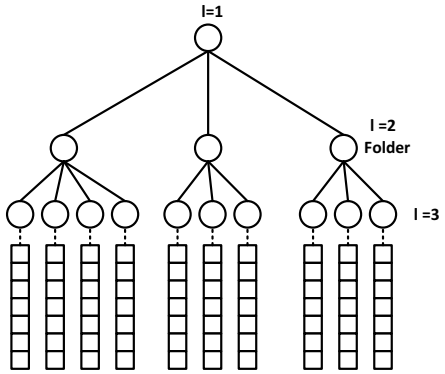

Fig. 2. Sample partition tree with three levels (L = 3) for items showing the groups of items' folders as nodes at multiple resolutions. At the finest level (l=3), each node corresponds to a column in the user-item matrix.

Specifically, a diffusion distance can be computed as follows: first, the affinity matrix W_X is normalized such that the sum of the elements in each row of W_X is equal to 1 as shown in (3) and (4).

$$P_X = O^{-1} * W_X \quad (3)$$

and O is a diagonal matrix such that

$$O_{i,i} = \sum_j W_X(i,j) \quad (4)$$

Then, we compute the eigenvectors and eigenvalues of P_X^T. The diffusion distance between two items is computed by finding the L_2 norm between the corresponding eigenvectors to these items (5).

$$D_X(i,j) = \|A_X(i) - A_X(j)\| \quad (5)$$

where $A_X(i)$ represents the eigenvectors of item i for the normalized affinity matrix P_X^T, and similarly for item j.

A more detailed explanation of the diffusion map theory can be found in [26].

Clustering starts at the finest level L by computing a base radius ρ>0, by averaging the diffusion distance between an item and its *KNN* neighbors. The choice of number of nearest neighbors, k_n, in the *KNN* method is manually set at the first iteration and is then decreased by half when moving to the next level. The choice can be experimentally chosen to achieve highest accuracy from a given set of training data. Once the radius ρ is computed, the items' data are clustered into disjoint groups of radius ρ, called balls. The number of resulting balls is denoted by k, and the centers of the balls, or centroids, are used to represent the disjoint folders at the next level of the tree. At the finest level, a node i, which corresponds to an item data, is merged with a group of centroid z_i and belongs to the corresponding folder X_i^{L-1} as per (6):

$$X_i^{L-1} = \{i \in X | D_X(i, z_i) < \rho\} \quad (6)$$

for $i = 1 \ldots n(L-1)$, where $n(L-1) = \#\{z_{L-1}\}$ is the number of centroids at level L-1. In order to move to the next level $L-2$ of the partition tree, we define an affinity between folders at nodes i and j, as:

$$\widehat{W_X}(i,j) = \left\langle W_{X_i^{L-1}}, W_{X_j^{L-1}} \right\rangle = \sum_{x \in X_i^{L-1}} \sum_{y \in X_j^{L-1}} \sum_r W_X(x,r) W_X(r,y) \quad (7)$$

$\sum_r W_X(x,r)W_X(r,y)$ is an affinity measure for reaching node x to node y in two steps. This hierarchical grouping process is repeated until the coarsest level ($l = 1$) is reached, with a single folder X^1.

D. Dual Affinity between Users and Items

The partition tree of the items (columns), T_X, is then used to define the interplay between items and users through a dual affinity matrix. This dual affinity is also called a dual geometry on the users (rows) of M. The dual affinity on users can be computed by taking the similarity between users at every level of the item partition tree. A similar approach can be done for dual affinity on items. To represent the average of similarities across different levels of the partition tree, consider the case of dual affinity on users. The idea is to compute similarity in rows of the original user-item matrix M, by computing the similarity of the corresponding rows across the multiresolution tree achieved on items. The novelty in this approach is that it captures an overall geometry of the users that depends on the geometry of the items and vice versa. Because of these dependencies, we refer to these geometries as a coupled geometry. Interpreting users' ratings as functions or mappings on T_X is similar to many wavelet-based norms (e.g. Besov norms).

