An Enhanced Memory-Based Collaborative Filtering Approach for Context-Aware Recommendation

Guan-Yu Tseng Wei-Po Lee

Abstract—**Recommender systems have been advocated in different domains for years and there are various recommender systems developed in different application services. In addition to using recommendation techniques, it is helpful to employ contextual information in determining the relevance of an item to a user's needs. To enhance recommendation performance, we present in this study a new approach that, in a direct way, integrates different types of contextual information and user ratings with computational methods. To verify the proposed approach in making collaborative recommendations, we conduct a series of experiments to evaluate its performance. The results show that the proposed context-aware method outperforms other conventional approaches. Moreover, we implement a mobile recommendation system on a cloud platform to show that our approach can be used to develop a real-world application.**

Index Terms—**recommender system, collaborative filtering, contextual information, mobile multimedia, nearest neighbor**

I. Introduction

Recommender systems have been developed in different application services [1,2]. Traditional recommender systems address two entities, *users* and *items*, for application services. Initially, the systems collect some ratings specified by users. Based on these records, the systems try to estimate a rating function. Once the function is constructed for the entire *Users* × *Items* domain, a system can recommend the items with the highest ratings to the users. In practice, however, it is not necessary to estimate the unknown ratings of the items with the highest ratings or the entire *Users* × *Items* space beforehand because this is an expensive task for applications with large numbers of users and items. Instead, various methods have been developed to find efficient solutions that require less computational effort. These methods range from content-based user modeling to group-based collaboration. Generally, the group-based approach is more efficient and effective than content-based user modeling [2,3].

In addition to recommendation techniques, context plays an important role in determining the relevance of an item to a user's needs and is useful to achieve service personalization. As indicated in [4,5,6], incorporating contextual information in computational methods to make better recommendations, the classical two-dimensional *Users*×*Items* recommendation domain is extended to a multi-dimensional model: *Users* × *Items* × *Contexts*. The recommendation problem is thus to estimate the new rating function *R*: *Users* × *Items* × *Contexts* → *Ratings* (or equally, to estimate the unknown rating values of this multi-dimensional model through the available entry values).

In this work, we present a new approach that incorporate contextual information with the common collaborative filtering method (i.e., memory-based method) to enhance the prediction performance. Unlike most of the context-aware recommender systems that induce separated context rules from rating data as preconditions for choosing appropriate items, we instead develop a straightforward method to embed the contextual information in the computation procedures of collaborative filtering to improve the recommendation performance. The details are described in the sections below. For the memory-based method, our approach combines contexts and user preferences as a multi-feature vector and uses it to measure similarity. To verify the proposed approach of collaborative recommendation with contextual information, we conduct a series of experiments to evaluate performance. The results show that using contexts is beneficial to item recommendation, and the proposed context-aware methods outperform other conventional ones.

II. Background and Related Works

As mentioned above, due to the tremendous amount of digital items, a recommendation mechanism is needed to offer better services. In general, the recommendation techniques can be categorized into three types: content-based, collaborative filtering, and hybrid methods [1,7]. The content-based approach predicts the user's preferences for new items based on historical records. Therefore, the most important issue in this approach is constructing a computational model to perform the prediction. Nevertheless, the content-based approach largely relies on sufficient samples to construct the model. This approach often recommends items within a specific scope and thus loses item diversity.

In contrast to the content-based method, the collaborative filtering (CF) method does not build a personal model for prediction. In general, there are two major techniques to perform CF methods: memory-based methods (nearest neighbor methods) and model-based methods (latent factor models) [1,3]. The memory-based methods recommend items to the user according to the evaluations of other users with similar tastes (or recommends items similar to the ones with high user ratings in a similar way). In such an approach, therefore, the measurement of similarity between users is most important so that the system can employ a *k*-nearest neighbor method to find the most similar users to perform the

G.-Y Tseng and W.-P. Lee are with Department of Information Management, National Sun Yat-sen University, Kaohsiung 80424, Taiwan. Email: {wplee@mail.nsysu.edu.tw}

ISBN: 978-988-19253-4-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

recommendation. The system's prediction of a new item for a user is thus based on a combination of the ratings of the user's nearest neighbors. This approach has been widely used in different applications, for example [6,8].

