A Content-based Movie Recommender System based on Temporal User Preferences

Bagher Rahimpour Cami
Faculty of Computer Engineering & IT
Shahrood University of Technology
Shahrood, Iran
rc_bagher@yahoo.com

Hamid Hassanpour
Faculty of Computer Engineering & IT
Shahrood University of Technology
Shahrood, Iran
h.hassanpour@shahroodut.ac.ir

Hoda Mashayekhi
Faculty of Computer Engineering & IT
Shahrood University of Technology
Shahrood, Iran
hmashayekhi@shahroodut.ac.ir

***Abstract*—Recommender systems have emerged as the essential part of many e-commerce web sites. These systems provide personalized services to assist users in finding favorite items among the huge number of available media on the World Wide Web. Identifying temporal preferences of individuals is one of the major challenges of recommender systems to provide personalization for users. In this paper, a content-based movie recommender system is proposed that captures the temporal user preferences in user modeling and predicts the preferred movies. The proposed method provides a user-centered framework that incorporates the content attributes of rated movies (for each user) into a Dirichlet Process Mixture Model to infer user preferences and provide a proper recommendation list. We implement the proposed method and use the MovieLens dataset to perform experiments. The evaluation results show that the performance of proposed recommendation method outperforms the existing movie recommender systems.**

Keywords—movie recommender system; content-based movie recommendation; temporal user preferences

I. Introduction

Recommender systems encapsulate powerful filtering methodologies and techniques, and construct software tools that are able to recommend favorite resources. These systems employ the past transactions of users and extract users' interests to predict users' preferences. Past transactions constitute the user profile, Which may contain user demographics data (e.g. age, gender), feedback data (e.g. rating, click), context data (e.g. time and location), and resource features (e.g. keyword vector) [1]. A recommender system helps users to find interested resources. Moreover, it provides a framework for e-commerce web sites to represent their products to the target users.

There are several methods that are used to construct recommender systems including: collaborative filtering, content-based, context-based, knowledge-based, and hybrid recommender systems [2]. In the collaborative filtering approaches the user profile consists of users' feedback (e.g. ratings) and the neighborhood measure is used to provide recommendations [3, 4]. This approach recommends items that are similar to the ones that the user or neighbors have preferred in the past. The content-based recommender systems usually use the content features of resources (e.g. textual description) to construct the user profile. These approaches use a similarity measure [5, 6] to recommend the items that are similar to the ones the user has previously selected. Context-aware recommender systems, as the third category, employ contextual information such as time, location, and social data to make recommendations [7]. With the user requirements, or resource specifications gathered in user profile, the knowledge-based approaches use inferencing/ case-based reasoning to find resources that fulfill the user needs [8, 9]. The hybrid approaches combine the above mentioned approaches and use a rich user profile that consists of several components such as ratings, temporal/spatial, social info [9, 10].

The success of a recommender system to provide a proper recommendation highly depends on the user interests and using them to predict the user preferences. Each user interest shows the group of similar resources that a user selected in the past. User preferences indicate the probability that the user may select resources from a specific interest. The similarity based and latent factors are two essential methods that are used to process users' profile and extract user interests. The latent factors map the users and items with latent correlation into a common space. Each latent factor represents a user interest. Item-based and user-based similarity are another methods that are used for grouping items and users [11]. For example, in the item-based approach, each group shows the correlated items that may be selected together.

The temporary feature of user profile is among the main challenges of capturing user interests. This feature poses the user interests, and may be varied over time. At two state of the art studies on temporal recommender systems[11,13], the temporary feature is captured using time bins [12, 13]. Each bin represents a group of movies that are rated at the same time slice.