First, we define a mapping from a row in the original matrix to a row at any level of the partition tree. At the finest level, the elements of a row y from the original matrix M correspond to the cells $M(y,i)$ corresponding to item i. This mapping process can be generalized, as shown in (8), to identify cells at any levels of the partition tree.

$$For\ each\ y \in M: y \to M_l(y,i) \quad (8)$$

where M_l represents the user-item matrix at level l, and $1 \leq i \leq n(l)$. The set of values for row y, can be represented by $M_l(y)$.

With this mapping in place, we can define the dual affinity between rows u and v based on the partition T_X, as the average of similarities between the rows at every level as shown in equation (9) and illustrated in Fig. 4.

$$W_{T_X}(u,v) = \frac{1}{L}\sum_{l=1}^{L} W\left(M_l(u), M_l(v)\right) \quad (9)$$

The example in Fig. 4 shows that there are three available columns at level 2 of the tree. The dual affinity is the average of the three affinities computed at each level respectively as shown in Fig. 4. Similarly, the dual affinity on items based on partition T_Y is defined as:

$$W_{T_Y}(i,j) = \frac{1}{L}\sum_{l=1}^{L} W\left(M_l^T(i), M_l^T(j)\right) \quad (10)$$

where T_Y is the partition tree for users constructed based on the user affinity matrix W_Y, M_l^T represents the transpose of the original user item matrix at level l of the partition tree T_Y. We iterate the computations of the dual affinities on X & Y until (11) is satisfied, where W_X^k represents the updated affinity matrix at iteration k.

```
Input:
    - X: Set of item columns from the user-item matrix
    - Wx: Affinity matrix corresponding to items.
    - kn /*number of nearest neighbors used to compute
      average diffusion distance*/
Output:
    - Tx: Partition tree for items
Algorithm:
Initial Finest level;
Dx = computeDiffusionDistance(Wx);
Rho = computeBaseRadius(X, Dx, kn);
InitialSetofCentroids = findCentroids(Dx, X, Rho);
count=1;
for each centroid in IntialSetofCentroids
{
 Folder=groupItems(X, Dx, Rho, centroid);
 /*A folder is represented by its centroid*/
 Folders[level+1, count]= Folder;
 count ++;
}
/* Moving from a finer level to coarser levels in the
partition tree:*/
number_of_Folders_in_Level = count;
while (number_of_Folders_in_Level != 1)
{
 level ++;
 kn = ceil(kn/2);
 Rho = computeBaseRadius(X, Dx, kn);
 WTemp = computeAffinityBetweenFolders(Folders[level,:]);
 DTemp = computeDiffusionDistance(WTemp);
 TempCentroids =findCentroids(DTemp,Folders[level,:],Rho);
 count =1;
 for (centroid in TempCentroids)
 {

Folder=groupItems(Folders[level,:],DTemp,Rho,centroid);
Folders[level+1, count]=Folder;
count ++;
}
  number_of_Folders_in_Level = count;
}
Tx = Folders;
```

Fig. 3. Algorithm for creating partition trees on items.

Ideally, the difference between iterations should converge to 0. However, the ideal equality situation may not exist, or at least may take a long time to achieve. As a result, the choice of ϵ provides a tradeoff between best possible accuracy and computation cost. The closer the value of ϵ is to 0, the more the

time the algorithm takes for convergence. The results of these iterations give us the desired two multiresolution partition trees, one for the items and one for the users.

E. User-user and Item-item Affinity Matrices

Once the partition trees are built, we construct an overall representation of the whole tree, which would allow us to project the original matrix into a multiresolution space represented by the two trees of users and items. This is equivalent to a scale-space transformation in two dimensions X and Y. The items' tree provides the equivalent of the X-transformation, and the users' tree provides the equivalent of the Y-transformation. This can be represented mathematically by the X and Y transformations provided by two matrices H_X and H_Y as shown in equation (12):

$$M_{Coeff} = H_Y * M * H_X^T \qquad (12)$$

where M_{Coeff} represents the coefficients of the user-item matrix in the multiresolution space. Specifically, let the basis derived from the partition tree T_X be represented by an $n*n$ matrix H_X, where the columns are the basis vectors. The rows of M are first projected to have coordinates in H_X. The columns of M are then projected to the Haar basis matrix H_Y derived from the partition tree T_Y. As a result, M can be represented using the tensor product of the Haar bases H_X and H_Y, and one can get the coefficients as in (13).