The second type of CF technique, a model-based method provides an alternative by transforming both users and items to the same latent factor space. This space explains the ratings on several implicit factors obtained automatically. The intuition behind this method is that there should be some latent features that determine how a user rates an item (with the assumption that the number of features would be smaller than the number of users and the number of items). Therefore, if we can discover these latent features, we should be able to predict a rating regarding a certain user and a certain item because the features associated with the user should match the features associated with the item. Different algorithms have been proposed to derive these factors by minimizing the discrepancy between the predicted ratings and the observed ratings (e.g., [9,10]).

The above memory-based methods are very popular because they are intuitive and relatively simple to implement. They also offer useful and important properties: explicit explanation of the recommendations and easy inclusion of new ratings [2,3,6]. Because our major goal here is to investigate how to use contextual information and to analyze its effect on recommendation, we thus choose this easy-to-implement approach in the experiments.

Context is an important issue to be considered in personalized recommendations. Any small contextual changes may lead the user to select a different service. Dealing with the context issue involves defining contexts relevant to the application service and identifying the key contexts in which people often use the service. Regarding different mobile applications, context factors can be defined as any information used to characterize the user situation that can influence his decision in requesting a service. There are two types of context factors: personal and environmental [11,12]. Personal context is the personal state or condition of the user himself (such as his emotional and physical states), whereas environmental context means the full set of a user's external circumstances (such as location, distractions, and crowds to indicate the geographical setting). Currently, the former is relatively difficult to measure, whereas the latter can be automatically detected and applied to several application domains, for example [12,13]. More extensive surveys on context-aware collaborative recommendations can be found in [5,13].

III. The Proposed Approach

A. System Framework

To achieve context-aware recommendations, we present a system framework with cloud-based client-server architecture. The server is constructed on the cloud to manage user profiles and perform computations for recommendations. Figure 1 illustrates the framework. Based on the collected contextual information and the user's ratings, the server component uses the collaborative filtering techniques implemented in the recommendation module to produce a candidate list from the current target items. This list is then sent to the user for his reference regarding item selection. Figure 2 is the interface shown on the client side. Users can provide the contextual information and their feedbacks (ratings) through the interface, and the system can take them into account to reason the candidate list. The following subsections describe the major part of the system —the recommendation module.

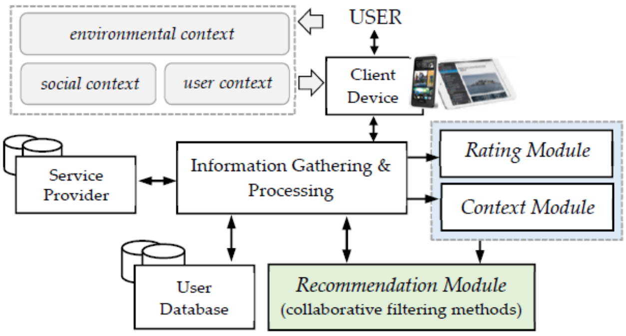

Fig. 1. The context-aware recommendation framework

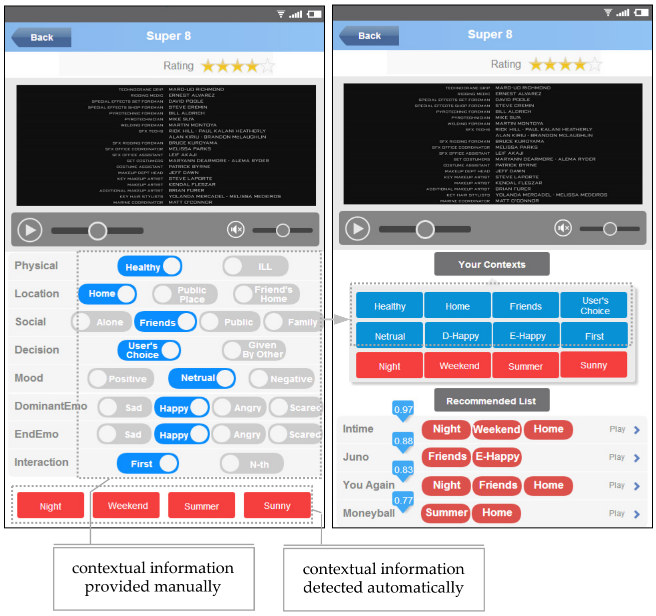

Fig. 2. The interface shown on the client side

B. Evaluation Criteria

In this work, the recommendation is evaluated by two standard criteria: the mean absolute error (MAE) and the root mean squared error (RMSE). MAE is the average of the absolute difference between the predicted and actual ratings over all items. It is defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|r_i - \hat{r}_i\right|$$

where r_i and and $\hat{r}_i$ are the actual and predicted ratings for item i, respectively, and n is the number of items. The other criterion, RMSE, squares and accumulates the differences between the actual and predicted results over all items, and then averages and roots the summation. More precisely, RMSE can be defined as:

ISBN: 978-988-19253-4-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - \hat{r}_i)^2}$$

In the collaborative filtering approach, the choice of using a user-based or item-based similarity measure is case-sensitive and depends on the application. In this work, we conducted a preliminary test for the major dataset with various contextual information. We found that better results can be obtained when the user-based model was used. This difference is mainly because this dataset has a relatively high co-rating rate making the user-based model suitable. The user-based similarity measure was thus adopted in the experiments.

C. Memory Based Context-Aware Collaborative Recommendation

Traditionally, many collaborative recommender systems have tried to predict the rating of an item for a particular user based on how other users previously rated the same item. This work adopts a memory-based approach for collaborative recommendation. Memory-based algorithms are heuristics that make rating predictions based on the entire collection of items previously rated by the users. That is, the value of the unknown rating $r_{u,i}$ for user u and item i is usually computed as an aggregate of the ratings of the top k most similar users for the same item i. There are many methods to calculate this similarity (e.g., Cosine similarity and Eucliclean distance). Here, the Pearson correlation coefficient is used. For two users x and y, the similarity between them is defined as:

$$Sim(x,y) = \frac{\sum_{i \in CoR(X,Y)} (r_{x,i} - \overline{r_x}) \times (r_{y,i} - \overline{r_y})}{\sqrt{\sum_{i \in CoR(x,y)} (r_{x,i} - \overline{r_x})^2} \sqrt{\sum_{i \in CoR(x,y)} (r_{y,i} - \overline{r_y})^2}}$$

In the above equation, $CoR(x, y)$ is the set of items that users x and y have already rated (i.e., the co-rated items). This coefficient is between 1 (the preferences of both users are the same) and -1 (their preferences are opposite one another); a value of zero means their preferences are not correlated. For a user u, users with the most similar preferences are selected as a set of neighbors $Neig(u)$, and their collective opinions on a certain item m are used to predict whether u will like the item. That is, the rating of the preference of a specific item m is defined as:

$$r_{u,m} = \overline{r_u} + w \times \sum_{n \in Neig(u)} Sim(u,n) \cdot (r_{n,m} - \overline{r_n})$$

In this equation, $r_{u,m}$ represents the predictive rating of user u on item m; $\overline{r_u}$ (or $\overline{r_n}$) is the average rating of user u (or user n) regarding all items he has rated. $Sim(u, n)$ is the similarity between two users u and n; $r_{n,m}$ is the rating of user n who is a neighbor of user u. Finally, w is the weighting factor that indicates the importance of each similar user. The weighting factor is often considered a normalized factor, which can be calculated as:

$$w = 1 \Big/ \sum_{n \in Neig(u)} \left| Sim(u,n) \right|$$

Because user preferences are changing over time, this user-based approach must repeatedly calculate the similarity of different users in real time to consider the most up-to-date referring opinions. To overcome the problem of computationally inefficient in calculating similarity, the item-based (item-item) model was proposed [14], which measures the similarity between items (rather than users). That is, in the above equations, the similarity measurement between two users u and n is modified to two items. The prediction performance is case-sensitive depending on the dataset used.

Following the original k-NN method for memory-based collaborative recommendations, we propose an approach (multi-dimensional k-NN, MDKNN) to incorporate both the contextual information and user ratings in the k-NN method to achieve context-aware recommendations. Then, a useful method (condensed multi-dimensional k-NN, CMDKNN) is presented for further performance enhancement.

