Research Journal of Applied Sciences 11 (10): 1143-1150, 2016
ISSN: 1815-932X

Design 3D-SVDs Algorithm for Location Based Recommendation System

[1]Tawfiq A. Alasadi and [2]R. Wadhah Baiee
[1]Dean of Information Technology College, Babylon University, Hillah, Iraq
[2]Department of Software, College of Information Technology, Babylon University, Hillah, Iraq

Abstract: Mobile devices are widely used todays with a huge number of applications usage that support users agreements. Rating matrices represent the users' behaviors through the case study timeline and have some limitations like high dimensionality. Here, Singular Values Decomposition (SVD) is used to reduce feature space dimensionalities but with developed techniques. The paper proposes 3D-SVDs algorithm which splits the tracks and although the inducted rating matrices into multi-level data segments each one represents one period of time slices within system data. The extracted latent features from each level of 3D-SVDs are used to checking user similarities to his neighbors and then the system picks the group of nearest users to recommend their similar preferences to the current user. The system uses most frequent item recommendation technique to recommend best positions to the user from the latent grouped nearest user's preferences in addition to using current user place from GPS as a new combination function to enhance recommendation list. The system recommends the daily preference places for users. Finally the results are shown on map supported application. The proposed system is built by using C#.NET and ArcMap GIS oriented programming for desktop version and Android Java for mobile devices version of the same system.

Key words: Rating matrix, recommender system, SVD, 3D-SVDs GPS tracks, GIS

INTRODUCTION

Mobile devices are wide used nowadays and their popularity spreads precipitately. Communication, applications, functionalities and challenges of these devices are rising every year. Users deal with mobile devices to facilitate their requirements under any circumstances. Many applications are developed for that reason and datasets become vast and varied, these data must be organized, analyzed and processed to be meaningful to the user. Traveling is an important area of mobile applications and an inconceivable number of facilities are now obtained to serve the users while they travel. It is important to know the capabilities of this filed and study the behavior of mobile users. GPS Trajectories have presented unique information to understand moving items and places, calling for regular research and improvement of new computing techniques to process, retrieve and mine trajectory data and discovering its applications.

Baltrunas *et al*. (2012) take a new approach for modeling the association among contextual features and user-item ratings. Instead of using the traditional method to data collection, they simulate contextual circumstances to more simply capture data concerning how the context affects user ratings. Zheng *et al*. (2010a, b) they model the users' location and activity histories that are taken as feedback to the system. they mine knowledge such as the position features and activities activities relationships from the Web and GIS databases to collect additional inputs (Mac *et al*., 2009). This research introduces an approach which observes user's activity and generates a user profile reflecting his information desires based on the interactions of the user with the system, physical location and user movements. The system recognizes the user profile and adjusts consequently both to provide appropriate information. Savage *et al*. (2012) implement a more complete universal location based recommendation algorithm that by gathering user's preferences and take into account geography of time and comparison measurements automatically. Martinez introduces a restaurants guide hybrid RS, they used both collaborative recommender system and knowledge based recommender system, this mergence is able to offer suggestions in any state of user requirements; also it offers metadata stated by the Maps of Google, regarding the recommendations. REJA is the name of the proposed system. This system is consists of hybrid RS model so as to escape from

Corresponding Author: R. Wadhah Baiee, Department Software, College of Information Technology, Babylon University, Iraq

the cold start challenge of the former models by using knowledge based procedure that uses fuzzy linguistic preferences relation. Huang and Gartner (2012) studies the Context-Aware Collaborative Filtering (CA_CF) and the way that it is presented by in the smart devices as a user guidance. The study focuses on how the (CA_CF) be applied on the existing GPS trajectories to advance users with context awareness POIs recommendations. Two stages are offered to find circumstances constraints that related and wanted to be demonstrated in a (CA_CF) approach. Then they discover a statistic method to calculate similarities among various conditions. With these (CA_CF) methods, smart services like get people who similar to you can be provided.