Overall, the existing methods mentioned above focus on user interests. They attempt to identify user interests and take into account some context information to capture it in a temporal manner [9, 14]. In this paper, we analyzed the rating behaviors of users collected in MovieLens dataset[1], and found that users' rating behaviors in social media systems are influenced by not only intrinsic interest but also the temporal preferences. Thus the need arises for inferring user preferences

[1] https://grouplens.org/datasets/movielens/

as the main factor in predicting future favor resources of the user. We focus on the domain of movie recommendations and propose a novel content-based technique that utilizes a temporal user model. In this paper, we gather the movies information from IMDB[2] and employ them into a Bayesian nonparametric framework that presented in [15, 16] to capture the temporal preferences of a user. This is a novel method in movie domain and we extend it to apply for the movie recommendation. In contrast to the temporal recommendation method presented in [12, 13], the proposed method provides a temporal movie recommender that is able to decide about the new movies. We implemented the proposed method and performed the experiments with MovieLens dataset. The calculation of the evaluation metrics in terms of F1-measure and NDCG [4, 15], shows the proposed method outperforms some of the state of the art methods.

In the rest of the paper, we provide related work in Section II. The proposed method is presented in Section III. We give an overview of our dataset, the experimental results, and comparisons in Section IV. In Section V, we present our conclusions and future work.

II. Related Works

Recommender systems are employed in different application domains such as on-line news services (Google news), products (Amazon), and media (Netflix) [2, 11, 17]. E-commerce websites that supply various products are one of the well-known hosts of recommender systems. One of the most widely used application domain is movie recommendation [18]. Movie recommender systems help customers to access preferred movies from a huge on-line multimedia library automatically. Also, a large number of recommender systems use the movie-based datasets to confirm their results [13, 19, 20]. We categorized the approaches in movie recommender systems into three groups as follows:

- **Latent factor**, in which the user profile (rating matrix) is analyzed and some latent factors are extracted to model the user behavior in selecting movies. Authors in [21] applied the SVD algorithm to rating matrix and extracted latent factors. Each latent factor represents vectors of hidden attributes for users and items that the importance of each attribute is calculated with rating matrix. To predict the rating of item i respected to a user u, the correlation of $vector_i$ and $vector_u$ is calculated. The non-negative matrix factorization (NMF) technique is used to improve the latent factor extraction [22, 23]. NMF focuses on the analysis of data matrices whose elements are non-negative, and can be used to obtain a new part-based representation in some lower dimensional space. Resulting factorization from NMF often enables better semantic interpretation, and thus can be used to derive more accurate clustering [24]. This approach is improved to provide temporal recommendation in [12] by incorporating the time of each activity (rating). Also, authors in [13] presented a temporal recommender that uses a Bayesian framework to infer latent factors.
- **Similarity based**, in which groups of similar items/users are discovered. Neighborhood and K-means are two popular techniques used in collaborative movie recommender systems [11, 18]. Authors in [25] used artificial immune system (AIS) to construct an artificial immune network and employ it to find neighborhood of users. Also, the correlations of movie genre is used to improve the item-based similarity in [26].
- **Hybrid approach**, in which a combination of approaches is used to improve the performance of recommender systems. The contextual information (e.g. location, trust, semantic-ontology, social network, and time) and content information (e.g. tag, genre, description, and visual features) are two main tools that are used in this approach to provide a better recommendations. [10].

In this paper, we focus on identifying user preferences from user profile. We gather the movie information from IMDb site and incorporate them into a Bayesian framework and create a temporal preferences of users.

III. Proposed method

We propose a novel method to overcome two main issues in movie recommender systems. In following, we describe the modeling and construction phases of proposed content-based movie recommender system.

A. Modeling The Temporal Preferences of User

In this paper, we extend the temporal preferences model introduced in [15, 16] to construct a content-based movie recommender system. In this model the user profile consists of user activities as $\langle userId,\ activity_1, ...,\ activity_n \rangle$, where each $activity_i$ indicates the content and access time of selected items denoted as $\langle itemId,\ itemDesc,\ accessDate \rangle$. This model is user-centered and employs profile of each user to create user model for individuals. In movie domain, each rating record of rating matrix ($\langle movieId,\ movieDesc,\ rate,\ accessDate \rangle$) is corresponded to an activity.

The temporal preferences model is based on Bayesian non-parametric framework and has three main component: $interest\ extraction$, $inferring\ of\ preferences$, and $prediction$.