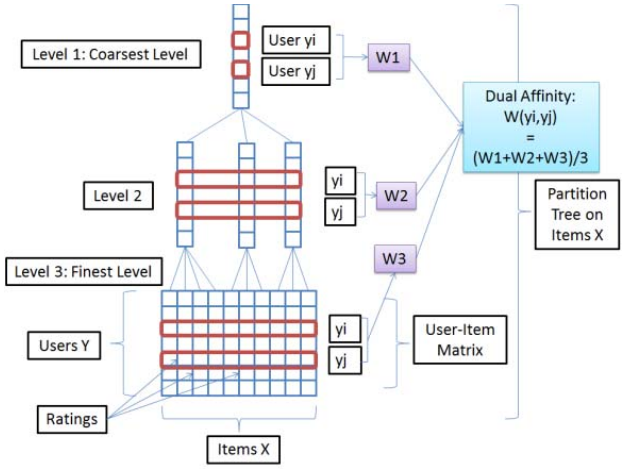

Fig. 4. Example of using dual affinity from items' partition tree to compute similarity for users.

The two matrices H_X and H_Y can be derived by a set of Haar bases for the users' tree and items' tree respectively. The construction is performed by adding the orthonormal bases constructed at each level of the partition tree using a Gram-Schmidt process, starting with the coarsest level (level 1) and iterating to finer levels until level L. Intuitively, the construction is equivalent to having low-pass filters to provide the projection to coarsest level of a tree, and high-pass filters for the details in the finer levels. In fact, the basis is derived by first projecting into the coarsest level. Subsequently, the projections at a finer level of the tree are subtracted from projections at the previous coarser level, to provide the components of the bases for the finer level.

To illustrate of the construction of the bases, let's consider the example shown in Fig, 4. At the coarsest level, the folder was generated by consecutive clustering as was explained in section III.C. As a result, the folder at the coarsest level corresponds to 10 columns (one for each item). Since there is only one folder at this coarsest level, the projection is simply a scalar. Since an orthonormal basis is needed, the value of the Haar basis is of dimension 1 and its value is $\frac{1}{\sqrt{10}}$.

For level 2, there are 3 folders. Thus a projection into these three vectors would require a basis matrix of size 3*3, call it B as in equation (13):

$$B = [e_1 \quad e_2 \quad e_3] \qquad (13)$$

e_1, e_2, and e_3 represent the orthonormal basis vectors for level 2. One of the vectors in this new basis B, e_1 can be extended from the basis of the finer level. Thus, the three cells of the first column vector e_1 of B have the value of $\frac{1}{\sqrt{10}}$.

Let f_1, f_2 and f_3 denote the folders from left to right. Let s_1, s_2 and s_3 denote the folder size of f_1, f_2 and f_3 respectively. In this case, s_1= 4 and s_2=s_3=3. To find the remaining two column vectors of B a Gram-Schmidt process is adopted. The steps are as follows: first, create a diagonal matrix *Diag* of size 3*3 whose diagonal entries are n_1, n_2 and n_3 , and which correspond to the sizes of the three folders at level l = 2 respectively. Then, define a matrix A of size *3*2* as shown in (14). Let $v_2$and v_3 denote the first and second column of A. Subsequently, the Gram-Schmidt process is applied on $v_2$and v_3 to derive the orthonormal basis as shown in (15):

$$A = \begin{bmatrix} 1/n_1 & 0 \\ -1/n_2 & 1/n_2 \\ 0 & -1/n_3 \end{bmatrix} \qquad (14)$$

$$e_2 = \frac{v_2}{\sqrt{v_2^T * Diag * v_2}}$$

$$w_3 = v_3 - \frac{v_3^T * Diag * e_2}{e_2^T * Diag * e_2} e_2 \qquad (15)$$

$$e_3 = \frac{w_3}{\sqrt{w_3^T * Diag * w_3}}$$

w_3 is residual in v_3 after subtracting the projection on e_2. This leads to three orthogonal basis v_1, v_2, and w_3. v_2 and w_3 are further normalized to provide the three orhornormal bases e_1, e_2, and e_3.

