Comparing contextual and non-contextual features in ANNs for movie rating prediction

Ghulam Mustafa and Ingo Frommholz

Institute for Research in Applicable Computing
University of Bedfordshire, Luton, UK
ghulam.mustafa4@study.beds.ac.uk, ifrommholz@acm.org

Abstract. Contextual recommendation goes beyond traditional models by incorporating additional information. Context aware recommender systems (CARs) correspond to not only the user's preference profile but also consider the given situation and context. However, the selection and incorporation of optimal contextual features in context aware recommender systems is always challenging. In this paper we evaluate different representations (feature sets) from the given dataset (LDOS-CoMoDa) for contextual recommendations, in particular looking into movie rating prediction as a subproblem of recommendation. We further cross-compare these representations to select useful and relevant features and their combination. We also compare the performance of standard matrix factorization to Artificial Neural Networks (ANNs) in CARs. Our evaluation shows that dynamic, contextual features are dominant compared to non-contextual ones for the given task in the given data set. We also show that ANNs slightly outperform matrix factorization approaches typically used in CARs.

Keywords: Artificial Neural Networks, Feature Selection, Rating Prediction, Context-aware Recommender Systems, Matrix Factorization

1 Introduction

Context aware recommender systems (CARs) incorporate additional contextual information into recommender systems and have emerged as one of the hottest topics in the domain of recommender systems [2, 18]. Traditionally recommender systems focus on recommending the most relevant items to the users or the most appropriate users to the items [1]. While traditional recommendation approaches have performed well in many applications [10], in a number of other applications and contexts, such as location and time based service recommender systems and travel recommendations, it may not be sufficient to consider only users and items [21]. It is also important to incorporate additional contextual information into the recommendation process [10]. A context can be defined as a dynamic set of factors that further describe the state of a user at the moment of user's experience [6]. Nowadays, CARs have become very popular for many applications such as movie, music and mobile recommendations, services for learning, travel and tourism, shopping assistance and multimedia [5, 11]. Most CAR approaches assume the contextual information does not change significantly and remains static, but some dynamic contextualization approaches have been proposed.

In CARs some additional contextual information is available to influence the rating behaviour. A context in this case can be defined as a set $c \in C$, e.g. $c_1 = \{happy, sad, ..\}$, the time at which the movie was watched e.g. $c_2 = \{Morning, Afternoon, \cdots\}$ or the location $c_3 = \{Home, Public, \cdots\}$. In this case multiple contexts $C_1, .., C_m$ are available besides users U and items I, so the function y to estimate the rating R can be expressed as $y : U \times I \times C_1 \times \ldots \times C_m \rightarrow R$.

Instead of providing the user with a recommendation decision, in this work we focus on an important sub-problem — the prediction of the ratings (e.g., 1 to 5 stars) a user might give a certain item. In a later step this prediction can be used for recommendation, for instance by recommending items with a predicted rating of 5 stars. To predict ratings, machine learning algorithms are reported in the literature to develop models and find patterns based on training data. Some of the well-known model based techniques are clustering, associating rules, matrix factorization, restricted Boltzmann machines and others. In context aware recommender systems, the selection of the appropriate context feature remains a persisting challenge [15]. In CARs, using too many context features may result in low accuracy and high dimensionality in the process of recommendation. Recommendation algorithms usually depend on the assumption that the features selected in advance will result in better accuracy [23].

To aim of our study is to gain more insights to aid the feature selection process. We investigate different feature sets (which we call *representations*) and their performance either as single representation or combined. To conduct our studies we use LDOS-CoMoDa, which is the most prominent collection for contextual movie recommendation. This is a very specific collection for the evaluation of CARs as it contains *dynamic contextual features* like location, mood, etc. Previous work has shown that applying dynamic features leads to highly accurate results. However, said previous work has only considered dynamic contextual features, but did not look at other available non-contextual ones (like gender, movie type, etc). Hence one aim of this study is to check the performance of non-contextual features, either alone or combined with contextual ones. Furthermore we show that utilising Artificial Neural Networks (ANNs) instead of matrix factorization, which is prominent in CARs, improves the performance of the rating categorization.