This study introduces a new algorithm to enhance the usage of SVD algorithm, by supposing 3D-SVDs which depends on time periods of sub-trajectory inducted ratings, then build optimum SVD matrix from previously calculated time level SVDs. The proposed system uses this algorithm to enhances and produces POIs and new route recommendation. And also implement and design cooperative application, it is built upon GIS databases and it can be an Add-On extension to ArcMap GIS software and the results of recommendation can be shown on desktops and/or mobile devices for each user. The recommendations are displayed on digital maps.

The proposed system uses GeoLife Trajectories dataset by Zheng *et al.* (2010a, b, 2009), this GPS dataset was composed in (Microsoft Research Asia) by 182 user in a period of over 5 year (2007-2012). A trajectory of this dataset is denoted by points sequences; each one has the information of (latitude, longitude and altitude). The system has been built by using C#.NET and ArcMap GIS object oriented programming to generate implicit feedback from GPS tracks.mailto:C@.NET The proposed system create a list of recommendations to the users, it consists of preference places that can be visit by the users according to their tracks histories and placement of his query. The recommended places are the daily used places like work, home, common roads and restaurants.

Dimensionality reduction: There're fundamentally two methods for producing dimensionality reduction to enhance recommendation system.

User-item rating matrix decomposition: Common dimensional reduction method to items recommendations is the Latent Semantic Indexing (LSI). For this technique, $|U|\times|I|$ users/items matrix of ratings R with rank n is estimated as a matrix R = PQT of rank k<n, so P is a $|U|\times k$ matrix of users features and Q is a $|I|\times k$ matrix of object features. Naturally, the uth row of P where $p_u \in R^k$, denotes the indexes to the user u that is estimated in the new k dimensions LS. Similarly, the ith row of Q where $q_i \in R^k$, may be understood as indexes to the item i in this LS. The two arrays Q and P are typically established by applying minimization on the rebuilding error identified with the Squared Frobenius Norm:

$$\begin{aligned} err(P, Q) &= \| R - PQ^T \|_F^2 \\ &= \sum_{u,i} (r_{ui} - p_u q_i^T)^2 \end{aligned} \tag{1}$$

To find the Singular Value Decomposition (SVD) the above error in Eq. 1 should be minimized (Desrosiers and Karypis, 2011).

Sparse similarity matrix decomposition: The idea behind the other dimensions reducing method is similar to the first method with some difference. It is simply analyze the matrices into their primary factors to apply projections of items and persons in the new hidden space. Though, in place of decomposing fully informed matrix of ratings, a sparse similarity array is processed (Desrosiers and Karypis, 2011).

MATERIALS AND METHODS

Matrix factorization models: Latent factor model uses collaborative filtering technique beside the general objective for exposing hidden features that clarify illustrated rating; for instance, Neural Networks, Latent Dirichlet Allocation and some techniques which are brought by ratings matrix factorizations applied on users-items matrices, similarly identified as SVD based models. Matrix factorization model has gained acceptance according to their good scalability and accuracy. For the recovery of information, SVD used to find latent semantic factors (Koren and Bell, 2011).

Singular value decomposition: In linear algebra, every $m\times n$ matrix may be factorized like follows:

$$A = U\Sigma V^T \tag{2}$$

In the above equation U is the symbol of $m\times m$ orthogonal matrix, the columns represent the values of eigenvector of matrix AA^T, V is the n×n orthogonal matrix where the columns represent the values of eigenvector of A^TA and Σ is an m×n diagonally matrix in the flowing form:

$$\sum = \begin{pmatrix} s_1 & & & & 0 \\ & s_r & & & \\ & & \dots & & \\ & & & 0 & \\ 0 & & & & 0 \end{pmatrix} \quad (3)$$

With σ1 ≥σ2≥...≥σr>02 σr > 0 and r = rank(A). In the above, σ1, ..., σr are the eigenvalues square roots of the matrix A^TA. These values are named singular values of A.

