International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:11, No:9, 2017

Inferring User Preference Using Distance Dependent Chinese Restaurant Process and Weighted Distribution for a Content Based Recommender System

Bagher Rahimpour Cami, Hamid Hassanpour, Hoda Mashayekhi

***Abstract*—**Nowadays websites provide a vast number of resources for users. Recommender systems have been developed as an essential element of these websites to provide a personalized environment for users. They help users to retrieve interested resources from large sets of available resources. Due to the dynamic feature of user preference, constructing an appropriate model to estimate the user preference is the major task of recommender systems. *Profile matching* and *latent factors* are two main approaches to identify user preference. In this paper, we employed the latent factor and profile matching to cluster the user profile and identify user preference, respectively. The method uses the Distance Dependent Chines Restaurant Process as a Bayesian nonparametric framework to extract the latent factors from the user profile. These latent factors are mapped to user interests and a weighted distribution is used to identify user preferences. We evaluate the proposed method using a real-world data-set that contains news tweets of a news agency (BBC). The experimental results and comparisons show the superior recommendation accuracy of the proposed approach related to existing methods, and its ability to effectively evolve over time.

***Keywords*—**Content-based recommender systems, dynamic user modeling, extracting user interests, predicting user preference.

I. Introduction

IN the current WWW, where the quantity of resources are huge (information overloading), a recommender system is a very useful tool to support people in making decisions. Recommender systems have emerged as an essential part of the online websites to tackle information overloading problem. These systems collect user(s) transactions as the user(s) profile and process them to provide a personalized environment. The personalization assists users in retrieving preferred items. Recommender systems are employed in different domains such as online marketing, online news, and social networks. Amazon (amazon.com) and Google News (news.google.com) are two examples of the well-known websites that use recommender systems to suggest products and news articles in a personalized environment [1], [2].

Hamid Hassanpour is with the Faculty of Computer Engineering and Information Technology, Shahrood University of Technology, Shahrood, Iran (corresponding author, phone: +98(23)32300251; fax: +98(23) 32300251; e-mail: h.hassanpour@shahroodut.ac.ir).

Bagher Rahimpour Cami and Hoda Mashayekhi are with the Faculty of Computer Engineering and Information Technology, Shahrood University of Technology, Shahrood, Iran (e-mail: rc_bagher@yahoo.com, hmashayekhi@shahroodut.ac.ir).

One of the most important challenges in recommender systems is their ability to identify the user’s preferences and used them to generate personalized recommendations. There are several methods that process the past transactions of users to understand their interests [3], [4]. These methods differ in their input data and the applied recommendation algorithm. The input data, commonly known as the user profile, includes user related information (demographic), item specification (e.g. item content), context (e.g. time and location) and explicit/implicit feedback (rating) [5]. The user profile is application dependent and may consist of any subset of the above mentioned information.

Collaborative filtering, content-based filtering, knowledge-based filtering and hybrid methods are among the basic approaches of recommender systems [3]-[8]. In *collaborative* filtering, the user profile consists of demographic and feedback information and recommending new items is based on user-user, user-item, or item-item similarities [9]. This method provides a framework for finding groups of similar users to employ their feedbacks about the selected items for recommendation. It ignores the items representation. In *content-based* filtering, the user profile consists of the content of selected items (items’ features). The user-item profile matching is further adopted as the recommendation algorithm [10]. In *knowledge-based* filtering, a knowledge-base is constructed from the users' requirements and items' constraints, which enables recommendation by inference procedures [11]. If multiple input data such as feedback, item features, and context parameters are available, the *hybrid* approach, which combines several recommendation algorithms, can be used to improve the performance of the recommender system [12].

In this research, we focus on the content approach and attempt to construct a proper user model. The existing approaches of user modeling can be categorized into three non-disjoint groups including profile matching, long-short-term interests and latent factors.

Profile matching: This approach constructs personalized profiles to model the users. There are different methods to construct the content and structure of user profiles [13]. Some researchers incorporate the content of items to construct user profile [14], [15]. The authors in [14] incorporated the user profile into a context-tree model and provided a framework to estimate next items. In addition to content of items, some other exclusive characteristics such as named entities and similar access patterns may also be used to construct a richer user profile [16]. Also, the click behavior or tracking logs are employed to construct the user profile and a Bayesian framework is used for recommendation [2], [17].