The remainder of this paper is structured as follows. In the next section we discuss some related work in context recommender systems. Subsequently in Section 4, we present our ANN-based approach to predict the ratings and some information about the data set used. We also introduce the contextual and non-contextual representations applied in our work. Furthermore, we present and discuss results of our experiments combining different representations with ANNs in Section 4.2, before we conclude.

2 Related Work

Context aware recommender systems have become very popular in many areas such as movies, music, mobile recommendations, services for learning, travel and tourism, shopping assistance and multimedia [18]. Feature selection in context aware recommender systems is always a challenging task. Since all the features and contexts do not contribute equally to generating valuable recommendations, it is very important to analyse the contextual features to choose the best ones. A number of studies have focused on the selection of contextual features [23, 20]. Different approaches of context aware recommender systems can be categorized by the contextual factors they are considering [16]. Many approaches assume that the contextual information does not change significantly and remains static. This assumption is made in most of the cases, while some recent research has been proposed for dynamic contextualization [9]. Recent work on CARs has focused on developing the models by integrating the contextual information with the user/item relations and models the user and item as well as context interactions [19]. To date, different approaches have been proposed under different categories of CARs including Hybrid Recommender [3], Tensor Factorization and Factorization Machine (FM).
In CARs researchers also suggest the incorporation of meta data such as user or item attributes into the prediction, however meta data normally yields only small improvements over the strong baseline methods that are used for the prediction of ratings [22].

In contextual recommender systems, machine learning algorithms are used to develop models and find patterns based on training data. Most of models are based on using a cluster technique for identification of a user based on test set. Some of the well-known model based techniques are Clustering, Associating rules, Matrix Factorization, Restricted Boltzmann Machines and others [24, 8, 13, 12]. In our approach, we are using ANNs, which haven't been used in detail for contextual recommendations we are dealing with yet.

3 Contextual Recommendations with ANNs

In order to compare the different contextual features, we introduce our ANN-based approach which is composed of a three layers architecture as illustrated in Fig. 1, consisting of an input, hidden and output layer. The input layer is composed of the 6 representations, which are provided as input to ANNs to predict the output y that represents the ratings from 1 to 5. These representations are manually formed based on the nature of the different contextual attributes and explained at the end of this section. The different representations are also combined as input, for example Dynamic representation is combined with the Category and User to find a better match in terms of accuracy. Each of the representations and their combinations are evaluated against the target data, which is the ratings data that comes from the users. The user rating (1-5) is transformed into binary rating as explained in the following section, so that the ANNs can be trained. The optimal set of representations of the context features will be identified and recommended based on the accuracy that comes from the ANNs for each input. A brief description of the different representations with respect to the list of features is given in the Table 1.

In order to train ANNs on the different manually formed representations, we normalize the contextual features. This results in better accuracy for different features and their combinations. The hidden layer, a feed-forward multi-layer perception neural network, is used to map the input into the output binary classes y.

