UP-TreeRec: Building Dynamic User Profiles Tree for News Recommendation

Ming He
Beijing University of Technology
heming@bjut.edu.cn

Xiaofei Wu
Beijing University of Technology
wuxiaofei@emails.bjut.edu.cn

Jiuling Zhang
Beijing University of Technology
zhang90@emails.bjut.edu.cn

Ruihai Dong
University College Dublin
ruihai.dong@ucd.ie

ABSTRACT

Online News recommendation systems aim to address the information explosion of news and make personalized recommendation for users. The key problem of personalized news recommendation is to model users' interests and track their changes. A common way to deal with user modeling problem is to build user profiles from observed behavior. However, the majority of existing methods make static representation to user profiles and little research has focused on the effective user modeling that could dynamically capture user interests in news topics. To address this problem, in this paper, we propose UP-TreeRec, a news recommendation framework based on user profile tree (UP-Tree), which is a novel framework combining content-based and collaborative filtering techniques. First, by exploiting a novel topic model namely UI-LDA, we obtain the representation vectors for news content in topic space as the fundamental bridge to associate user interests with news topics. Next, we design a decision tree with a dynamically changeable structure to construct a user interest profile from his feedback. Furthermore, we present a clustering based multidimensional similarity computation method to select the nearest neighbor of UP-Tree efficiently. We also provide a Map-Reduce framework based implementation that enables scaling our solution to real-world news recommendation problems. We conducted several experiments compared to the state-of-the-art approaches on real-world datasets and the experimental results demonstrate that our approach significantly improves the accuracy and effectiveness in news recommendation.

KEYWORDS

News recommendation; user profiling; content-based recommendation; collaborative filtering

1 INTRODUCTION

Due to the explosive growth of the World Wide Web, people are increasingly prefer to read news online instead of reading from traditional media such as newspapers and TV. Online news sites, such as Google News and Yahoo News, collect news from various information sources around the world and provide an aggregate view of news for readers. However, with the increasing massive amount of news events released, a challenging problem of online news services is how to assist readers seek the news items are supposed to match the reader's reading preference as much as possible. This issue refers to personalized news recommendation [1, 2, 3, 4, 5, 6, 7, 8].

Existing personalized news recommendation systems strive to adapt their services to individual users by virtue of both user and news content information. Various approaches have been proposed by researchers to tackle personalized news recommendation. In general, these approaches fall into three classes: content-based methods, collaborative filtering based methods, and hybrid methods. Content-based (CB) [6, 9] methods make use of user profiles or product descriptions for recommendation. Collaborative filtering (CF) [10, 11] based methods use the past activities or preferences, such as user ratings on items, without using user or product content information. Hybrid methods seek to get the best of both worlds by combining content-based and collaborative-based methods [7, 12].

The critical problem of personalized news recommendation is to discover a proper way to model user's interests and track their changes. A simple way to solving the problem is to allow the user to configure personal filters or provide explicit input on news providers, topics or entities that the user would like to be shown content on. However, moving the burden of configuration may not ideal as the user has to update continually his interests to catch up with the ever-changing nature of news topics or be satisfied with following a limited number of providers.

Generally, user profile construction is quite difficult when faced with three major challenges. First, building profiles for individual users based on the large volume of online news articles requires computation over a massive feature space. Second, unlike other items such as movies and consumer products, news items tend to highly time-sensitive and their relevance expires quickly in a short period. The fresh news events substitute for out-of-date ones frequently, which make traditional ID-based methods such as collaborative filtering less effective. Third, people are topic-sensitive in news reading as they are usually interested in multiple specific news categories. How to target a user's interest dynamically based on his diversified reading behaviors for current candidate news is crucial to news recommender systems.

Considering the above challenges in news recommendation, in this paper, we provide a novel approach by building user profiling based on the user's feedback on news content.

In summary, the key contributions of this paper are:

- We apply Latent Dirichlet Allocation (LDA) [13] model to automatically extract the thematic structure and obtain interpretable lower dimensional representations of news content. The learned representations can further explore intrinsic relation between users and news with latent topics.
- To the best our knowledge, this is the first work leveraging UP-Tree to build a user profile based on the user's feedback on news content, which can dynamically capture users' reading interests in news recommendation.
- We propose a global-local multidimensional similarities computation method that could not only discovery clusters of news topic categories (global view) but also speed up similarity computation between user profiles in a cluster (local view).
- We provide a Map-Reduce framework that can scale to building user profiles for real-world news recommendation problem.
- Based on five real-world datasets, we conducted extensive experiments to evaluate the effectiveness of our work. The results demonstrate that our approach significantly outperforms the baseline methods.