The proposed approach considers the contextual information as data features and incorporates them in the original users' ratings for a similarity calculation. In this way, each single value rating provided by the user for a certain item is currently encoded as a multi-dimensional vector comprising various contexts and the rating. Therefore, in the equation of the Pearson correlation coefficient, both ratings $r_{x,i}$ and $r_{y,i}$ are extended as vectors where each context represents a feature dimension, in addition to the original rating. For example, in a dataset with n different types of context, the rating vector for item i by user x is represented as:

$$(Con_{x,i}^1, Con_{x,i}^2, ..., Con_{x,i}^n, r_{x,i})$$

where each Con^j ($1 \le j \le n$) is the feature corresponding to context type j. The $\overline{r}_x$ (and $\overline{r}_y$) in Pearson similarity is then extended to be a vector with the averages of all features accordingly, with the form of

$$(\overline{Con_x^1}, \overline{Con_x^2}, ..., \overline{Con_x^n}, \overline{r_x})$$

In the same equation, the discrepancy between the rating and the averaged rating for a co-rated item i (e.g., ($r_{x,i} - \overline{r}_x$) or ($r_{y,i} - \overline{r}_y$)) is substituted by the vector subtraction, which is

$$(Con_{x,i}^1 - \overline{Con_x^1}, Con_{x,i}^2 - \overline{Con_x^2}, ..., Con_{x,i}^n - \overline{Con_x^n}, r_{x,i} - \overline{r_x})$$

With these substitutions, the similarity and the rating prediction are then calculated accordingly. In general, the above context features can be scalar, ordinal, or categorical. To combine different types of features in the similarity calculation, we have defined cost matrices for categorical features (determined by a preliminary test) and performed normalization on different features for value aggression. In this calculation, weighting factors can be used to indicate the importance of contexts and user ratings (though it is presently not used in this work).

A useful computational technique has also been proposed to enhance the performance of the k-NN recommendation method. This method is based on the observation that in the traditional user-based model, when the co-rating rate of two

ISBN: 978-988-19253-4-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

users is low, an over-dominated situation occurs where the recommendation is only based on very few co-ratings. This basis results in a serious bias. Therefore, we present a modified similarity measure (condensed similarity) to alleviate such a situation. The newly defined similarity *S'* is:

$$S' = \frac{S \times |C_{u,v}|}{|C_{u,v}| + \alpha}$$

In this equation, S is the original similarity, $|C_{u,v}|$ is the condensing factor (indicating the number of co-ratings made by two users u and v), and α is a constant determined empirically by the size of the dataset. For example, it was set at 0.65 for the Comoda dataset in the experiments below.

IV. Experiments And Results

To assess the proposed approaches that incorporate contextual information with collaborative recommendations, in this section we describe the series of experiments conducted. These experiments used a dataset of multimedia items (i.e., Comoda dataset) to evaluate the performance of our approaches, including MDKNN and its condensed version CMDKNN.

A. Results of the MDKNN Method

The Comoda dataset was used in this series of experiments. At the time of use, this dataset includes 121 users, 1,620 items, and 2,296 ratings (for movies). This dataset has 12 types of contextual information: time, daytype, season, location, weather, social, endEmo, dominantEmo, mood, physical, decision, and interaction (for more details, refer to [15]). The performance evaluation can be conducted in two ways: a quantitative comparison of user preference prediction or a qualitative investigation of user satisfaction. The quantitative comparison focuses on the computational methods whereas the qualitative investigation focuses on the users' perspectives. Since our goal is to develop a more precise recommendation mechanism in a mobile multimedia recommender system, we adopted the first method (i.e., preference prediction) for the experiments.

The first set of experiments evaluated the effect of using contextual information with the users' ratings to calculate the user similarity in the *k*-NN collaborative filtering method. As described in section III.C, the original data similarity is currently replaced by a linear combination of context similarity and rating similarity. Different combinations of weighting factors for context (i.e., w_1) and rating (i.e., w_2) were tested. Figure 3 presents the MAE results of the four best combinations, where results of the original *k*-NN method (without using contexts) are also shown for comparison. As can be seen, the use of weighting factors $w_1 = 0.1$ and $w_2 = 0.9$ can provide the best results for this dataset. Compared with the original *k*-NN method, using users' ratings only (i.e., the results with weighting factors (0,1) shown on the left side of the figure), considering contextual information in a similarity measure captures the characteristics of rating data more precisely and thus improves the prediction performance.