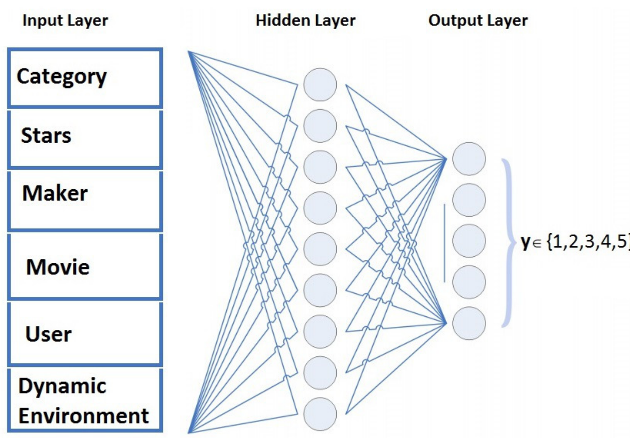

Fig. 1. Architectue of the proposed ANN approach

We pre-process the data to train a model using neural networks. The features available in the dataset are a dynamic set of features and static features. The different features available in the dataset are manually categorized into the 6 representations based on the type of contextual attributes as described in Table 1. Each of the representations is evaluated against the target data (user ratings) in our experiments. Since ANNs are binary classifiers, the target data is converted into binary representations for comparison and evaluation. To turn each of the 5 classes into a binary classification decision, each of the 5 possible ratings is compared to the rest as a yes/no decision (e.g. "Class 1 / not Class 1" to decide if the rating was 1 or not), resulting in 5 classes Class 1 to Class 5. In comparison, matrix factorization considers the ratings provided by the users for the items to map the users and the items in a joint latent feature space [4]. Different representations are also combined to find a better match in terms of accuracy, with less error rate. After evaluating the different representations and their combinations, the optimal feature set combination with highest accuracy will be considered for the recommendation process. Note that an item might be classified into more than one of the above classes (e.g., the ANN may predict 1 star and 4 stars based on the single binary decisions). In this case, our policy is to select the highest rating prediction. The prediction of the ratings for items will also allow to rank itemsSince the other approaches such as probabilistic neural networks are slower than multilayer perceptron networks and require more memory space to store the model, they are not better options at this stage.

Table 1. Representations and Features from LDOS-Comoda dataset

Representation	List of Features
Category	g1, g2, g3 (Genres of the movie)
Stars	a1, a2, a3 (Actors of the movie)
Maker	dir, budget
Movie	movie language, movie year, movie country
User	age, gender, city, country
Dynamic Environment	time, day-type, season, location, weather, season, dominant emotions, end emotion, mood, physical, decision, interaction

Following the representations given in the Table 1, the contextual features are distributed among 6 representations. The *Category* representation consist of movie genres which shows each movie is presented by three genres. The representation of *Stars* consist of the cast of movies, whereas the *Maker* representation contains information about the director of the movies as well as the their budget. The representation *Movie* consist of information about the movie country, movie language and movie year. The representation *User* consists of the static information of users including age, gender, city and country of the user. The *Dynamic Environment* representation contains dynamic variables such as time, day type, season, location, weather, social, dominant emotions, end emotions, mood, physical, decision and interaction. Different representations with the associated contextual information from the LDOS Comoda dataset are shown in Table 2. Once the different representations are identified, a neural network is trained to compare every single representation and combinations thereof with the target data to evaluate the performance and accuracy of the different context features. The optimal set of representations of the context features will be identified and recommended based on the experiments. Further details are provided in the next section.

4 Evaluation

In this section we briefly describe the dataset and the method used for our experiments. First of all we examine the dataset to find which information can be used as potential context from it. Based on the structure of the dataset we define a method how different representation can be formed based on the nature of the contextual features.

4.1 Dataset

The chosen dataset LDOS-CoMoDa[1] consists of 4381 movies which are rated by 121 users. The number of ratings available in this dataset are 2296 and the maximum number of ratings provided by a single user is 220; the minimum number of ratings is 1. The dataset consist of 12 contextual variables in addition to static user information. The basic statistics are given in Table 2.

Table 2. Basic Statistics of LDos-Comoda

Users/Items	121/4381	**Ratings**	2296
Rating scales	1–5	**Context factors**	12
User attributes	4	**Item attributes**	7

In order to evaluate the performance of different representations using binary classification, the true positive rate vs. false positive rate are more helpful than other predictive accuracy matrices [17].

4.2 Results and Discussion

Results In this section the different representations derived from the given contextual variables in the LDOS-Comoda dataset are evaluated, presented and discussed. The work presented in [15] on detecting the relevant context in movie recommender systems provides the relevance and irrelevance of contextual variables. However, we can categorize the contextual variables into the 6 different representations discussed above and cross-compare the representations as well as their combinations to find successful sets of contextual representations. In order to train the neural network on the chosen dataset, the data is preprocessed and normalized in the first stage. The rating data is transformed into a binary form as the neural network performs better using binary classifications. Different features available in the dataset are then normalized with respect to the number of available context features. Then, the ANN is trained using the method described in Section 3 and the samples are divided among the training data, selection data and the validation data. The statistics from the ANN samples division are giving in the Table 3. The number of the samples used by the Neural Network for training purpose is 1608 (70%), 344 for the selection purpose and 344 for the validation. The cross entropy during the training stage of ANN is measured as 1.4213 which shows a small fraction of error occurs during the training stage. The error percentage in the training stage is 3.17% which shows a small fraction of samples are mis-classified during the training stage. Similarly, the selection stage of the Neural Network utilizes 244 (15%) samples with the cross entropy 3.85 and error percentage 2.03. The validation stage also utilizes 344 samples (15%) with cross entropy 3.87 and the percentage of error at 3.19. We have also