The rest of the paper is organized as follows. In Section 2 we make a brief review the related work. We then present the details of decision tree based model used to build user interest profiles in Section 3. The overview of our recommendation framework and design details are given in Section 4 and the experiment results and analysis are presented in Section 5.

2 RELATED WORK

This section presents related work to our proposed approach.

2.1 LDA Topic Modeling for Recommendation

Topic modeling is gaining increasingly attention in different text mining communities. A topic model as a core component in user recommendation task namely UI-LDA is designed to jointly model a user's preferences with respect to the set of latent interest topics and social topic[14]. Work in [15] presents a LDA model to to generate subjects and judge the subject similarities between target user and recommended friends . Research on [16] proposes a community-based user recommendation method, namely CB-MF, which utilizes LDA for clustering users into communities to enhance the existing MF-based user recommendation. Recommendation system [17] adopts topic models at the user-level where documents are replaced by users' streams and recommends users that have a distribution highly similar to a target user. Compared to previous works, our proposed framework starts by modeling each news article as a mixture of the topics it addresses. We utilize LDA as a probabilistic topic model to learns latent user preferences and build user profile for each individual user according to the news reading experience. We specifically apply a clustering algorithm that's scalable and eventually recommend news articles based on how closely their topics match preferred topics for a user.

2.2 User Profiling

User profiling in principle is to find a representation of the user's interest in the same feature space as that of the items being recommended [18]. One of the most typical tasks involving user profiling is content recommendation [19, 20, 21]. A recent review of the interaction between user profiling and content-based recommendation can be found in [9, 10]. User profiling was also applied on personalized web search to enhance the user experience [22, 23]. However, most of the previous approaches do not take into account the dynamic nature of changing user interests. Profiles usually take a long time to capture the user interests' changes. In this paper, we proposes a decision tree based mechanism for the dynamic construction of user profiles based on implicit user feedback taking into consideration the dynamic nature of the user interest deviation and thus provide an exploration function while prior work does not.

2.3 News Recommendation

News recommendation has been received extensive attention in recommender systems. Nonpersonalized news recommendation aims to model relatedness among news [24] or learn human editors' demonstration [25]. In personalized news recommendation, CF-based methods [11] often suffer from the cold-start problem since news items are substituted frequently. Therefore, a large amount of content-based or hybrid methods have been proposed [4, 5, 6, 8, 26]. For example, [8] proposes a Bayesian method for predicting users' current news interests based on their click behavior, and [26] proposes an explicit localized sentiment analysis method for location-based news recommendation. Recently, researchers have also tried to combine other features into news recommendation, for example, contextual bandit [7], topic models [27], and recurrent neural networks [2]. The major difference between prior work and ours is that we leverage a topic model to discovery relatedness between users and news items based on clustering structure of topics for better exploration in news recommendation.

3 USER PROFILE MODELING

The modeling of user interest profiles is a key component of the news recommendation system. In this section, we first formulate the task of user profile modeling and then present the UI-LDA model that extracts user's preference according to the user behavior on news items. Specifically, we then describe our proposed model UP-Tree.

3.1 Problem Formulation

Generally, user profile modeling in news recommendation system aims to represent user interests in the same space as that of the news features for effectively retrieving news articles that are relevant to the user preferences. We formulate the news recommendation problem in this paper as follows.

Given news articles as well as user implicit feedback, we aim to recommend each user with a ranked list of articles he will interested.

3.2 Topics Extraction

LDA is an unsupervised machine learning technique that identifies latent topic information in large document collections. It uses a "bag of words" approach, which treats each document as a vector of word counts. Each document is represented as a probability distribution over some topics, while each topic is represented as a probability distribution over a number of words.

Motivated by the LDA model, we propose a novel topic model namely UI-LDA, which is an extension of LDA to model a user's preferences with respect to the latent interest topics. In UI-LDA model, each word *w* in a news article is associated with two latent variables: a user, *u* and a topic, *t*. Similarly to LDA, each user in the collection is associated with a multinomial distribution over *T* topics, denoted as θ. Each topic is associated with a multinomial distribution over words, denoted as ϕ. Here, differing from LDA, the observed variables for an individual news article is the set of users and the words in the news articles. The formal generative process of UI-LDA is as follows:

1. For each user u=1,…,U choose $\theta_\alpha \sim$ Dirichlet(α)
2. For each topic t=1,…,T choose $\phi_t \sim$ Dirichlet(ϕ)
3. For each news article d =1,…,D given the vector of users $\boldsymbol{u}_d$.
4. For each word i =1,…,N_d in the news article, conditioned on the user set u_d, choose an user $u_{di} \sim$ Uniform($\boldsymbol{u}_d$).
5. Conditioned on u_{di}, choose a topic t_{di}.
6. Conditioned on t_{di}, choose a word w_{di}.