Long-short-term interests: In this approach, which is introduced by Pazzani et al. [18], [19], the long-term interests are used to specify the general preferences of user and the short-term interests are used to determine the hot events. A classifier such as naive Bayes is applied on the long-term interests to identify candidate items and KNN is used to select appropriate items for recommendation. In some studies hierarchical clustering is employed to derive long-term interests [20]-[22]. The authors in [21] construct a user-item affinity graph from the short-term and then the Absorbing Random Walk is employed to provide a recommendation list. In another study [23], the long-short-term interests are incorporated in a graph-based recommender system and a Random Walk method is used for recommendation.

Latent factor: Some approaches believe that the user interests are influenced by a set of latent factors that are specific to the domain. For example, in a movie recommendation, the comedy versus drama, or amount of action in the movie, may affect the interest of users in different movies. The Matrix Factorization (MF) and Hidden Markov Model (HMM) are two famous methods to infer latent factors. MF infers the latent factors from a rating matrix [24] and predicts the rate of unknown items. Koren introduced a temporal matrix factorization that incorporates the access date of each items [25]. MF is also extended by incorporating content [26], [27] and context [28] information. The HMM is used to capture latent factors based on probabilistic framework and to employ them to predict new items [29].

Modeling the dynamic feature of user interests and finding the influential interests are two main challenges of above approaches. Profile matching and long-short-term interests try to find items which are most similar to recently selected items. These approaches instead of identifying user preference, uses similarity measure to provide a recommendation list. Latent factor approaches infer the user interests for finding desirable items, but they do not consider the influence of each interest to find preferred interests. In addition, these approaches are unable to capture the dynamics in interests.

In this paper we develop a content-based recommender system using the distance dependent Chinese Restaurant Process (dd-CRP) [30] as the underlying framework. The dd-CRP is a Bayesian non-parametric model which can be used for incremental and dynamic clustering. Using the content of selected items to construct the user profile, we adopt dd-CRP for clustering the user profile and extracting user interests. Each interest corresponds to a specific cluster. This approach results in dynamic model with no need of determining the number of interested categories (clusters) in advance. Moreover, we infer a weighted distribution over user interests to measure user preferences, i.e. the influence of each interest category for the user. The proposed method leverages these preferences to provide a more accurate recommendation list.

The rest of the paper is organized as follows: Section II provides a brief representation of distance dependent chines restaurant process and weighted distribution and their applications. Section III describes the proposed method. 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. Distance Dependent-CRP and Weighted Distribution

We use dd-CRP as a Bayesian nonparametric framework to identify clusters in user profile. This framework uses the latent factor approach to identify clusters. Also, it provides an incremental clustering approach based on observations (selected resources) with no need of determining number of clusters in advance. In the first subsection, we describe dd-CRP. Having the clusters corresponding to user interests at hand, a weighted distribution is constructed to calculate the influence of each interest.

A. Distance Dependent-CRP

The Chinese Restaurant Process (CRP) is a Bayesian non-parametric model that encodes the dependencies between the observations [31]. CRP is an alternative formulation of the Dirichlet process mixture model, providing a clustering method that determines the number of clusters from the observations. In other words, the observations that have the same latent similarity factor will be placed in the same cluster. It uses a Dirichlet process with the base distribution of G_0 and dispersion parameter, α to create the tables (cluster) distribution. CRP maps the incremental clustering task to the process of assigning customers entering sequentially to a number of infinite tables in a Chinese Restaurant. The table (cluster) distributions have the exchangeability property [32]. The assignment strategy is implemented using posterior distribution over table assignments as follows:

- The first customer is placed on the first table;
- The k^{th} table is assigned to the i^{th} customer (τ_i= k) with the probability defined in (1).

$$p(\tau_i = k \mid \tau_{1..i-1}, \alpha) = \begin{cases} \mathcal{F}(i,k) & if\ i \leq K \\ \frac{\alpha}{i-1+\alpha} & if\ i = K+1 \end{cases} \quad (1)$$

Here, K is the total number of current tables. Function $\mathcal{F}(.)$ in (1) shows the probability of a customer sitting at the k^{th} table that is proportional to m_k (the number of other customers already sitting at the k^{th} table) and is defined in (2).

$$\mathcal{F}(i,k) = \frac{m_k}{i-1+\alpha} \quad (2)$$

A larger value of the dispersion parameter results in a higher probability of picking a new table, and incrementing the number of clusters. The distance dependent Chinese Restaurant Process is an extension of CRP that allows for a non-exchangeable distribution on partitions [30]; rather than representing a partition by the customers assigned to a table, dd-CRP defines the probability distribution on groups of similar customers connected to each other. In other words, in traditional CRP, customers are assigned to tables, which in dd-CRP customers are assigned a direct relationship to another specific customer. The distance (similarity) factor can be spatial, temporal, or any other relevant characteristic that may be used to measure the similarity of two items.