Interests extraction, where analysis the user profile to discover user interests. This model employs the user profile into Distance Dependent Chinese Restaurant Process (DDCRP) [27] and performs clustering. DDCRP is based on Bayesian non-parametric thus, the clusters can grow whenever new data is observed. each cluster is mapped to an $interest$ and indicates a group of similar items. To perform clustering for each observation a_i the following equation is used:

$$p\left(t_{a_i} = k \mid \mathbf{t}, a_{-i}, \alpha\right) = \begin{cases} Sim\,(.) * \mathcal{L}(.) & k \in \mathbf{t} \\ \alpha & k > |\mathbf{t}| \end{cases} \tag{1}$$

[2]www.imdb.com

where $Sim(.)$ calculates the similarity of a_i related to other items, $\mathcal{L}(.)$ calculates the likelihood of a_i given other parameters, and α is the probability of new cluster creation which is initiated with the average similarity of all items in user profile. The values are scaled in range [0,1]. Also, **t** indicates the cluster assignments of all items and t_{a_i} represents the cluster assignment of i^{th} observation.

Preferences inferring. By extracting user interests, we consider the age and amount of user activities to calculate the preference vector. Each element of the preference vector shows the probability that a user prefers to select an interest to perform new activities. Equation (2) is used to construct preference vector for a user.

$$Pref\,(cluster_i) \sim \sum_{j=1}^{|cluster_i|} Age(activity_j) \qquad (2)$$

Prediction, where the probability that a new item a^* will be selected by a user. For prediction process, first, we calculate the probability that new item a^* is assigned to cluster c_i using similarity function $Sim(a^*,\ .)$. Then, the preference vector is used to determine the priority of cluster c_i. Therefore, to calculate selection probability of a new item a^*, (3) is used.

$$p\,(a^* \mid .) \sim \sum_{c_i \in Cluster} (Sim(a^*, c_i, .) * Pref\,(c_i)) \qquad (3)$$

To construct the proposed movie recommender system, we use $interests\ extraction$ and $preferences\ inferring$ for user modeling and $prediction$ to provide recommendation list. To create recommendation list, we calculate (3) for all new items and sort them.

B. Content-based Movie Recommender System

To construct the content-based movie recommender system, we consider the user profile consisting of movie information (such as $story-line$ and $genre$), which user rated in a specified time. The movie information is gathered from IMDb. We incorporate the profile of individuals into the $interests\ extraction$ module and discover user interests. Each interest indicate a group of similar movies that are selected by the user in the past.

With the groups of movies, $preference\ inferring$ module uses the rating time and number of movies within each group to calculate the priority (probability) of each group. This module creates a probability vector as $\langle\theta_1, ...,\ \theta_k\rangle$, where θ_j indicates the probability that user performs the selection from j^{th} group.

After constructing user model, the $prediction$ module is used to provide a recommendation list. This module calculates the likelihood of each new item, if it was assigned to each group to determine the selection probability of that item. The block diagram of the proposed method is depicted in Fig 1.