Singular value decomposition is a great method to deal with sets of matrices or equations which may be singular or mathematically exactly near to be singular. Several approaches that use Gaussian elimination decomposition fail to provide acceptable results; singular value decomposition not only identifies the challenge but likewise gives useful answers in numeric form. Likewise, this technique supports the choice for solving many linear least squares problems.

Singular value decomposition methods work with A^TA and AA^T matrices; they are symmetric, also they contain n and m orthogonal eigenvectors. Thus, next decompositions are:

$$A^TA = VDV^T \quad (4)$$

$$AA^T = UD'U^T \quad (5)$$

Here V represents n×n orthogonal matrix contains eigenvectors of the A^TA, D represents n×n diagonal matrix which owns eigenvalues of the A^TA on the matrix diagonal. U represents m × m orthogonal matrix contains eigenvectors of AA^T and D' represents m×m diagonal matrix of eigenvalues of the AA^T matrix also on its diagonal. It highlights on such case of D and D' have similar no-zero values in the diagonal except which the order could be alternative (Koren and Bell, 2011). From following:

$$\underset{m\times n}{A} = \underset{m\times m}{U}\underset{m\times n}{\Sigma}\underset{n\times n}{V^T} \quad (6)$$

SVD has numerous facts as shown below:

- The ranks of A and Σ are equal to r
- The first r columns in U are extended over the columns in A
- The last n-r columns in V are extended over null space in A
- The first r columns in V are extended over Row space in A
- The last m-r columns in U are extended over null space of A^T

SVD training: Matrix factorization models map objects and users together into a related space of hidden factor in the case of f dimension; the user-object interaction is modeled as inner product in mentioned space. LS attempts to clarify ratings by symbolizing the objects with the users on features routinely concluded from users feedbacks. As an instance while the item is film, features could measure clear dimensions like action versus comedy, quantity of actions or it is for children; fewer good cleared dimensions like deepness of the actor progress or totally interpretable dimension.

Therefore, every object i is connected with a vector qi∈Rf and every user u is related to vector pu∈R. For the object i, features of qi calculate how far the item holds these factors, positively or negatively. To the user u, the components of pu calculate how the user is interested with items that have great values on the corresponding features (Koren and Bell, 2011).

The results are dot products $q_i^Tp_u$, they capture the interactions related of user u to the object i, e.g. (the total interests of the users in the features of items). The last ratings are generated by adding in the baseline predictor that depends merely on the users or items. Therefore, ratings are expected by Eq. 7:

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u \quad (7)$$

According to learning the model parameters (b_i, b_u, P_u and q_i), Eq. 10 minimizes the regularized squared error:

$$\min_{b,q,p} \sum_{u,i\in k} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda_4 (b_i^2 + b_u^2 + \|q_i\|^2 + \|q_u\|^2) \quad (8)$$

The value of λ_4 controls the degree of regularization is regularly identified by cross validation. Minimizing process is naturally completed by alternating least squares or stochastic gradient descent.

The algorithm will continue looping over all rating values in the dataset of training. In every rating r_{ui} a prediction is ($\hat{r}_{ui}$) completed and the related prediction error e_{ui} def = r_{ui}- $\hat{r}_{ui}$ is calculated. For a certain training case r_{ui}

the system allow to adjust the parameters by proceeding in the reverse path of the gradient to produce (Takacs *et al*., 2008):

$$\begin{aligned} b_u &\leftarrow b_u + \gamma.(e_{ui} - .\lambda_4.b_u) \\ b_i &\leftarrow b_i + \gamma.(e_{ui} - .\lambda_4.b_1) \\ q_i &\leftarrow q_i + \gamma.(e_{ui}.pu - .\lambda_4.q_1) \\ p_u &\leftarrow p_u + \gamma.(e_{ui}.q_i - .\lambda_4.p_u) \end{aligned} \tag{9}$$

Applying SVD for collaborative filtering: Sparsity dataset lead us to discover different recommender system techniques. The idea starts by a method that tried to tie the sparsity by including agents worked with semi intelligent filtering technique into the system. Those agents estimated and rated every item by using syntactic characteristics. Via supporting dense rating sets, they assisted improve coverage and better quality.