[1] http://www.ldos.si/comoda.html

tried the combinations of all representations and observed the higher error rate of 62.20% which shows that it is not an ideal condition to use the features from all representations. A full intersection of the all six representations is not better matched, however, a combination given in Table 4 performed at the rate of 80% which shows the intersection of the Category, User and Dynamic can perform better in the scenario.

Table 3. Sampling from ANN

Entire dataset size	**No. of Samples**	**Cross-Entropy Measure**	**% Error**
Training dataset size	1608	1.4213	3.17
Selection dataset size	344	3.85	2.03
Validation dataset size	344	3.87	3.19

Using the method described in Section 3, we cross-compared the different representations with the target data to find the relevant representation which is a set of features. Features are cross-compared one by one by training the neural network which learns over 2296 samples (70% for training, 15% for testing and 15% for validation). The results from the experiments shows the performance of the Dynamic Environments representation performs better than the other representations Maker, Category, User, Movie and Stars. The performance of the Dynamic Environment representation remains above the threshold line when the binary classes are used to obtain the performance. The representations other than Dynamic Environment struggle with the errors and shown inferior performance, so the set of the features given as part of Dynamic Environment are a good set of features that can be used potentially for generating the recommendations. So the recommended stand alone representation is the set of features given in Dynamic Environment.

As we can see in the Table 4, the context features available in the Dynamic Environment representation performed better while the other representations struggle with respect to the performance and errors. So the Dynamic Environment representation is picked as the single optimal set of features. We also tried combinations of Dynamic Environment with other representations such as Category, Makers, Stars and User Statics to study combinations of representations. The comparison of combinations given in Figure 2 shows that the performance of the Dynamic Environment is not improved when combining this representation with others; the representations do not seem to complement each other. This means Dynamic Environment is indeed the dominant representation in the LDOS-CoMoDa collection.

We use the results reported in [14], using matrix factorization (MF), as baseline to compare the performance of our ANN approach since it utilizes the contextual attributes which are part of the Dynamic Environment in our method. The results comparison in Fig. 3 between the contextual attributes in the baseline method on the one hand and the ANNs on the other hand shows that contextual attributes performed better with ANN.