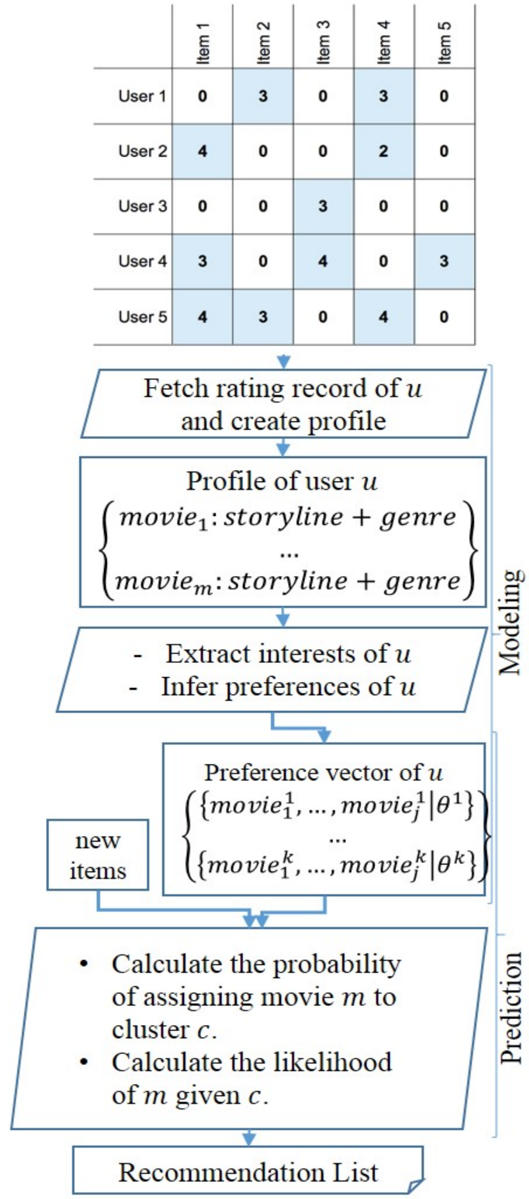

Fig. 1. block diagram of the proposed content-based movie recommender system. u indicates anonymous user.

IV. Experimental Results

We implemented the proposed method for constructing a content-based movie recommender system. To perform experiments, we use the MovieLens dataset. It consists of 10 million ratings, 10000 movies, and 72000 users. We preprocessed the data and selected a subset of users with at least 100 ratings. To provide the content information for movies, we gathered the movie story-line and genre from IMDb for each movie item within dataset. Then, the movies information were processed using text processing tools (NLTK) [28] and the textual feature vector for each movie. The profile of each user is constructed with textual feature vector of rated movies. Therefore, the user profile consists of the content of rated movies per each user. For comparison, we used the movie recommendation system (Time-SVD++) which introduced in [12] . This method used the latent factor approach to construct temporal movie recommendation.

To evaluate the results of proposed method, the measures Precision@N (P@N), Recall@N (R@N), F1-measure@N (F1@N) were used [29]. We implemented the proposed method with Python and recorded results for each user. Then, the evaluation metrics were averaged over all users for different lengths (top@N) of recommendation list such as {5, 10, 15, 20}. To perform the evaluation, we selected 30% of user profile as test data. To calculate metrics, we selected the rating records with rate value greater than 4 as the relevant data. Then we provided recommendation list with length {5, 10, 15, 20} and calculate metrics.

The evaluation results are depicted in Table I. Each row of this table represents the values of evaluation metrics for a specific top@N. Also, the graphical view of results is depicted in Fig. 2. The solid line of this figure shows the results of the proposed method. The precision, recall, and f1-measure are depicted in sub-figures of Fig. 2, respectively. The results show that, the proposed method improved the accuracy of the movie recommender system. Therefore, results verify that the temporal preferences modeling and user-centered approach are useful in the movie recommender system. Moreover, against the time-SVD++ that has over than four parameters, the proposed method has only one hyper-parameter (parameter α in Eq. 1) that is initiated with the average similarity of the user profile items. On the other hand, the proposed method can also provide recommendation for new movies. In the time-SVD++, the recommended movies should be rated by other users (there are not new movie), whereas the recommended movies of proposed method may have been never seen previously.

TABLE I: Performance comparison of the Proposed and time-SVD++ Methods using various Evaluation Metrics.

top@N	Method	precision	recall	F1
5	time-SVD++	0.76	0.16	0.30
	proposed RS	0.76	0.18	**0.29**
10	time-SVD++	0.71	0.36	0.46
	proposed RS	0.77	0.37	**0.49**
15	time-SVD++	0.68	0.51	0.56
	proposed RS	0.73	0.46	**0.56**
20	time-SVD++	0.66	0.65	0.63
	proposed RS	0.75	0.57	**0.64**

V. Conclusions

In this paper, we developed a content-based movie recommender system. The proposed method used a temporal preference model of the user to recommend favor movies to the users. For user modeling, the proposed method after extracting user interests from user profile, the priority of each interest is inferred as the user preference. We evaluated the proposed method with MovieLens dataset. Experimental results show that the proposed method improved the accuracy of movie recommendation. Also, it is able to recommend new movies based on their content.