Through decreasing the dimensionality of the item space, the system can increase density and thus get additional ratings. Detecting of hidden relationships from the dataset could possibly resolve the synonymy problem in recommender systems. Latent semantic indexing uses singular value decomposition as its original matrix factorization algorithm, maps kindly to the CF recommendation algorithm task. Singular value decomposition is functioned in recommendation system to accomplish two variant tasks (Sarwar *et al*., 2000).

- Singular value decomposition is used to extract hidden relationships among customers and items that permit the systems to calculate the predicted likelihood of a specific item by a user
- Singular value decomposition is used to create an original user-item space representation with small number of dimensions and then calculate neighborhood in the compact space. Then, it can be used to produce the list of top-N item recommendations for users

Proposed algorithm: The main part of the proposed system is calculating Singular Value Decomposition (SVD) and recommends places for user. After inducting and creating rating matrices of users-places relation that matrices become the input to the current stage of process. As mentioned above the SVD algorithm has the functionality of dimensions reduction because if we need to extract features among users and POIs and we have this huge data of ratings, the mechanism of dimensions reduction is the right choices for them.

The reduced data become easy to store out because the storage that is needed to save for example as we have in the system about user's record each one should have 10,000 fields so we have about 1,820,000 if fields if we want to deal with the total amount of data. For this reason SVD used to reduce the the feature space and down the number of dimensions of features to two dimensions by using SVD spherical dimension projection.

At this point the system has less feature space data which can be stored easily and also can be used later to predict and return back the values of semi-original rating matrix without storing the original matrix anywhere. The rating matrix is inducted from the user tracks histories of GPS, it is calculated from preprocessing step. The rating matrix consists of rating (Ricci *et al*., 2011; Lee and Krumm, 2011; Baltrunas *et al*., 2012; Zheng *et al*., 2010a, b; Aoidh *et al*., 2009; Savage *et al*., 2012; Huang and Gartner, 2012; Zheng *et al*., 2010a, b, 2009) for each place that is inducted from feedback system depending on GPS tracks.

Also, SVDs has the two main matrices as mentioned in section two U and V^T where U is the User features matrix against the places or POIs in the map and V^T is the feature matrix of Places against users in the map. The system additionally appoints the U matrix to calculate the nearest group of users to the studied user which is used later to recommend the appropriate places on the digital map's mobile for him as a part of our system results.

SVD is used in new manner by proposing 3D-SVDs three dimensional singular value decomposition; this is for adding time functionality to the algorithm itself.

Making SVDs have tempo features beside the spatial features that are calculated from normal SVD will add some properties to the prediction accuracy and also enhance the recommendation decisions that listed to the users on the map in their mobile devices. Figure 1 illustrates the main steps of recommendation process stage of the proposed system.

Calculating SVDs and feature extraction: The idea behind the proposed system is to recommends a list of places that user needs to be the next places he would go to from his current place.

Algorithm of 3D-SVDs:

Inputs: Rating Matrices
Outputs: Feature Vectors for Users
Process:
Begin
    For each period of time do
        Calculate SVD as: $A' = U\Sigma V^T$
        {for SVD training use Eq. 9}
        Trunk S singular values using threshold value
        Prepare U matrix for current period
End