The graphical model corresponding to this process is shown in Figure 1.

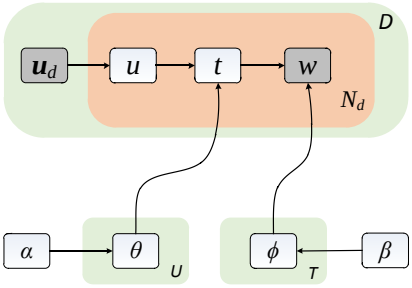

Figure 1: Graphical model for the UI-LDA topic model

Under this generative process, each topic is drawn independently when conditioned on probabilities of topics given users Θ, and each word is drawn independently when conditioned on ϕ and topic assignments **t**. The probability of the corpus **w** conditioned on Θ and ϕ is defined as

$$P(\mathbf{w}|\Theta,\phi,\mathcal{U}) = \prod_{d=1}^{D} P(\mathbf{w_d}|\Theta,\phi,\boldsymbol{u}_d), \quad (1)$$

where $\mathcal{U}$ is users of the corpus. We can obtain the probability of the words in each document $\mathbf{w}_d$ by summing over the latent variables **u** and **t**.

Among various variables presented in our UI-LDA, Θ and ϕ are random variables. In sampling and variables inference, our aim is to infer the posterior distribution $P(\Theta,\phi|,\mathcal{D}^{train},\alpha,\beta)$ in the set of users and words in the training data $\mathcal{D}^{train}$. Gibbs sampling [28] is a general efficient algorithm for obtaining Markov chain of samples to approximate latent parameters in the model. Our inference scheme is based upon the observation that

$$P(\Theta,\phi|\,\mathcal{D}^{train},\alpha,\beta) = \sum_{\mathbf{t},\mathbf{u}} P(\Theta,\phi|\mathbf{t},\mathbf{u},\mathcal{D}^{train},\alpha,\beta)\; P(\mathbf{t},\mathbf{u}|\mathcal{D}^{train},\alpha,\beta), \quad (2)$$

We obtain an approximate posterior on Θ and ϕ by using a Gibbs sampler to compute the sum over **t** and **u**. this process involves two steps. First, we obtain an empirical sample-based estimate of $P(\mathbf{t},\mathbf{u}|\mathcal{D}^{train},\alpha,\beta)$ using Gibbs sampling. Second, for any specific sample corresponding to a particular **t** and **u**, $P(\Theta,\phi|\mathbf{t},\mathbf{u},\mathcal{D}^{train},\alpha,\beta)$ can be computed directly by exploiting the fact that the Dirichlet distribution is conjugate to the multinomial.

3.3 Gibbs Sampling

To infer the above latent topics, we do posterior inference on the model using Gibbs Sampling. The process of Gibbs Sampling is described in Algorithm 1. For unknown new articles, which are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. Formally, the topic distribution is represented as a topic vector = {<t1, w1>, <t2, w2>, …}, where each entry consists of a representative word and the corresponding weight.

Algorithm 1 LDA Gibbs sampling

Input: Word vectors $\{\vec{w}\}$, hyper parameters α, β, topic number *K*.

Output: topic associations $\{\vec{z}\}$,multinomial parameters $\underline{\boldsymbol{\phi}}$ and $\underline{\Theta}$, hyper parameters estimate α, β

Global Data: Count statistics $\{n_m^{(k)}\}$, $\{n_k^{(t)}\}$ and their sums $\{n_m\}$, $\{n_k\}$, memory for full conditional array $p(z_i|\cdot)$

1. Initialize $n_m^{(k)}$, n_m, $n_k^{(t)}$, n_k to zero.
2. **for** all articles $\boldsymbol{m} \in [\mathbf{1}, \boldsymbol{M}]$ **do**
4. **for** all words $\boldsymbol{n} \in [\mathbf{1}, \boldsymbol{N_m}]$ in article *m* **do**
5. Sample topic index $\mathbf{Z_{m,n}} = \boldsymbol{k} \sim \boldsymbol{Mult}(\frac{\mathbf{1}}{\boldsymbol{K}})$.
6. Increment article-topic count: $n_m^{(k)} \mathrel{+}= 1$.
7. Increment article-topic sum: $n_m \mathrel{+}= 1$.
8. Increment topic-term count: $n_k^{(t)} \mathrel{+}= 1$.
9. Increment topic-term sum: $n_k \mathrel{+}= 1$.
10. **end for**
11. **end for**

//Gibbs sampling over burn-in period and sampling period.