At level l =3, this process is repeated to generate a 10*10 Haar basis matrix. Since this level is the finest level, the resulting matrix is the desired H_X Haar basis matrix.

F. User-user and Item-item Affinity Matrices

After computing the coefficients as in (12), we select the dominant coefficients by keeping the ones that are greater than a pre-defined threshold R and eliminating other values as per (16). The size of the threshold is directly linked to the size of the support of the tensor product of the two Haar-like basis functions.

$$M_{Coeff} = M_{Coeff} .* boolean(M_{Coeff} > R) \qquad (16)$$

where ".*" is the pointwise multiplication.

Extending on the theory of Coifman et al. [4], one only needs to consider the coefficients that are large enough in order to reconstruct an updated filled user-item matrix with estimation of missed ratings. By this method, we would only keep the relevant coefficients that will be involved in approximating and filling the original user-item matrix. The reconstruction is performed by following (17) where $\widehat{M}$ is the final enhanced user-item matrix with the desired estimated ratings.

$$\widehat{M} = H_Y^T * M_{Coeff} * H_X \qquad (17)$$

IV. Evaluation

Several experiments are conducted to evaluate the effectiveness of the proposed method. In IV.A, we compare the accuracy of our approach against conventional user-based and item-based collaborative filtering techniques. Then in IV.B, we compare the accuracy of our approach against state of the art methods using larger data. In IV.C, we study the time performance of the multiresolution approach and compare it to recent and state of the art methods. In IV.D, we run the proposed approach with Netflix data set.

A. Comparison with Conventional Methods

The experiments described in this section are conducted on data selected from MovieLens [28], a web-based movie recommendation system that debuted in 1997. The selection of this dataset specifically is made in order to compare our results to user-based collaborative filtering and item-based collaborative filtering performed on the same dataset as described in [6]. The data was collected from hundreds of users who had visited MovieLens to rate and receive recommendations for movies. Several data sets exist on the site [28]. For the chosen data set, 100,000 ratings provided by 943 users to 1682 items were used for the evaluation. The data has a sparsity level of 0.94, which indicates that the matrix has 94% of its entries equal to zero and a high degree of sparsity. The sparsity level is computed as in (18).

$$Sparsity\ Level = 1 - \left(\frac{nonzero\ entries}{Total\ entries}\right) \qquad (18)$$

The data was divided into 5 training sets and 5 corresponding testing sets and thus, a 5-fold cross validation approach was applied (i.e. 80% training data and 20% test data). The accuracy of the proposed technique was compared to user-based collaborative filtering as stand-alone and item-based collaborative filtering as stand-alone. For fair comparisons, the algorithm was repeated 5 times, and the results were averaged consistent with the spin cycle procedure.

While several measures are possible for assessing the accuracy of the system, we used mean absolute error (MAE) to measure the deviation of recommendations from their true user-specified values. For each rating-prediction pair $<p_i, q_i>$, p_i being the predicted value and q_i the correct value available in the testing data, the absolute error is computed as $|p_i-q_i|$. The MAE is then evaluated by examining all N ratings-prediction pairs, and then computing the average error as shown in (19) below. The lower the MAE, the better the estimation is of missing ratings.

$$MAE = \frac{\sum_{i=1}^{N}|p_i - q_i|}{N} \qquad (19)$$

Fig. 5 shows the results achieved in terms of MAE compared to user-based collaborative filtering and item-based one for each of the five runs. As can be seen in Fig. 5, the proposed Harmonic Analysis approach achieved an improvement of 40% compared to user-based collaborative filtering and item-based collaborative filtering. The harmonic analysis approach has the lowest MAE compared to the two other methods.