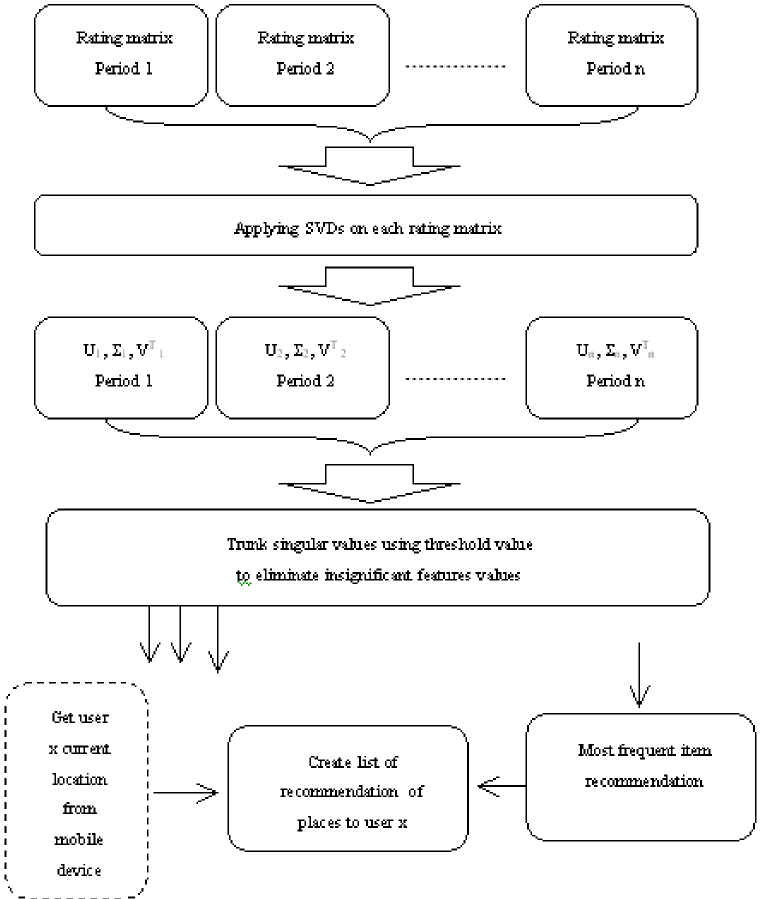

Fig. 1: Block diagram of recommendation stage

The recommendation is calculated in this system using proposed 3D-SVDs algorithm. The features that are produced in this algorithm are spread on time line of the case study. Each user will have its relevant vector of features for each period of time, so the system calculates the similarities of users depends on each level of time line and then calculates the recommendation from the POIs that the nearest users are liked or visits from their features also. Figure 2 shows a sketch of how 3D-SVDs are formulated in the proposed system.

Fig. 2: 3D vector space of user features in 3D-SVDs

Finding similarities: The next step to find recommendation is to applying similarities on user features vectors that are produced from the previous section. Each user x in the data space of the proposed system has multilevel vector of features each vector from U_i which is the matrix of user features that is came from SVDs from this point the system can calculate similarities of one user against other users to find from their movements histories the appropriate recommendations to the current user.

The Cosine similarity is adopted here in this step to find similarities among users. As mentioned previously,the user which the system want to calculate his similarity to the others, the system takes the vector of features from each time slice from the previous step and

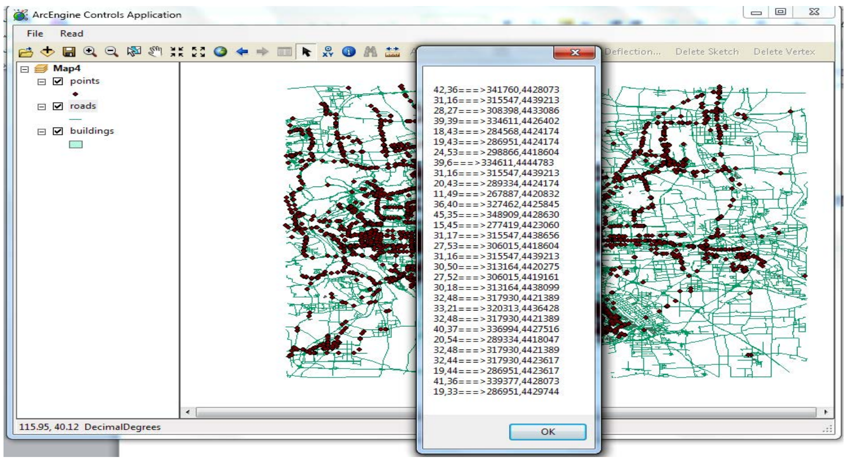

Fig. 3: List of recommended regions into user interactive interface of windows version of the system

calculate the similarity in the same period then the system saves the results temporarily to used later with other similarity results from other time slice feature space to calculate nearest neighbor by combining the similar users in one results.