12. **while** not finished **do**
13. **for** all articles $\boldsymbol{m} \in [\mathbf{1}, \boldsymbol{M}]$ **do**
14. **for** all words $\boldsymbol{n} \in [\mathbf{1}, \boldsymbol{N_m}]$ in article *m* **do**
15. $\boldsymbol{n_m^{(k)} \mathrel{-}= 1, n_m \mathrel{-}= 1, n_k^{(t)} \mathrel{-}= 1, n_k \mathrel{-}= 1}$.
16. Sample topic index $\tilde{\boldsymbol{k}} \sim \boldsymbol{p(z_i|\vec{z},\vec{w})}$.
17. $\boldsymbol{n_m^{(k)} \mathrel{+}= 1, n_m \mathrel{+}= 1, n_k^{(t)} \mathrel{+}= 1, n_k \mathrel{+}= 1}$.
18. **end for**

19. **end for**
//check convergence and read out parameters
20. **if** last read out **then**
//the different parameters read outs are averaged.
21. Read out parameter set $\boldsymbol{\phi}$ according to Eq. 4.
22. Read out parameter set $\boldsymbol{\theta}$ according to Eq. 5.
23. **end if**
24. **end while**

3.4 Dynamic Model of UP-Tree

In order to capture a user's reading interest on news articles, generally, personalized news recommendation system needs to construct the user's profile. Traditionally, the user profile can be captured by the track of user reading history. A survey of various construction techniques for user profile is provided in [29, 30].

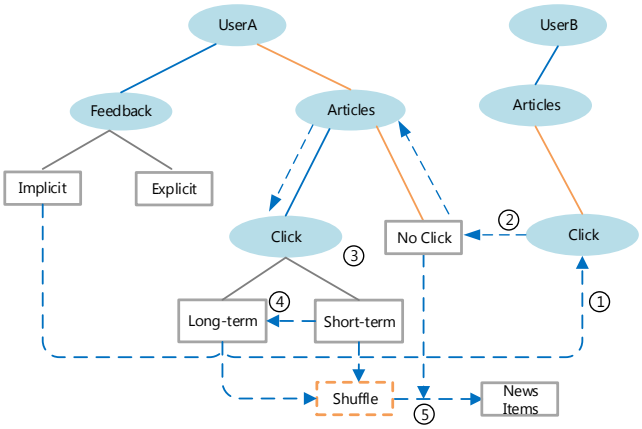

Figure 2: A UB-Tree generated using DT and Dynamics with topics

Our goal in this paper is to dynamically build user interest features for news recommendation. After applying the UI-LDA to extract the latent interest topics of users as presented in Section 3.2, a specially designed UP-Tree model is used to construct model from users' implicit feedbacks on news items and use the model dynamically modified to adapt to user interests changes for the user profile's content.

The process of dynamic UP-Tree model is illustrated in Figure 2. Our model mainly consists of three steps.

3.3.1 *Construction of UP-Tree*

In order to capture a user's reading interests on news articles, news recommender systems start with constructing the user's profile. To build user profile, we utilize the decision tree-based approach for modeling user interest based on his news reading behavior [31]. To illustrate the feature selection of decision tree (DT), we consider a small dataset in Table 1 described by four attributes namely Username, Feedback, Article, Clicked, which represent the classifiers condition of DT. Each attribute has several unique attribute values. The first row in Table 1 represents the class category of each instance. It indicates whether a particular attributes is suitable or not for User Profile in news content. We find that each feature have binary value, so CART algorithm， described by [32], which will be used to generate classification tree which selects the best feature with the Gini index and determines the optimal two value segmentation point of the feature. Figure 2 shows the decision tree model constructed using the news feature selection shown in Table 1.