B. Comparison with State of the Art Methods

This set of experiments are targeted for comparing proposed approach with some more recent and state of the art work in the field [29], [30] and [31]. The authors in [29] present a mixed matrix factorization approach that relies on exploiting latent factors and extracting the context of the user to predict item ratings. Treerattanapittak et al. propose in [30] an approach for improving collaborative filtering using a fuzzy clustering algorithm. For the experiments, we choose two different large datasets: the MovieLens 1M ratings which includes 6040 users and 3952 movies, and the MoviLens 10M ratings which has 10681 movies and 71567 users. A portion of the user partition tree the 1M dataset is illustrated with 58 levels in Fig. 6.

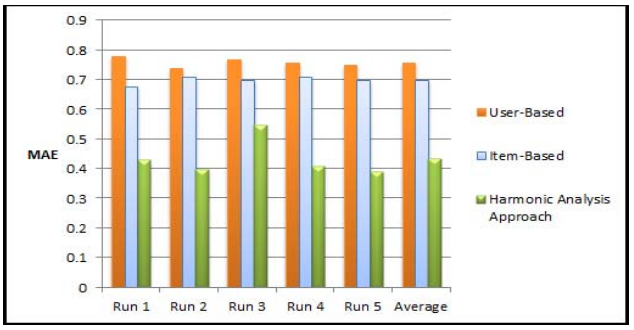

Fig. 5. Mean Absolute Error for user-based and item-based collaborative filtering and for Harmonic Analysis approach.

Following the same testing process described in the previous section, the results are reported in Table I. The new method showed, on average, an accuracy improvement of around 25%, 13% and 14% compared to [29], [30] and [31] respectively.

It is important to note that our approach as well as the one proposed in [30] are based on the provided ratings only whereas the one in [29] investigates latent factors, context and other inputs that are required to be provided by the user. This could be a disadvantage since many users prefer to skip the process of providing additional information.

C. Analysis of Time Performance

To evaluate performance of the algorithm, we evaluated the tradeoff between system accuracy and time of running the algorithm. The experiments were implemented on MATLAB installed on an Intel core I7 device with 6GB DDR3 RAM running Windows 7 64-bit. Time measurements were collect to reflect the duration needed to run one full iteration of the algorithm and reconstruct the user-item matrix. 5 iterations were performed and ϵ was set to 10^{-4}. As shown in Fig. 7, the experiments indicated that the total algorithm time decreases with the number of nearest neighbors k_n used to construct the partition trees. When k_n decreases to less than 5, the algorithm time increases exponentially.

In Fig. 8, the MAE and time measurements are plotted based on varying the k_n parameter for the 10M dataset. It can be seen from this graph that the choice of k_n gives a tradeoff between accuracy and time. As can be seen in the graph, MAE increases when k_n increases while the time required by the algorithm decreases when k_n increases.

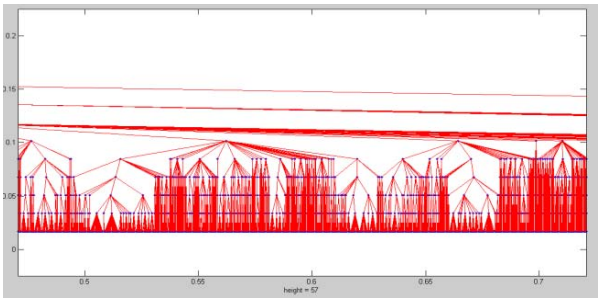

Fig. 6. Portion of the partition tree on users of the 1M dataset.

TABLE I. MAE for 1M and 10M MovieLens datasets.