Algorithm of finding similarity:

Algorithm 2: Finding similarities
Inputs: 3D-Vector of features, User x
Outputs: Ordered list of similar users
Process:
Begin
For each matrix U_i (Features Matrix) do
Loop the vectors in U_i except vector of User x
Apply the equation:

$$CV = (u,v) = \cos(x_u, x_v) = \frac{\sum_{i \in Iuv} r_{ui} r_{ui}}{\sqrt{\sum_{i \in Iuv} r_{ui}^2 r_{vj}^2}}$$

Store the similarity in vector V_i
Sort by descending the vector V_i
Store the ordered vectors as a matrix of similarity
End

The algorithm in Fig. 3 explains the mechanism that used to find similarities from vector of features. The nearest users that have values near 1 while the farther ones have a value near -1. In the proposed system we used a threshold value to cut off the list of similar users, so the users who have a value over than 0.6 they become in the search space and their rating data will be used to find the recommended list of POIs as will be illustrated in the next section.

Most frequent item recommendation: The last part of the proposed recommended system is to finding the list of recommended items of the current user. From the last section we have ordered vectors of similarities among users against the studied user, these vector is used to find the nearest neighbor users to the current user, so after applying the most frequent item recommendation technique we will have the most visited places for each similar user then the system can sort the result to find the top N values which are the recommended places for the current user.

Figure 3 illustrates the algorithm of main steps of listing recommendation to the user from nearest users. The choice of the POIs depends on two main criteria:

- The relation of current user to other users and this will produced by features that extracted from rating matrix and user histories
- The current location of the user which has been got from the GPS sensor in his mobile device

These two features should combined together to have more accurate and enhanced recommendations in the proposed system. This combination will produce recommended POIs near the user and the word near means-after applying this technique-it means two meanings: first is the places that other users who have high similarity to the current user are more likely to the current user as like as them, second, the near places should be taken into account to filter and recommend a really near places to the current user according to his current location.

The second criteria is calculated using Manhattan Distance 16 the system uses Manhattan Distance to calculate the similarity of current location of the user and the locations that are more likely to the users and produced from the most frequent item recommendation.

Manhattan distance is used because the system depends on a map that is segmented to a regions-rows and columns-so, this distance measure is the right tool for this type of data.

Algorithm of listing recommendation:

Algorithm 4: Recommendation
Inputs: Ordered list of similar users, Map, User location
Outputs: List of recommended POIs near the user
Process:
```
Begin
    Get point p from User location
    For each period of time do
        For each similar user ui in (List of similar users) do
            Sort the most visited places by ui in descent
            Take the top 10 most frequent places
        For each near place do
            Dist←Manhattan distance from p to top 10 places
            Score←Rates of most visit places
            If this places (Dist<v1) and (Score>v2)
                Add this place to List placeList with Dist and Score
    Sort placeList descending for Score and Dist values
    Take top 10 places from placeList
    Draw their locations on map mobile application
End
```

RESULTS AND DISCUSSION

The huge dimensionality of data can be reduced by using SVDs as it is mentioned above. The rating matrices from the previous stage of the proposed system are the input to this stage of processing, absolutely there are several rating matrices to be processed. Each period of time should has its own rating matrix as we proposed in this dissertation. We propose 12 periods of time through the five years of recording user movements, each period represents five months of user histories. After applying the proposed algorithm for feature extraction of 3D-SVDs the proposed system will introduces feature space of the system data which are used later to calculate user nearest neighborhood and predict desired places to the user according to his histories and his location.

Table 1 shows samples of the 3D-SVDs, each table represents the result of one slice of time period from the twelve periods that are proposed to the system. Here we select 4 singular values to reduce the details of user vectors of features length to 4.