The dd-CRP connects the i^th customer to the j^th customer using the assignment distribution probability as follows:

$$p(\tau_i = j\,|Sim, \alpha) = \begin{cases} Sim(i,j) & if\ i \neq j \\ \frac{\alpha}{i-1+\alpha} & if\ i = j \end{cases} \quad (3)$$

Here, the *Sim*(i, j) shows the similarity between the i^th and j^th customers. Equation (3) chooses the j^th customer as the target table for i^th customer, if the j^th customer is the most similar customer to the i^th customer. Therefore, tables of dd-CRP set up clusters of connected customers. The main goal in dd-CRP is to compute the posterior distribution of new customer assignments given a set of previous customer assignments.

The posterior inference for dd-CRP is derived using Gibbs sampling iteratively, using (4):

$$p(\tau_i^{new}|\tau_{-i}, A, \Psi) \propto p(\tau_i^{new}|Sim, \Psi) * p(A|\tau_{-i} \cup \tau_i^{new}, \Psi) \quad (4)$$

where Ψ is the set of model parameters such as α.

The first term in right hand side of (4) shows the table assignment of new item which is calculated using (3). The second term of (4) calculates the likelihood given the destination of new item. There are two cases to consider. The first case is that the new item creates a new table. Therefore, there is no change for likelihood term. Second, new item might connect to the existing tables that cause change in the likelihood. Thus, the posterior inference is done as follows:

$$p(\tau_i^{new}|\tau_{-i}, A, \Psi) \propto \begin{cases} \alpha & if\ \tau_i^{new} = i \\ Sim(.) * \mathcal{L}(.) & if\ \tau_i^{new} \neq i \end{cases} \quad (5)$$
$$, \mathcal{L}(.) = p(A|\tau_{-i} \cup \tau_i^{new}, \Psi)$$

dd-CRP involves some additional conditions for merging and partitioning tables, and similarity calculation; interested readers are referred to [30] for more details.

B. Weighted Distribution

Weighted distribution methods arise in the context of data gathering, modeling, inference, and computing, to assist in providing a unified approach in dealing with encountered data [33]. When the sampling units in observational studies do not have equal probability, the weighted distributions provide a unifying approach in model specification and data interpretation [34]. Weighted distributions take into account the method of ascertainment, by adjusting the probabilities of the actual occurrence of events, to arrive at a specification of the probabilities of those events as observed and recorded [35].

Let X be a non-negative random variable with the probability density function $f(x \in X;\, \theta)$, where θ is the parameter vector. Also, w(x) indicates the relative probability that x will be observed and recorded under parameter β. For weighted version of X with weight function w, the weighted distribution is denoted as follows:

$$f^*(x;\, \theta, \beta) = \frac{w(x;\beta)\, f(x;\theta)}{\bar{w}} \quad (6)$$

where $\bar{w}$ is the normalizing factor to achieve a valid probability value.

III. The Proposed Recommender System

In the proposed method, the user profile is a vector of user activities which represents the selected resources by the user along with time of selection. We denote the profile of user u by vector $A^u = \{a_1^u\!:(r_1, d_1), a_2^u\!:(r_2, d_2), \dots\}$, where a_i^u indicates user u selects resource r at date d. The individual user profile is employed to construct a Bayesian nonparametric framework for estimating user preference. To this end, the user interest is defined as a latent factor that indicates a group of activities with a related resource. Each user has an infinite number of interests which may increase or decrease along with user transactions. We apply dd-CRP for grouping user activities into separate clusters where each cluster (table) is referred to as an interest. For employing dd-CRP, we consider user interests as tables, and user activities as customers. Having determined a user's interests, we define user preferences as the main factors which guide future selection of resources by the user. User preferences are inferred by calculating the influence of each interest using a weighted distribution. Users prefer to select resources from interests which have more influence.

The architecture of proposed recommender system is brought in Fig. 1 and consists of two steps as follows:

- *User modeling:* using dd-CRP to construct user model based on his/her activity vector. The model exposes the actual interests of the user.
- *Inferring user preference:* constructing a weighted distribution to infer user preferences. The preferences are used for future recommendations.