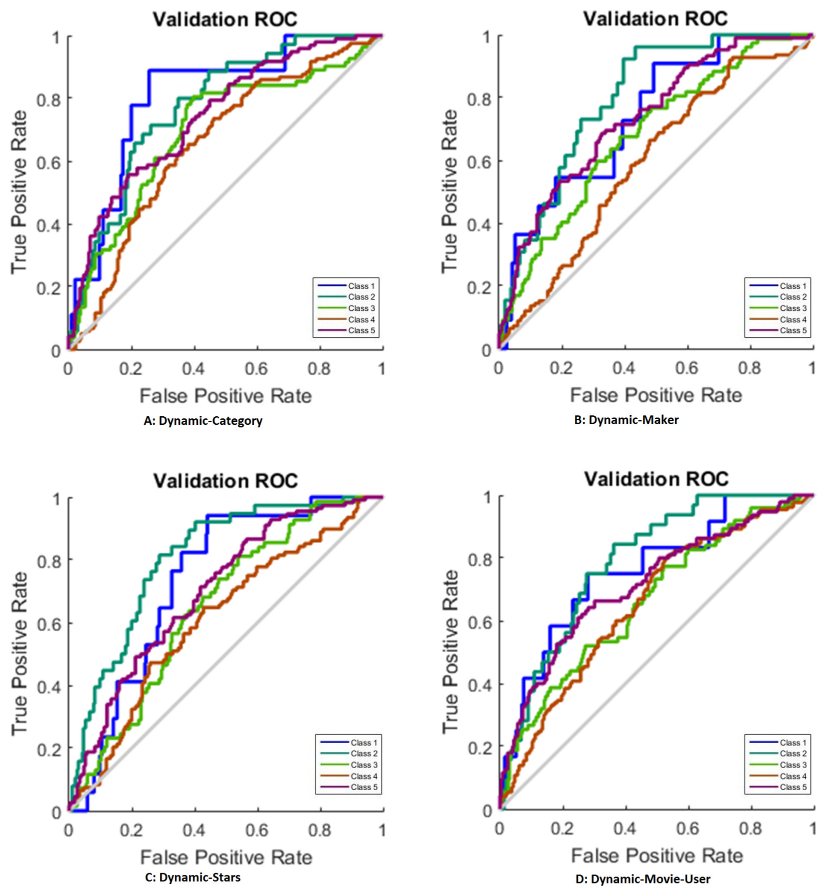

Fig. 2. True Positive Rate vs. False Positive Rate for combination of Dynamic Environment representation with other representations

Table 4. Performance of different features from ANN

Representation	Performance (Accuracy)	MF
Dynamic	**97.12**	**96.9**
Category	80.68	Not Reported
Makers	65.8	Not Reported
Where	66.58	Not Reported
User	64.9	Not Reported
Category + User + Dynamic	80.81	Not Reported

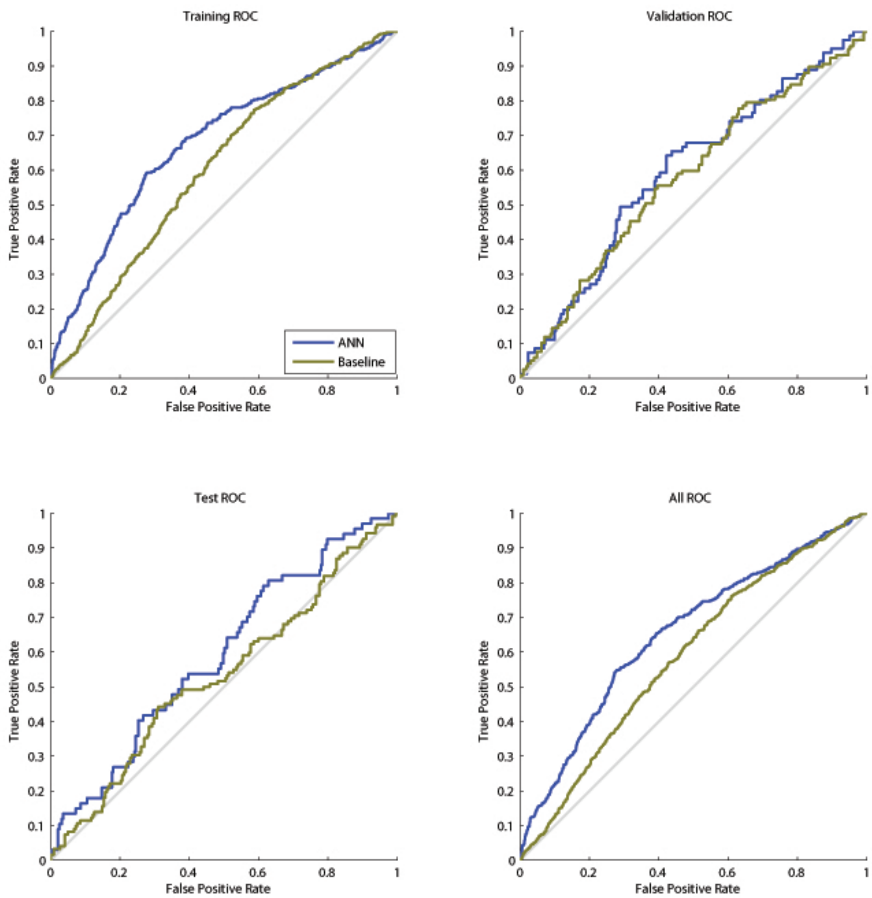

Fig. 3. Comparison of Contextual variables from Dynamic Environments with Baseline