Table 1: Decision tree features of classification

Username	*Feedback*	*Article*	*Clicked*
Feedback	Implicit	Clicked	Long-term
Article	Explicit	Non-clicked	Long-term
Article	Explicit	Clicked	Short-term
Feedback	Explicit	Clicked	Short-term
Feedback	Implicit	Clicked	Long-term
Article	Implicit	Non-clicked	Long-term
Article	Explicit	Non-clicked	Short-term
Feedback	Explicit	Clicked	Short-term
Feedback	Implicit	Clicked	Short-term

Note that UP-Tree involves two types of data: news articles and feedback data. There are two options for users to push each piece of news: clicked reading or neglected. The click is to represent the interest of the user at this moment, ignoring means the user is temporarily not interested, but may be interested in the future. After the user clicks to read, some article topic users prefer to stay focused and become the topic of long-term preference, so there are short-term and long-term differences. After reading an article, the user will express his attitude, like thumbs up or dislike. This kind of user behavior is called feedback. The user feedback might be explicit or implicit. Explicit feedback may correspond to ratings, thumbs-up, focus, etc. In addition, implicit feedback may correspond to user actions. The recommendation task considered in this paper is targeted for implicit feedback.

3.3.2 *Dynamics update with user interests.*

In the dynamic update UP-Tree step, user will continually update his interests to keep pace with the ever-changing nature of news topics. The procedure of dynamitic update of UP-Tree for catching user interest change presented as arrow marking in Figure 2. Firstly, we calculate *k*-nearest neighbor of user *u* based on multidimensional similarity method described in section 4.3. Based on these neighbors of *u*, we recommend *u* news topics that come from other users who have similar preferences. Second, all topics of clicked news articles by *u* will become short-term preferences and long-term preferences can be correspondingly transformed by attenuation function $f(t)$

$$f(t) = (1 + \lambda t)e^{-\lambda t}, \quad (3)$$

where λ represents the topic decay rate. The long-term user profile is constructed using a time sensitive weighting scheme. Formally, given the short-item topic Γ of a specific user A: $\Gamma_S^A = \{\Gamma_{t_0}, \Gamma_{t_1}, \cdots \Gamma_{t_n}\}$, where$t_i$ (i=0,1,2,⋯,n) means the time period of clicking $item^i$.

$$\Gamma_L^A = \Gamma_S^A - (\Gamma_{t_0} \cdot f(t_0) \cup \Gamma_{t_0} \cdot f(t_1) \cup \cdots \cup \Gamma_{t_0} \cdot f(t_n)), \quad (4)$$

where Γ_S^A andΓ_L^A are all topic set, and Γ_L^A is decayed complement set by Γ_S^A. Intuitively, we are concerned with a user's recent reading preference, as the recent one represents the user's current reading interest, if one always keep this interest constant after a period of time, this interest can be divided into long-term.

3.3.3 *Shuffle recommendation*

To make the proposed model scale up in real-world application, we propose to address this issue through the Map/Reduce implementation, as shown in Figure 3.

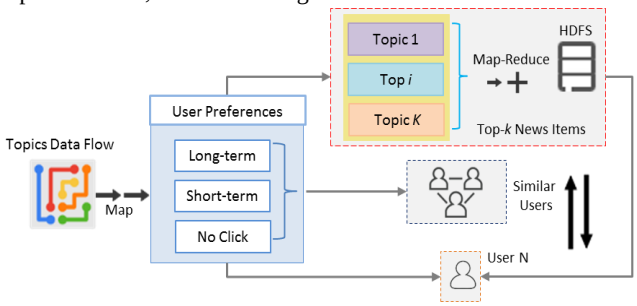

Figure 3: Shuffle recommendation model based on MR

The topics data flow is proportionately partitioned based on different types of topics (e.g., 50% of long-term, 30% of short-term and 20% of no click) and the partitioned data is taken as the input in Kafka. After physically mapping, shuffling, merging and reducing computation, the processed data is stored in HDFS as candidate news to other similar users. We define the format of input and output as flows:

$$u:\{topicj:\{<item1, dwT_1^u>, \ldots, <itemN, dwT_N^u>, \ldots, topicN:\{<itemj, dwT_2^u>, \ldots, <itemN, dwT_N^u>\}\} \rightarrow u:\{itemi:\{(topic1, \ldots, topicN):dwT_i^u\}, \ldots,$$