From the samples above the system calculates similarities among users and then sorts the results to find near neighbors for the user who is using the system now. Table 2 shows the results of finding cosine similarities between user number 1 against other 5 users in each period of time, so the system find that users relations are differs through time.

Table 1: Sample of 10 users vector of features, period 1

	Features			
Users	1	2	3	4
1	-3810.811	535.370	-151.001	-22.697
2	-511.313	-755.282	431.253	-141.222
3	-1642.150	-1483.411	1989.345	125.128
4	-12914.259	2852.882	-1487.315	-418.471
5	-8614.926	1338.770	-265.280	466.124
6	-1415.277	-471.756	361.753	67.476
7	-1090.654	-339.623	109.324	71.593
8	-1112.895	-202.123	325.023	98.916
9	-1260.106	-663.842	578.453	-324.621
10	-1432.265	-620.232	579.043	310.416

Case study: For example we have the current user which is No. 1 has been logged into the system he want to know: what are the best places he should go in the city of the case study (Beijing).

The system picks up the user's current place within the region (30, 45) and the system calculates cosine similarities for user No. 1 against the other users to know who users are nearby features that are calculated from 3D-SVDs. Recall that 3D-SVDs will produce vector of features to the users from their movements history.

By knowing the nearby users depending on extracted features an cosine similarity, recall Table 2, the system take the (Top N) users, this is the nearest users to the current user according to the rating histories, here the value of N is selected by trial and we choose N = 6. The system has 6 nearly users to the user No. 1 for each period of time. That, now we have 6×12 = 72 nearly users by using 3D-SVDs.

This technique will spread the feature space on the time slices instead of depending on top N nearest users that can be calculated using the original algorithm. The system calculates the most frequent items for each user and take the (Top N) places that are most visited by each user, these places are the most visited places by each user, here we choose N = 10 by trial. From the near 10 places and 72 nearly users, the system have 10×72 = 720 POIs or regions. Those regions are sorted depending on the most frequently visited scores saved to the next level of filtration.

Table 2: Calculating cosine similarities for user number 1 for all periods

Periods	12	11	10	9	8	7	6	5	4	3	2	1
User 1	1	1	1	1	1	1	1	1	1	1	1	1
User 2	0.659	0.889	0.6629	0.8356	0.5774	0.8772	0.7637	0.889	0.7707	0.9188	0.6686	0.3778
User 3	0.8351	0.9334	0.8291	0.793	0.7215	0.8681	0.8279	0.9373	0.208	0.2734	0.8509	0.4499
User 4	0.9886	0.8673	0.9754	0.9989	0.9956	0.9843	0.7364	0.9677	0.9748	0.9645	0.6277	0.994
User 5	0.9954	0.8733	0.9891	0.9961	0.9972	0.9722	0.8588	0.9875	0.7561	0.9454	0.5609	0.9981

The second type of filtration and checking similarities is the Manhattan distances check. This type is proposed here to merge the two technique into one application. The user current place is very useful and most significant to recommend the next place to go. From the 720 regions that are detected from nearest neighbor previously, each region will be calculated against the current user region as the running example (30, 45) this region has the similarities to the other regions as shown in Table 2, the places sorted by using the low distance values are the best, then the system choose the (Top N) regions, these regions are the nearest filtered places according to the current user placed where he is query the recommended list and here N = 20. Figure 3 illustrates a list of recommended regions with the point of interest vales that are related to them.

CONCLUSION

In the proposed system we developed a new mechanism for using singular values decomposition algorithm, we suggested 3D-SVDs feature extracting algorithm to dealt with rating matrices in each period of time and produces layers of features each layer represents time period. Similarities are calculated among users to predict and calculate recommendation of POIs to the users.

The recommendations which are the main results of the proposed system are produced from the combinations of historical movements behavior of the users that are extracted by 3D-SVDs and the current location of user who uses the system to avoid the redundant results and to be more accurate recommended POIs to him.