Method	1M ratings	10M ratings
Multiresolution approach	**0.65**	**0.61**
Mixed Matrix Factorization [29]	0.86	0.84
Fuzzy Clustering [30]	0.72	0.71
SVD [31]	0.725	0.715

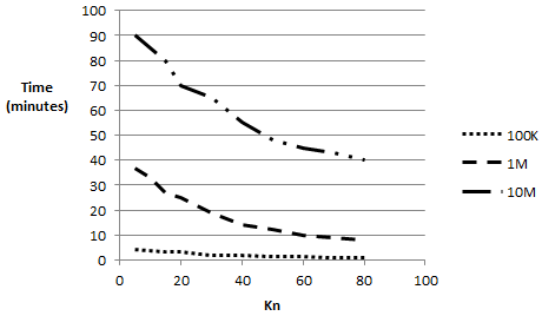

Fig. 7. System Performance (in minutes) for running the proposed approach in terms of kn for the three cases of Movielens dataset: 100 K, 1M, and 10M.

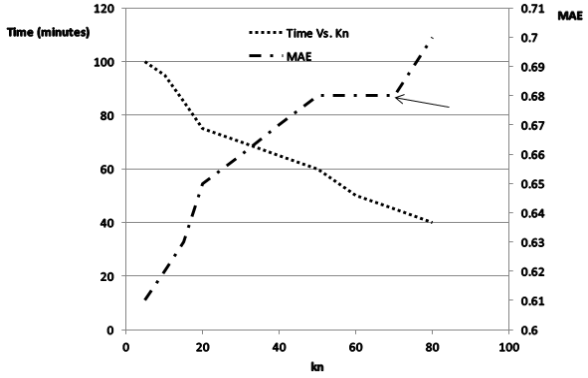

Fig. 8. Performance (in minutes) and MAE versus choice of nearest neighbors (k_n) parameter for the 10M MovieLens dataset.

By checking the corresponding accuracies, the k_n values were chosen to be 15, 40 and 70 for the datasets 100k, 1M and 10M ratings respectively. These choices of k_n were based on the variation of time compared to the variation of MAE for each case of k_n. We chose a point where the MAE was given a higher tradeoff of accuracy versus computation time. As an example for the 10M dataset, the choice is pointed out by the arrow.

For comparison of time performance, we provide time measurements per iteration for the 1M dataset in comparison with state of the art approaches as shown in Table II.

As seen in Table II, the proposed approach provides better scalability performance. It is worth noting that the time comparison per iteration was chosen for consistency and fairness in comparison. Since the other methods are also iterative and the accuracy of their system also improves with the increase in the number of iterations, we provided time comparison per iteration. This reporting is also consistent with the way previous results were reported.

TABLE II. Comparison of time performance per iteration for the 1M MovieLens Dataset.

Method	Time in minutes / iteration
Our approach	**16**
Mixed Matrix Factorization [29]	19
Fuzzy Clustering [30]	21
SVD [31]	18

D. *Netflix Dataset*

In this section, we run the proposed approach on the Netflix data obtained from [32]. The Netflix dataset consists of more than 100 million ratings provided by 480K randomly-chosen, anonymous Netflix customers for 17K movie titles. The rating scale is from 1 to 5. In this experiment we report the root mean square error (RMSE) (20) and compare to [29] and [31].

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(p_i - q_i)^2}{N}} \qquad (20)$$

The results are shown in Fig. 9. Using harmonic analysis outperformed the reported results of [29] and [31] by 3% and 5% respectively. It is important to note that our method only relies on the provided ratings while the other two methods make use of context data describing the movies and the users.

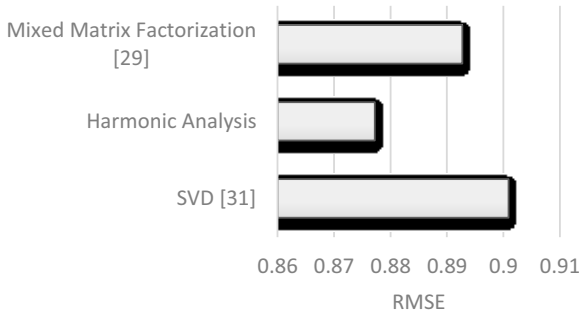

Fig. 9. RMSE measure for different approaches using Netflix dataset.