In the next subsections, we describe the above two steps.

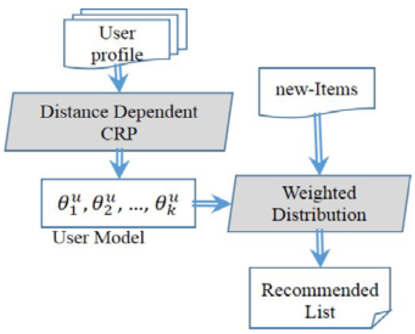


Fig. 1 The proposed model for recommender system

A. Using dd-CRP to Construct User Model

To demonstrate the process of constructing a user model, we consider a news application (based on twitter data) as our case study. In this application, the user profile consists of a vector of selected news articles with selection times. Therefore, a selected news article at a specific date/time is considered as customer of dd-CRP. The similarity between news articles is used as the similarity measure in (3). LSI [36], [37] and LDA [38] implemented in NLTK tools [39] are used to construct the similarity matrix.

By applying dd-CRP on the user profile, a set of connected activities are produced. Each group of connected activities represents a user interest their probability distribution are extracted. Theses probability distribution indicate the proportion of each interest that affects user selections. In other words, each interest returns some of the similar news articles that are assigned to the same table. We denote the result of dd-crp (for each user) as the vector $\Theta = \{(\tau = 1,\ \theta_1), \dots, (\tau = k, \theta_k)\}$, where $\tau = i$ and θ_i show a specific interest (table) and its proportion, respectively. In other words, vector Θ, represents the probability distribution of user interests. Fig. 2 shows the process of constructing user model using dd-CRP.

```
Input: A = {user activities};
Output: table (interests) partitions and their proportions;
While Iterations:
    For each activity, a_i ∈ A:
        Determining destination table of a_i, using (5);
        Update table partitions and their proportions;
```

Fig. 2 The process of constructing user model using dd-CRP.

B. Inferring User Preference

After constructing the user model and extracting the interest vector Θ, we customize (6) in (7) for populating the weighted distribution which shows the influence of each user interest. This weighted distributed actually exposes the preference of a user to select new item x. This equation calculates the preference of user to select new item x. According to (7), selecting new item x depends on the proportion of each interest and the weighted similarity of a new item related to that interest. The proportion of each interest, given new item x, is calculated by $p(x;\theta_k)$. We suppose the observations follow the categorical distribution. Therefore, $p(x;\theta_k)$ indicates the likelihood of item x given its category that is equal to θ_k. Also, the weighted similarity of new item is calculated by $w(x;\ \tau = k)$. This function calculates the average similarity of new item x related to each item of the k^{th} interest. We note that the parameter β, is replaced by the table assignment of new item x.

$$p(x;\ \Theta) = \sum_{\theta_k} \frac{w(x;\tau=k)\ p(x;\theta_k)}{\bar{w}} \tag{7}$$
$$,p(x;\ \theta_k) \sim Categirical(x;\ \theta_k) = \theta_k$$
$$,w(x;\ \tau = k) = \frac{\sum_{item \in z(k)} Sim(x,\ item)}{|z(k)|}$$
$$,z(k) = \{item|\ \tau_{item} = k\}$$

Equation (7) is used to calculate the rank of all new items then recommendation list is constructing from new items which are sorted based on their rank. Fig. 3 shows the recommendation algorithm using weighted distribution.

IV. Experimental Results

We implement the proposed method for constructing a content based recommender system. For experiments, we gather tweets as news articles along with users' access history from twitter. The provided dataset consisted of tweets from the BBC gathered in the period of Jan 14th, 2016 to Jun 7th, 2016. We preprocessed the data by removing news articles that are rarely accessed. The prepared dataset contains up to 1000 users. For comparison, a state of the art content-based recommendation system, LOGO [20] is chosen. This method aggregates the contents of activities from all users to extract a unified set of groups of items for all users. Then for each user, the whole user profile (long-term) is used to find the interested groups, and a recent partition of user profile (short-term) is employed to recommend interested items. To evaluate the results of proposed method, the measures Precision@N (P@N), Recall@N (R@N), F1-measure@N (F1@N), and discounted cumulative gain (DCG@N) are used [9], [40]. We implement the proposed method in python within NLTK tools and record results for each user. Then, the evaluation metrics are averaged over all users for different lengths (top@N) of recommendation list such as $\{10, 20, 30\}$. Table 1 shows the calculated values of evaluation metrics. These results are averaged from multiple (up to 50) runs. In addition to the results of our implementation for LOGO, we show the original reported results of this method from [20] into Table I. According to results, the F1-measure shows the superior recommendation accuracy of the proposed method. For example, with top@N = 20, the proposed method has better accuracy related to LOGO method. Also, the nDCG@N measure indicates that the recommended list of the proposed method is more consistent with user selection in comparison to the LOGO method. Also, Fig. 4 represents the graphical view of results. The solid line of this figure shows the results of the proposed method. The precision, recall, f1-measure, and discounted cumulative gain (DCG) are depicted in sub-figures of Fig. 4, respectively.