Last, ArcMap GIS and its relevant data with GPS tracks are the core of developing our proposed system and there are two versions of the system: desktop and mobile application to be used by the users. Recommendations are displayed to the user on the map directly and the POIs are signed directly with shortly directions and roads to visualizing the recommendation lists to the users.

REFERENCES

Baltrunas, L., B. Ludwig, S. Peer and F. Ricci, 2012. Context relevance assessment and exploitation in mobile recommender systems. Personal Ubiquitous Comput., 16: 507-526.

Desrosiers, C. and G. Karypis, 2011. A Comprehensive Survey of Neighborhood-based Recommendation Methods. In: Recommender Systems Handbook, Ricci, F., L. Rokach, B. Shapira and P.B. Kantor (Eds.). Springer, New York, USA., ISBN-13: 9780387858197, pp: 107-144.

Huang, H. and G. Gartner, 2012. Using Context-Aware Collaborative Filtering for POI Recommendations in Mobile Guides. In: Advances in Location-Based Services, Georg, G. and F. Ortag (Eds.). Springer, Berlin, Germany, ISBN:978-3-642-24197-0, pp: 131-147.

Koren, Y. and R. Bell, 2011. Advances in Collaborative Filtering. In: Recommender Systems Handbook, Ricci, F. (Ed.). Springer, Berlin, Germany, ISBN:978-0-387-85819-7, pp: 145-186.

Lee, W.C. and J. Krumm, 2011. Trajectory Preprocessing. In: Computing with Spatial Trajectories, Yu, Z. and X. Zhou (Eds.). Springer, New York, USA., ISBN:978-1-4614-1628-9, pp: 3-33.

Mac, A.E., G. McArdle, M. Petit, C. Ray and M. Bertolotto *et al*., 2009. Personalization in adaptive and interactive GIS. Annal. GIS., 15: 23-33.

Ricci, F., L. Rokach and B. Shapira, 2011. Introduction to Recommender Systems Handbook. In: Recommender Systems Handbook, Ricci, F., L. Rokach, L. Shapira and P. Kantor (Eds.). Springer, Berlin, Germany, ISBN:978-0-387-85819-7, pp: 1-35.

Sarwar, B.M., G. Karypis, J.A. Konstan and J. Reidl, 2000. Application of dimensionality reduction in Recommender system: A case study. Web Mining for E-Commerce Workshop, ACM WebKDD 2000, pp: 8-90. http://robotics.stanford.edu/~ronnyk/WEBKDD 2000/papers/sarwar.pdf.

Savage, N.S., M. Baranski, N.E. Chavez and T. Hollerer, 2012. I'm Feeling Loco: A Location Based Context Aware Recommendation System. In: Advances in Location-Based Services, Gartner, G. and F.Ortag (Eds.). Springer, Berlin, Germany, ISBN:978-3-642-24197-0, pp: 37-54.

Takacs, G., I. Pilaszy, B. Nemeth and D. Tikk, 2008. Matrix factorization and neighbor based algorithms for the netflix prize problem. Proceedings of the 2008 ACM Conference on Recommender Systems, October 23-25, 2008, ACM, Lausanne, Switzerland, ISBN: 978-1-60558-093-7, pp: 267-274.

Zheng, Y., X. Xie and W.Y. Ma, 2010a. GeoLife: A collaborative social networking service among user location and trajectory. IEEE. Data Eng. Bull., 33: 32-39.

Zheng, V.W., Y. Zheng, X. Xie and Q. Yang, 2010b. Collaborative location and activity recommendations with GPS history data. Proceedings of the 19th International Conference on World Wide Web, April 26-30, 2010, ACM, Raleigh, North Carolina, ISBN:978-1-60558-799-8, pp: 1029-1038.

Zheng, Y., L. Zhang, X. Xie and W.Y. Ma, 2009. Mining interesting locations and travel sequences from GPS trajectories. Proceedings of the 18th International Conference on World Wide Web, April 20-24, 2009, ACM, Madrid, Spain, ISBN:978-1-60558-487-4, pp: 791-800.