V. CONCLUSION

In this paper, we proposed a comprehensive coverage of a new multiresolution method for recommender systems based on Harmonic Analysis. The method improves the accuracy of systems with sparse user-item ratings. Experiments with both Movie Lens and Netflix data showed improvements in accuracy of the recommender system compared to conventional user-based and item-based collaborative filtering by 40%. Moreover, the proposed approach outperformed state of the art approaches in terms of accuracy and time per iteration. For future work, we will consider other features for improved recommendations such as taking context into account.

ACKNOWLEDGMENT

This work was made possible by NPRP 6-716-1-138 grant from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

REFERENCES

[1] HengSong Tan; HongWu Ye, "A Collaborative Filtering Recommendation Algorithm Based on Item Classification," *Circuits, Communications and Systems, 2009. PACCS '09 Pacific-Asia Conference*, pp.694-697, May 2009.

[2] Martinez, L.; Rodriguez, R. M.; Espinilla, M., "REJA: A Georeferenced Hybrid Recommender System for Restaurants," *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT '09. IEEE/WIC/ACM International Joint Conferences*, vol.3, pp.187-190, September 2009.

[3] Adomavicius, G.; Tuzhilin, A., "Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering 17,* pp.634-749, 2005.

[4] R. R. Coifman and M. Gavish, "Harmonic analysis of digital data bases," in *Wavelets and Multiscale Analysis,* Birkhäuser Boston, pp. 161-197, 2011.

[5] G. Badaro; H. Hajj; A. Haddad and W. El-Hajj. "A Multiresolution Approach for Recommender Systerms," *In Proceedings of the 8th Workshop on Social Network Mining and Analysis for Business, Consumer and Social Insights, SNA KDD*, 2014.

[6] B. Sarwar; G. Karypis, J. Konstan, and J. Riedl. "Item-based Collaborative Filtering Recommendation Algorithms," In Proc. *of the 10th International WWW Conference,* 2001.

[7] T. Hoffman; "Latent Semantic Models for Collaborative Filtering," *ACM transactions on Information Systems,* vol.22, no.1, pp.89-115, 2004.

[8] Umyarov, A.; Tuzhilin, A.; "Improving Collaborative Filtering Recommendations Using External Data," *Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on* , vol., no., pp.618-627, 15-19 Dec. 2008.

[9] Chen, Gang; Wang, Fei; Zhang, Changshui; , "Collaborative Filtering Using Orthogonal Nonnegative Matrix Tri-factorization," *Data Mining Workshops, 2007. arXiv preprint arXiv:1206.6439 (2012)2007. Seventh IEEE International Conference on*, vol., no., pp.303-308, 28-31 Oct. 2007.

[10] Shan, H.; Kattge, J.; Reich, P.; Banerjee, A.; Schrodt, F. and Reichstein, M., "Gap Filling in the Plant Kingdom---Trait Prediction Using Hierarchical Probabilistic Matrix Factorization," *arXiv preprint arXiv:1206.6439 (2012).*

[11] Zhong, E; Fan, W.; Yang, Q., "Contextual Collaborative Filtering via Hierarchical Matrix Factorization," *In SDM.*, pp.744-755, 2012.

[12] M. Jamali and M. Ester; ICDM 2011, Conference Tutorial, Topic: "Mining Social Networks for Recommendation," Simon Fraser University ICDM 2011*,* Dec. 12, 2011.

[13] Basiri, J.; Shakery, A.; Moshiri, B.; Hayat, M.Z.; "Alleviating the cold-start problem of recommender systems using a new hybrid approach," *Telecommunications (IST), 2010 5th International Symposium on* , vol., no., pp.962-967, 4-6 Dec. 2010.

[14] Claypool, M.; A. Gokhale; T. Miranda; P. Murnikov; D. Netes; M. Sartin, "Combining content-based and collaborative filters in an online newspaper," *ACM SIGIR'99, Workshop on Recommender Systems: Algorithms and Evaluation*, vol., no., Aug. 1999.