For future works, item-based collaborative approach can be used to improve the diversity and serendipity measure of recommendation.

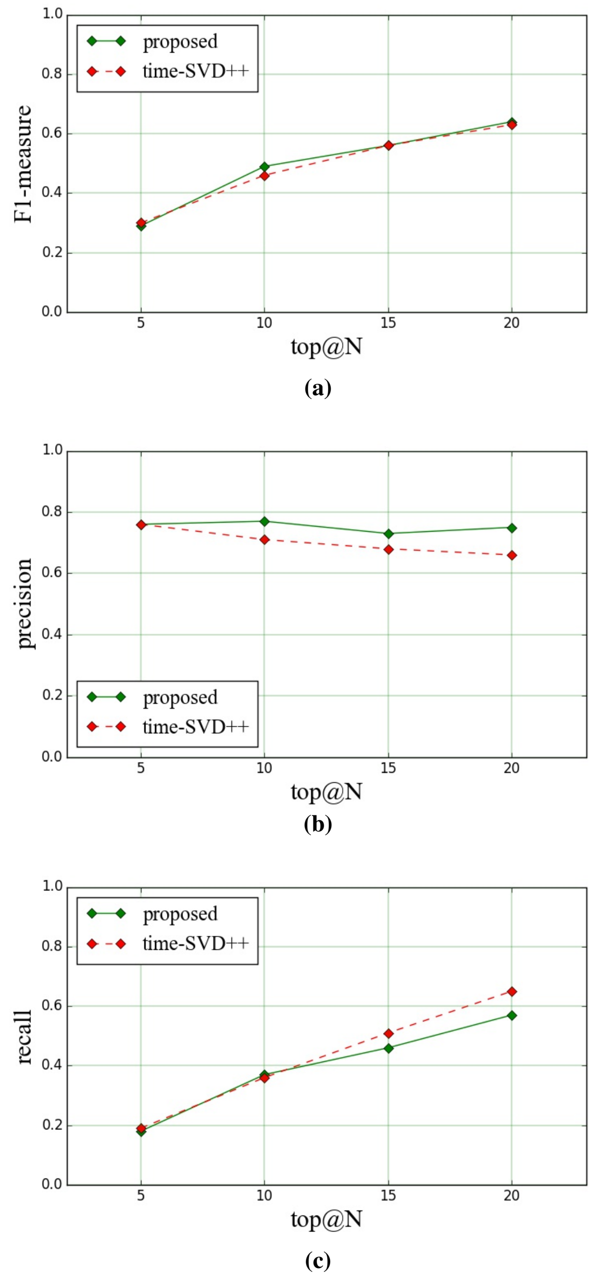

Fig. 2. Comparing the performance of the proposed RS with time-SVD++.

References

[1] Susan Gauch, Mirco Speretta, Aravind Chandramouli, and Alessandro Micarelli. User profiles for personalized information access. *The adaptive web*, pages 54–89, 2007.

[2] Charu C Aggarwal. An introduction to recommender systems. In *Recommender Systems*, pages 1–28. Springer, 2016.

[3] Nan Zheng and Qiudan Li. A recommender system based on tag and time information for social tagging

systems. *Expert Systems with Applications*, 38(4):4575–4587, 2011.

[4] Hao Wu, Yijian Pei, Bo Li, Zongzhan Kang, Xiaoxin Liu, and Hao Li. Item recommendation in collaborative tagging systems via heuristic data fusion. *Knowledge-Based Systems*, 75:124–140, 2015.

[5] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[6] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.

[7] Cosimo Palmisano, Alexander Tuzhilin, and Michele Gorgoglione. Using context to improve predictive modeling of customers in personalization applications. *IEEE transactions on knowledge and data engineering*, 20(11):1535–1549, 2008.