Results show that using dd-CRP to construct user model and inferring preference can improve the accuracy of content based recommender systems.

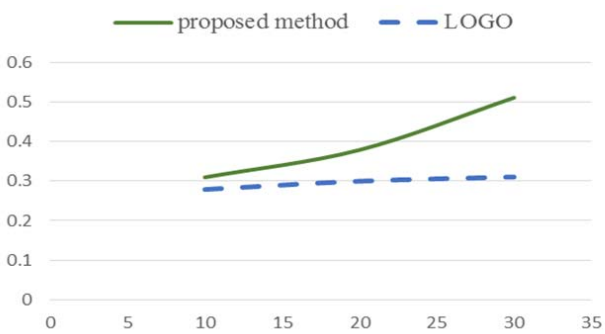
```
Input: table partitions and their proportions;
Output: Recommendation_list={}
For each new item, a_i^*:
    r_{a_i^*} = Calculate rank of new item a_i^*, using (7);
    Append r_{a_i^*} to Recommendation_list;
Return first N items of sorted Recommendation_list;
```

Fig. 3 The recommendation algorithm

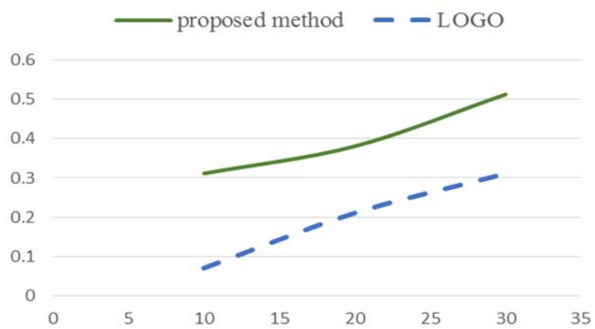
TABLE I
RESULTS OF EVALUATION METRICS

top@N	Metrics	*precision*	*recall*	*F1*	*DCG*
10	*Proposed method*	**0.31**	**0.31**	**0.31**	**0.31**
	*LOGO**	0.28	0.1	0.1	0.3
	*LOGO***	0.21	0.24	0.21	non
20	*Proposed method*	**0.38**	**0.38**	**0.38**	**0.38**
	*LOGO**	0.3	0.21	0.23	0.42
	*LOGO***	0.27	0.36	0.31	non
30	*Proposed method*	**0.51**	**0.52**	**0.51**	**0.52**
	*LOGO**	0.31	0.31	0.27	0.45
	*LOGO***	0.31	0.40	0.34	non

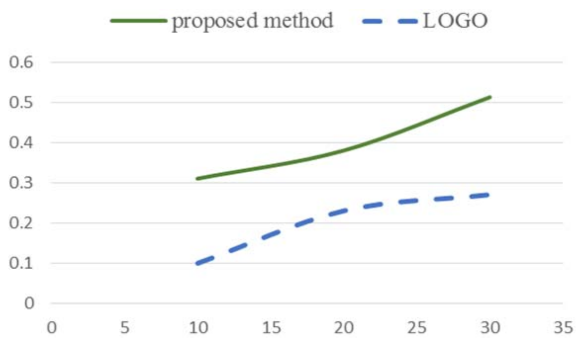
LOGO*: results of our implementation.
LOGO**: the original results that reported in [21].