The performance of the ANN is also evaluated by computing the Cross-Entropy which helps to evaluate the performance of three different stages of ANN (Train, Validation and Test) against the best performance. The results presented in the Fig. 4 shows that the performance of ANN remains better for all three stages when the ANN is trained for 22 epochs. In ANNs, an epoch is used to present the set of training vectors to the network for the calculation of new weights. The best performance is achieved in validation, as can be seen in the circle and gradient line in the figure, which means the performance is deemed acceptable according to the literature.

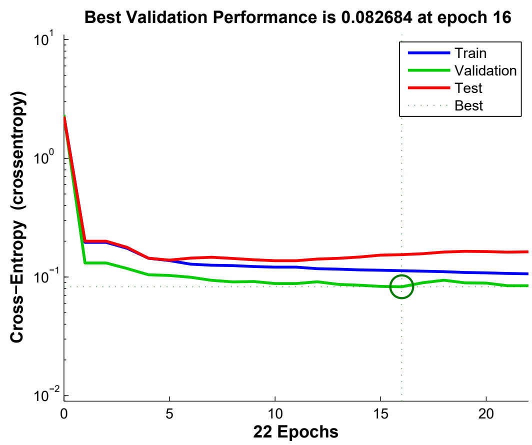

Fig. 4. Performance of ANN during Train, Validation and Test stages

Discussion The results have shown that contextual Dynamic Environment features by far outperform the static non-contextual features in the chosen LDOS-CoMoDa collection when it comes to rating prediction. The results also show that applying ANNs instead of matrix factorization improves the rating prediction accuracy even further when using the Dynamic Environment features. It confirms the important role of contextual features for CARs and the rather inferior role non-contextual features play, at least in the given data set. ANNs are indeed a very effective method for rating prediction, which is crucial for context-based recommendation.

5 Conclusion and Future Work

In this paper we introduced ANNs for rating prediction in contextual recommendations. We presented how to form different representations from a chosen dataset LDOS-CoMoDa. Different representations are cross-compared and concluded that the Dynamic Environment context features performed best when applied alone, also outperforming the chosen matrix factorization baseline method. We further cross-compared combinations of the Dynamic Environment with other representations and observed that they do not perform well and are even not able to further complement the dynamic features, at least not with the combinations of the different representations.

The LDOS-CoMoDa dataset is an interesting data set when it comes to providing a rich set of dynamic contextual features. The dominance of such features for the given rating prediction task is remarkable. In the future we will look into similar data sets and investigate the role of dynamic contextual features compared to static, non-contextual ones. In this respect, we will also check if there is still a way to combine non-contextual features with dynamic, contextual ones, given that other data sets do not possess a dominant feature set like we find with LDOS-CoMoDa. One potential idea is borrowed from the principle of polyrepresentation [7], which is also a reason why we called feature sets *representations* in this work. If documents are recommended by different classifiers using different representations (feature sets), we would expect that the set of documents recommended by all classifiers exhibits a high precision. This would also give rise to a more interactive approach to recommendation, for instance by presenting to the user those recommendations first that are confirmed by different representations and let the user decide which set of recommendations to visit next (for instance those that match the current mood vs. those that match other features like age, location or genre). Whether we can actually observe something 'polyrepresentation-like' in machine learning based recommendation is subject to further investigation.

References

1. Nana Yaw Asabere. Towards a viewpoint of context-aware recommender systems (CARS) and services. *Inernational Journal of Computer Science and Telecommunications*, 4(1), 2013.
2. Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 2015.
3. Claudio Biancalana, Fabio Gasparetti, Alessandro Micarelli, Alfonso Miola, and Giuseppe Sansonetti. Context-aware movie recommendation based on signal processing and machine learning. *Proceedings of the 2nd Challenge on Context-Aware Movie Recommendation - CAMRa '11*, pages 5–10, 2011.
4. Maunendra Sankar Desarkar and Sudeshna Sarkar. Rating prediction using preference relations based matrix factorization. In *UMAP Workshops*, 2012.
5. Negar Hariri and Robin Burke. Context-aware music recommendation based on latent topic sequential patterns. In *Proceedings RecSys 2012*, pages 131–138, 2012.
6. Balázs Hidasi and Domonkos Tikk. Fast ALS-based tensor factorization for context-aware recommendation from implicit feedback. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 67–82, 2012.