In addition to the weighting factors, we conducted experiments to investigate the effect of the number of similar neighbors used for collective recommendation. In the experiments with various weighting factors, different numbers of nearest neighbors were used as reference points for decision making. The results are shown in Figure 3. We observe that in the five combinations listed in Figure 3, the cases with three nearest neighbors (i.e., k = 3) making the final decisions produce the best results. This number was subsequently used in the experimental trials. However, it is notable that the most suitable number of nearest neighbors for collaborative recommendation is dataset-sensitive and depends on the distribution of the original data.

In these experiments, RMSE was also calculated for each trial. The results are presented in Figure 4. Similar to the MAE shown in Figure 3, the cases with weighting factors w_1 = 0.1 and w_2 = 0.9 provide the best results, and using three nearest neighbors to predict ratings was again the most effective strategy for this dataset.

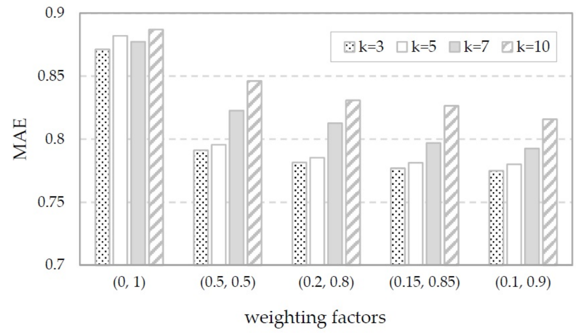

Fig. 3. Results of different combinations of weighting factors (w_1,w_2) and different k.

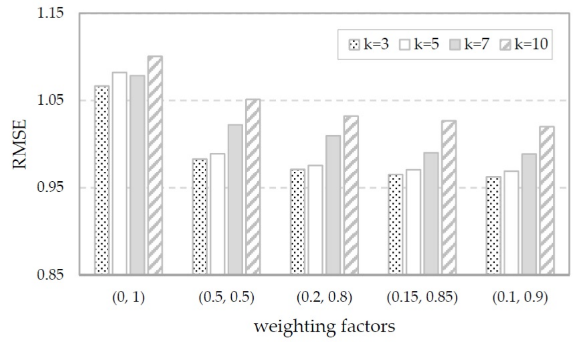

Fig. 4. Results of RMSE by different strategies.

B. Results of the CMDKNN Method

After showing that the developed MDKNN method incorporates contextual information with users' ratings to perform collaborative recommendations and produces better prediction performance, we conducted additional experiments to investigate the effect of the condensed factor presented in section III.C. As mentioned previously, this factor is introduced to modify the similarity measure to overcome the over-dominated situation that occurs when the co-rating rate between two users is low. In these experiments, the condensed similarity was used with the MDKNN method for rating prediction. The results are shown in Figures 5 and 6 where different numbers of nearest neighbors were considered in the trials. The weighting factors w_1 and w_2 for MDKNN are 0.1 and 0.9, respectively (which provide the

ISBN: 978-988-19253-4-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

best performance). The results indicate that the condensed similarity is a useful technique and can effectively reduce the error of MAE and RMSE for a *k*-NN based method.