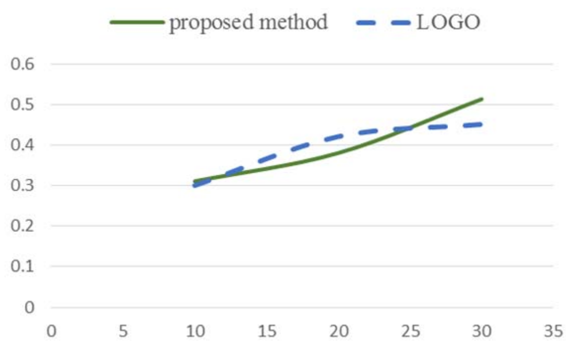
(a) Precision metric, top@N ={10, 20, 30}



(b) Recall metric, top@N ={10, 20, 30}



(c) F1-measure, top@N ={10, 20, 30}



(d) DCG metric, top@N ={10, 20, 30}

Fig. 4 The results of evaluation metrics for comparing the performance of proposed method

V. CONCLUSION

We employed the dd-CRP and weighted distribution to construct a user model in a content based recommender system. The proposed method used dd-CRP to extract latent factors from user profile. These latent factors (called user interests) indicate groups of similar items and are later used to infer preference for selection of new resources. The results of evaluating the proposed approach with a dataset containing the news tweets shows that using the proposed method to infer user preferences, improves the performance of a content-based recommender system. As for the future work, using the hybrid method to provide a richer user profile which contains contextual data can be used to better infer the user preference.

REFERENCES

[1] Linden, G., Smith, B., and York, J. “Amazon.com recommendations: Item to-item collaborative filtering”. IEEE Internet computing, 2003, 7(1), 76-80.
[2] Liu, J., Dolan, P., and Pedersen, E. R. “Personalized news recommendation based on click behavior”. In Proceedings of the 15th international conference on intelligent user interfaces, 2010, pp. 31-40.
[3] Aggarwal, Charu C. "An Introduction to Recommender Systems." Recommender Systems. Springer International Publishing, 2016, pp. 1-28.
[4] Bobadilla, Jesús, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. "Recommender systems survey." Knowledge-based systems, 2013, pp. 109-132.
[5] Gauch, S., Speretta, M., Chandramouli, A., and Micarelli, A. “User profiles for personalized information access”. The adaptive web, 2007, pp. 54-89.
[6] Adomavicius, G., and Tuzhilin, A. “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions”. Knowledge and Data Engineering, IEEE Transactions on, 2005, 17(6), pp. 734-749.
[7] Ricci, F., Rokach, L., and Shapira, B. Recommender systems: Introduction and challenges. In Recommender systems handbook, Springer, 2015, pp. 1-34.
[8] Felfernig, A., Jeran, M., Ninaus, G., Reinfrank, F., Reiterer, S., and Stettinger, M. “Basic approaches in recommendation systems”. In Recommendation systems in software engineering, Springer, 2014, pp. 15-37.
[9] Wu, H., Pei, Y., Li, B., Kang, Z., Liu, X., and Li, H. “Item recommendation in collaborative tagging systems via heuristic data fusion”. Knowledge-Based Systems, 2015, 75, pp. 124-140.
[10] Aggarwal, C. C. “Content-based recommender systems”. In Recommender systems, Springer, 2016, pp. 139-166.
[11] Aggarwal, C. C. “Knowledge-based recommender systems”. In Recommender systems, Springer, 2016, pp. 167-197.