7. Peter Ingwersen and Kalvero Järvelin. *The turn: integration of information seeking and retrieval in context*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
8. Salil Kanetkar, Akshay Nayak, Sridhar Swamy, and Gresha Bhatia. Web-based personalized hybrid book recommendation system. In *International Conference on Advances in Engineering and Technology Research*, pages 1–5, 2014.
9. Pavlos Kefalas, Panagiotis Symeonidis, and Yannis Manolopoulos. New perspectives for recommendations in location-based social networks. *Proceedings MEDES 2013*, pages 1–8, 2013.
10. Fabrício D. A. Lemos, Rafael A. F. Carmo, Windson Viana, and Rossana M. C. Andrade. Towards a context-aware photo recommender system. *CEUR Workshop Proceedings*, 889, 2012.
11. Asher Levi, Osnat Ossi Mokryn, Christophe Diot, and Nina Taft. Finding a needle in a haystack of reviews: Cold start context-based hotel recommender system. In *Proceedings RecSys 2012*, pages 115–122, 2012.
12. Jiankou Li and Wei Zhang. Conditional Restricted Boltzmann Machines for cold start recommendations. *arXiv preprint arXiv:1408.0096*, 2014.
13. Ziqi Liu, Yu-Xiang Wang, and Alexander Smola. Fast differentially private matrix factorization. In *Proceedings RecSys 2015*, pages 171–178, 2015.
14. Ante Odić, Marko Tkalčič, Jurij F. Tasič, and Andrej Košir. Relevant context in a movie recommender system: Users' opinion vs. statistical detection. *CEUR Workshop Proceedings*, 889, 2012.
15. Ante Odić, Marko Tkalčič, Jurij F. Tasič, and Andrej Košir. Predicting and detecting the relevant contextual information in a movie-recommender system. *Interacting with Computers*, 25(1):74–90, 2013.
16. Simon Renaud-Deputter, Tengke Xiong, and Shengrui Wang. Combining collaborative filtering and clustering for implicit recommender system. In *27th International Conference on Advanced Information Networking and Applications (AINA 2013)*, pages 748–755. IEEE, March 2013.
17. Antonio J. Serrano, Martin José D. Soria, Emilio, Rafael Magdalena, and Juan Gómez. Feature selection using ROC curves on classification problems. In *International Joint Conference on Neural Networks (IJCNN 2010)*, pages 1–6, 2010.
18. Gediminas Adomavicius Tuzhilin and Alexander. Advances in Collaborative Filtering. In *Recommender Systems Handbook*, pages 217–253. 2011.
19. Shu Lin Wang and Chun Yi Wu. Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system. *Expert Systems with Applications*, 38(9):10831–10838, 2011.
20. Hao Wu, Kun Yue, Xiaoxin Liu, Yijian Pei, and Bo Li. Context-aware recommendation via graph-based contextual modeling and postfiltering. *International Journal of Distributed Sensor Networks*, 2015:7–9, 2015.
21. Liu Xin and Karl Aberer. Personalized Point-of-Interest Recommendation by Mining Users Preference Transition. *Proceedings CIKM 2013*, pages 733–738, 2013.
22. Mao Ye, Peifeng Yin, and Wang-Chien Lee. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 458–461, 2010.
23. Saloua Zammali, Khedija Arour, and Amel Bouzeghoub. A Context Features Selecting and Weighting Methods for Context-Aware Recommendation. *2015 IEEE 39th Annual Computer Software and Applications Conference*, 5:575–584, 2015.
24. Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit Factor Models for Explainable Recommendation based on Phrase-level Sentiment Analysis Categories and Subject Descriptors. In *Proceedings SIGIR 2014*, pages 83–92, 2014.