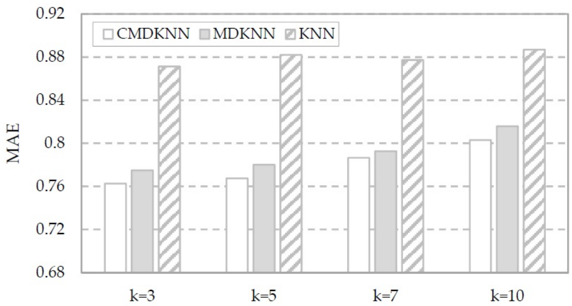

Fig. 5. Performance comparison (MAE) for *k*-NN, MDKNN, and CMDKNN

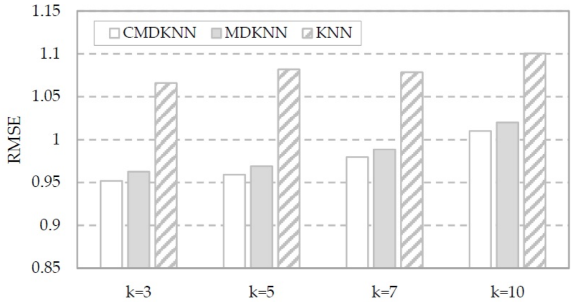

Fig. 6. Performance comparison (RMSE) for *k*-NN, MDKNN, and CMDKNN

V. Conclusion and Future Work

Contextual information has proven useful for building more accurate recommender systems. In this work, we emphasized the importance of integrating contextual information, rating data, and computational methods for making better recommendations. To undertake the proposed context-aware approach, we also implemented a mobile multimedia recommendation system on a cloud platform to show that our approach can be used to develop a real-world application. To overcome the sparsity and imbalance problems in traditional collaborative filtering methods, we presented a new approach that embeds the contextual information in the computational procedure of the common collaborative filtering method (memory-based method) in a straightforward manner for performance enhancement. A series of experiments were conducted to verify the approaches. The results show that by applying the contextual information, the proposed context-aware *k*-NN approach outperforms conventional methods. In addition to performance, the analyses and evaluations on contexts can provide useful insights to service providers to further develop and improve their services.

The work presented here shows prospects for further research. The experiments conducted were restricted to the available datasets that were relatively small in contrast to datasets without contextual information. We are collecting more datasets to perform extensive evaluations for the proposed approaches. Meanwhile, we are investigating new methods, including adopting the Hadoop MapReduce framework for parallelism [16,17], and accelerating the algorithm to ensure its efficiency for large datasets.

References

[1] F. Ricci, L. Rokach, and B. Shapira, “Introduction to recommender systems handbook,” in: F. Ricci, L. Rokach, B. Shapira, P. Kantor (Eds.), Recommender Systems Handbook, 1-35, 2011.

[2] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Trans. on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734-749, 2005.

[3] Y. Koren and R. Bell, “Advances in collaborative filtering,” in: F. Ricci, L. Rokach, B. Shapira, P. Kantor (Eds.), *Recommender Systems Handbook*, 2011.

[4] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, “Incorporating contextual information in recommender systems using a multidimensional approach,” *ACM Trans. on Information Systems*, vol. 23 no. 1, pp. 103-145, 2005.

[5] J.-Y. Hong, E.-H. Suh and S.-J. Kim, “Context-aware systems: a literature review and classification,” *Expert Systems with Applications*, vol. 36, no. 4, pp. 8509-8522, 2009.

[6] W.-P. Lee, C. Kaoli, and J.-Y. Huang, “A smart TV system with body-gesture control, tag-based rating and context-aware recommendation,” *Knowledge-Based Systems*, vol. 56, pp. 167-178, 2014.

[7] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” *Knowledge-Based Systems*, vol. 46, pp. 109-132, 2013.

[8] J. Bobadilla, A. Hernando, F. Ortega, and A. Gutiérrez, “Collaborative filtering based on significances,” *Information Sciences*, vol. 185, no. 1, pp. 1-17, 2012.

[9] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *IEEE Computer*, vol. 42, no. 8, pp. 30-37, 2009.

[10] S. Rendle, “Factorization machines with libfm,” *ACM Trans. on Intelligent Systems and Technology*, vol. 3, no. 3, Article no 57, 2012.

[11] L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci, “Context relevance assessment and exploitation in mobile recommender systems,” *Personal and Ubiquitous Computing*, vol. 16, no. 5, pp. 507-526, 2012.

[12] L. Liu, F. Lecue, N. Mehandjiev, and L. Xu, “Using context similarity for service recommendation,” in: *Proceedings of IEEE Fourth International Conference on Semantic Computing*, pp. 277-284, 2010.

[13] U. Panniello, A. Tuzhilin, and M. Gorgoglione, “Comparing context-aware recommender systems in terms of accuracy and diversity,” *User Modeling and User-Adapted Interaction*, vol. 24, no. 1-2, pp. 35-65, 2014.

[14] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in: *Proceedings of the Tenth International World Wide Web Conference*, pp.285-295, 2011.

[15] LDOS-CoMoDa dataset, University of Ljubljana, July 2012, http://212.235.187.145/spletnastran/raziskave/um/comoda/comoda.php

[16] J. Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters,” *Communication of the ACM*, vol. 51, no. 1, pp. 107-113, 2008.

[17] K.-H. Lee, Y.-J. Lee, H. Choi, Y. D. Chung, and B. Moon, “Parallel data processing with MapReduce: A survey,” *ACM SIGMOD Record*, vol. 40, no. 4, pp. 11-19, 2011.

ISBN: 978-988-19253-4-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)