[12] Burke, R. "Hybrid recommender systems: Survey and experiments". User modeling and user-adapted interaction, 2002, 12(4), pp. 331-370.
[13] Chen, Li, Guanliang Chen, and Feng Wang. "Recommender systems based on user reviews: the state of the art." User Modeling and User-Adapted Interaction 25.2, 2015, pp. 99-154.
[14] Garcin, F., Dimitrakakis, C., and Faltings, B. "Personalized news recommendation with context trees". In Proceedings of the 7th acm conference on recommender systems, 2013, pp. 105-112.
[15] Kompan, M., & Bielikova, M. "Content-based news recommendation". In International conference on electronic commerce and web technologies, 2010, pp. 6172.
[16] Li, L., Wang, D., Li, T., Knox, D., and Padmanabhan, B. "Scene: a scalable two-stage personalized news recommendation system". In Proceedings of the 34th international acm sigir conference on research and development in information retrieval, 2011, pp. 125-134.
[17] Das, A. S., Datar, M., Garg, A., and Rajaram, S. "Google news personalization: scalable online collaborative filtering". In Proceedings of the 16th international conference on world wide web, 2007, pp. 271-280.
[18] Pazzani, M. J., and Billsus, D. "Content-based recommendation systems". In The adaptive web, Springer, 2007, pp. 325-341.
[19] Billsus, D., and Pazzani, M. J. "User modeling for adaptive news access". User modeling and user-adapted interaction, 2000, 10(2-3), pp. 147-180.
[20] Li, L., Zheng, L., and Li, T. "Logo: a long-short user interest integration in personalized news recommendation". In Proceedings of the fifth acm conference on recommender systems, 2011, pp. 317-320.
[21] Li, L., Zheng, L., Yang, F., and Li, T. "Modeling and broadening temporal user interest in personalized news recommendation". Expert Systems with Applications, 2014, 41(7), pp. 3168-3177.
[22] Zheng, L., Li, L., Hong, W., and Li, T. "Penetrate: Personalized news recommendation using ensemble hierarchical clustering". Expert Systems with Applications, 2013, 40(6), pp. 2127-2136.
[23] Xiang, L., Yuan, Q., Zhao, S., Chen, L., Zhang, X., Yang, Q., and Sun, J. "Temporal recommendation on graphs via long-and short-term preference fusion". In Proceedings of the 16th acm sigkdd international conference on knowledge discovery and data mining, 2010, pp. 723-732.
[24] Koren, Y., Bell, R., Volinsky, C., et al. "Matrix factorization techniques for recommender systems". Computer, 2009, 42(8), pp. 30-37.
[25] Koren, Y. "Collaborative filtering with temporal dynamics". Communications of the ACM, 2010, 53(4), pp. 89-97.
[26] Forbes, P., & Zhu, M. "Content-boosted matrix factorization for recommender systems: experiments with recipe recommendation". In Proceedings of the fifth acm conference on recommender systems, 2011 pp. 261-264.
[27] Lu, Z., Dou, Z., Lian, J., Xie, X., and Yang, Q. "Content-based collaborative filtering for news topic recommendation". Aaai, 2015, pp. 217-223.
[28] Baltrunas, L., Ludwig, B., and Ricci, F. "Matrix factorization techniques for context aware recommendation". In Proceedings of the fifth acm conference on recommender systems, 2011, pp. 301-304.
[29] Sahoo, N., Singh, P. V., and Mukhopadhyay, T. "A hidden markov model for collaborative filtering". Management Information Systems Quarterly, Forth-coming, 2010, 36(4), pp. 1329-1356.
[30] Blei, D. M., and Frazier, P. I. "Distance dependent chinese restaurant processes". Journal of Machine Learning Research, 2011, 12, pp. 2461-2488.
[31] Gershman, S. J., and Blei, D. M. "A tutorial on Bayesian nonparametric models". Journal of Mathematical Psychology, 2012, 56(1), pp. 1-12.
[32] Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. "Hierarchical dirichlet processes". Journal of the american statistical association, 2012, 101(476), pp. 1566-1581.
[33] Patil, G. P. "Encountered data, statistical ecology, environmental statistics, and weighted distribution methods". Environmetrics, 1991, 2(4), pp. 377-423.
[34] Gupta, R. C., and Kirmani, S. N. U. A. "The role of weighted distributions in stochastic modeling". Communications in Statistics-Theory and methods, 1990, 19(9), pp. 3147-3162.
[35] Patil, G. P., and Rao, C. R. "The weighted distributions: A survey of their applications". Applications of statistics, 1977, 383, pp. 383-405.
[36] Lin, Y.-S., Jiang, J.-Y., and Lee, S.-J. "A similarity measure for text classification and clustering". Knowledge and Data Engineering, IEEE Transactions on, 2014, 26(7), pp. 1575-1590.
[37] Hofmann, T. "Probabilistic latent semantic indexing". In Proceedings of the 22nd annual international acm sigir conference on research and development in information retrieval, 1999, pp. 50-57.
[38] Blei, D. M., Ng, A. Y., and Jordan, M. I. "Latent dirichlet allocation". Journal of machine learning research, 2003, 3, pp. 993-1022.
[39] Bird, S. "Nltk: the natural language toolkit". In Proceedings of the COL-ING/ACL on interactive presentation sessions, 2006, pp. 69-72.
[40] Avazpour, I., Pitakrat, T., Grunske, L., and Grundy, J. "Dimensions and Metrics for Evaluating Recommendation Systems". In P. M. Robillard, W. Maalej, J. R. Walker, & T. Zimmermann (Eds.), Recommendation systems in software engineering, 2014, pp. 245-273. Berlin, Heidelberg: Springer Berlin Heidelberg.