[8] Walter Carrer-Neto, María Luisa Hernández-Alcaraz, Rafael Valencia-García, and Francisco García-Sánchez. Social knowledge-based recommender system. application to the movies domain. *Expert Systems with applications*, 39(12):10990–11000, 2012.

[9] Luis Omar Colombo-Mendoza, Rafael Valencia-García, Alejandro Rodríguez-González, Giner Alor-Hernández, and José Javier Samper-Zapater. Recommetz: A context-aware knowledge-based mobile recommender system for movie showtimes. *Expert Systems with Applications*, 42(3):1202–1222, 2015.

[10] Shouxian Wei, Xiaolin Zheng, Deren Chen, and Chaochao Chen. A hybrid approach for movie recommendation via tags and ratings. *Electronic Commerce Research and Applications*, 18:83–94, 2016.

[11] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

[12] Yehuda Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

[13] Hongzhi Yin, Bin Cui, Ling Chen, Zhiting Hu, and Xiaofang Zhou. Dynamic user modeling in social media systems. *ACM Transactions on Information Systems (TOIS)*, 33(3):10, 2015.

[14] Jennifer Golbeck and James Hendler. Filmtrust: Movie recommendations using trust in web-based social networks. In *Proceedings of the IEEE Consumer communications and networking conference*, volume 96, pages 282–286, 2006.

[15] Bagher Rahimpour Cami, Hamid Hassanpour, and Hoda Mashayekhi. User trends modeling for a content-based recommender system. *Expert Systems with Applications*, 87(30):209–219, 2017.

[16] Bagher Rahimpour Cami, Hamid Hassanpour, and Hoda Mashayekhi. Inferring user preference using distance dependent chinese restaurant process and weighted distribution for a content based recommender system. proceedings of the 19th artificial intelligence and signal processing conference. 2017. (accepted).

[17] Mohamed Sarwat, Justin J Levandoski, Ahmed Eldawy, and Mohamed F Mokbel. Lars*: An efficient and scalable location-aware recommender system. *IEEE Transactions on Knowledge and Data Engineering*, 26(6):1384–1399, 2014.

[18] Zan Wang, Xue Yu, Nan Feng, and Zhenhua Wang. An improved collaborative movie recommendation system using computational intelligence. *Journal of Visual Languages & Computing*, 25(6):667–675, 2014.

[19] Jason J Jung. Attribute selection-based recommendation framework for short-head user group: An empirical study by movielens and imdb. *Expert Systems with Applications*, 39(4):4049–4054, 2012.

[20] Jesús Bobadilla, Francisco Serradilla, and Jesus Bernal. A new collaborative filtering metric that improves the behavior of recommender systems. *Knowledge-Based Systems*, 23(6):520–528, 2010.

[21] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[22] Antonio Hernando, Jesús Bobadilla, and Fernando Ortega. A non negative matrix factorization for collaborative filtering recommender systems based on a bayesian probabilistic model. *Knowledge-Based Systems*, 97:188–202, 2016.

[23] Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*, 10(2):1273–1284, 2014.

[24] Mei Lu, Li Zhang, Xiang-Jun Zhao, and Fan-Zhang Li. Constrained neighborhood preserving concept factorization for data representation. *Knowledge-Based Systems*, 102:127–139, 2016.

[25] Meng-Hui Chen, Chin-Hung Teng, and Pei-Chann Chang. Applying artificial immune systems to collaborative filtering for movie recommendation. *Advanced Engineering Informatics*, 29(4):830–839, 2015.

[26] Sang-Min Choi, Sang-Ki Ko, and Yo-Sub Han. A movie recommendation algorithm based on genre correlations. *Expert Systems with Applications*, 39(9):8079–8085, 2012.

[27] David M Blei and Peter I Frazier. Distance dependent chinese restaurant processes. *Journal of Machine Learning Research*, 12(Aug):2461–2488, 2011.

[28] Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics, 2006.

[29] Iman Avazpour, Teerat Pitakrat, Lars Grunske, and John Grundy. Dimensions and metrics for evaluating recommendation systems. In *Recommendation systems in software engineering*, pages 245–273. Springer, 2014.