[15] Koren, Y., "Factorization meets the neighborhood: a multifaceted collaborative filtering model." *In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426-434. ACM, 2008.

[16] Hornick, M.F.; Tamayo, P.;, "Extending Recommender Systems for Disjoint User/Item Sets: The Conference Recommendation Proble," *Knowledge and Data Engineering, IEEE Transactions on,* vol.24, no.8, pp.1478-1490, Aug. 2012.

[17] Konstan, J.A.; Riedl, J.; , "Recommended for you," *Spectrum, IEEE,* vol.49, no.10, pp.54-61, October 2012.

[18] Wang, C., and M. Blei, D., "Collaborative topic modeling for recommending scientific articles." *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011.

[19] Purushotham, S., Liu, Y. and Jay Kuo, C-C., "Collaborative topic regression with social matrix factorization for recommendation systems." arXiv preprint arXiv:1206.4684 (2012).

[20] Quercia, D.; Lathia, N.; Calabrese, F.; Di Lorenzo, G.; Crowcroft, J., "Recommending Social Events from Mobile Phone Location Data," *Data Mining (ICDM), 2010 IEEE 10th International Conference on* , vol., no., pp.971-976, 13-17 Dec. 2010.

[21] Qi Liu; Yong Ge; Zhongmou Li; Enhong Chen; Hui Xiong, "Personalized Travel Package Recommendation," *Data Mining (ICDM), 2011 IEEE 11th International Conference on* , vol., no., pp.407-416, 11-14 Dec. 2011.

[22] Qingyan Yang; Ju Fan; Jianyong Wang; Lizhu Zhou; , "Personalizing Web Page Recommendation via Collaborative Filtering and Topic-Aware Markov Model," *Data Mining (ICDM), 2010 IEEE 10th International Conference on* , vol., no., pp.1145-1150, 13-17 Dec. 2010.

[23] Xin Jin; Spangler, S.; Ying Chen; Keke Cai; Rui Ma; Li Zhang; Xian Wu; Jiawei Han; , "Patent Maintenance Recommendation with Patent Information Network Model," *Data Mining (ICDM), 2011 IEEE 11th International Conference on* , vol., no., pp.280-289, 11-14 Dec. 2011.

[24] Tang, J., Wu, S., Sun, J., and Su, H., "Cross-domain collaboration recommendation." In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1285-1293. ACM, 2012.

[25] Armknecht, F.; Strufe, T., "An efficient distributed privacy-preserving recommendation system," *Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean*, pp.65-70, 12-15 June 2011.

[26] Jinoh Oh; Sun Park; Hwanjo Yu; Min Song; Seung-Taek Park; , "Novel Recommendation Based on Personal Popularity Tendency," *Data Mining (ICDM), 2011 IEEE 11th International Conference on* , vol., no., pp.507-516, 11-14 Dec. 2011.

[27] Coifman, R.; Lafon, S.; Lee, A.; Maggioni, M.; Nadler, B.; Warner, F. and Zucker, S., "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *In Proceedings of the National Academy of Sciences of the United States of America 102*, no. 21 pp. 7426-7431. 2005.

[28] "MovieLens Data Sets": http://www.grouplens.org/node/73, May 5, 2012 [Aug. 24 2013]

[29] Mackey, L. W.; Weiss, D. and Jordan, M., "Mixed membership matrix factorization," *In Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 711-718. 2010.

[30] Treerattanapitak, K. and Chuleerat, J., "Exponential fuzzy C-means for collaborative filtering," *Journal of Computer Science and Technology 27*, no. 3 pp. 567-576. 2012.

[31] Koren Y, Bell R. Advanced in collaborative filtering. *In Recommender Systems Handbook* (1st edition), Springer, 2011, pp. 145-186.

[32] "Netflix Dataset": http://www.lifecrunch.biz/archives/207, 2012 [Oct. 3 2014]