4 RECOMMENDATION FRAMWORK

A recommender system should build an appropriate user model to be able to predict the future activities of the user. Moreover, the accuracy of a recommender system highly depends on validity of the user model. Various researchers have used different set of techniques to develop user interests' prediction models. Firstly, it includes several machine learning techniques such as Bayesian Model, Neural Network, Support Vector Machines and some ensemble techniques like Random Forest, etc. [33–35]. Secondly, there are different methods to construct the content and structure of user profiles [36-37].Thirdly, in some studies long-term and short-term interest is also employed to model user profile [38]. Encouraged from these works, we found that the main challenge in model-based algorithms is to create a proper user model which is able to accurately discover and to incrementally adapt with changes of user interests. Therefore, we have proposed a theoretically novel framework of UP-TreeRec as shown in **Fig. 4**

Our proposed dynamic recommendation model can not only satisfy news readers' reading preferences, but also broaden their reading interests in a long run. To summarize, our proposed recommender framework consists of three core modules: Preprocessing and feature extraction Module, Dynamic UP-Tree Module and Global-Local multidimensional similarities Module. These major components and the processing flow in our framework are described as follows.

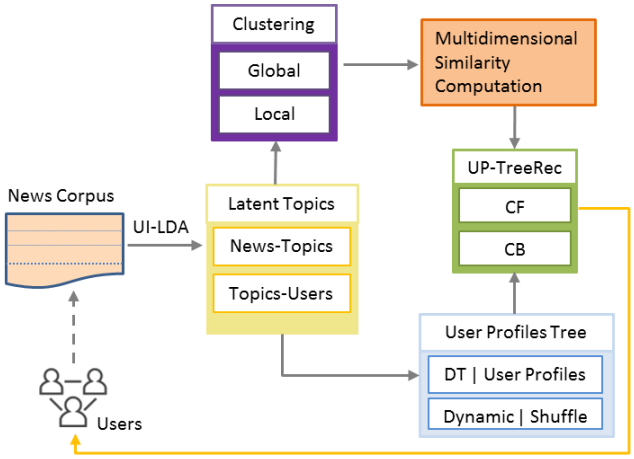

Figure 4: The proposed framework for recommender systems

4.1 Preprocessing and feature extraction

In the preprocessing and feature extraction phase, the descriptions of various news items are extracted. Although it is possible to use any kind of representation, such as a multidimensional data representation, the most common approach is to extract keywords from the underlying data. This choice is because unstructured text descriptions are often widely available in a variety of domains, and they remain the most natural representations for describing items. Our approach utilizes news contents that are generated by authors, so there are two significant problems when we use the words as they appear in text content: words stemmed from a single word and stop-words. Stemmed words are derived from the same word but they have different spelling formats, and thus they should be removed to improve performance. We used the StopWordsRemover tool takes as input a sequence of strings and drops all the stop words from the input sequences [39]. After filtering stop words, we can extract keywords from news text by UI-LDA.

4.2 Dynamic learning of UP-Tree

In section 3.3, the dynamic learning process of the UP-Tree model has been described. An enhanced UP model is constructed to predict user interests in news items, based on their history of corresponded items. In order to achieve this goal, user feedback is leveraged, which may be manifested in the form of previously specified ratings (explicit feedback) or user activity (implicit feedback). A learning model based on decision tree is constructed on this training data generated in preprocessing phase. The resulting model is referred to as the user profile because it conceptually relates user interests (ratings) to topics. More importantly, the decision tree storage model coupled with the data flow loop forms the key to dynamics.

4.3 Global-Local

A user's profile information can be enriched by analyzing other users' reading preference similar to that of the given user, which is essentially collaborative filtering. As previously mentioned, users need to focus on new topics in order to discover new taste

interests. The key issue for obtaining k-nearest-neighbor users is similarity calculation, which is not a simple display of ratings. We provide a new novel method of similarity calculation: Global-Local multidimensional similarities computation method, which is composed of two components, a global one, and another related to the local one.

Global: Newly published news collection is initially divided into small groups using Locality Sensitive Hashing (LSH) [40] purely based on topics vectors for each user. In reality, large volume of newly published news corpus requires substantial computational power. So we employ LSH to eliminate unnecessary similarity computations between unrelated articles, and get a rough separation on the original news corpus. The main purpose of using it is to reduce the dimension, that is, to map the high-dimensional feature vectors into f-bit fingerprints, and to characterize the news text similarity by comparing the Hamming distance of two news contents fingerprints.

In order to quickly navigate to specific groups, we employ hierarchical clustering with average-link on these small groups. Then, probabilistic language models are applied to summarizing news articles in each intermediate cluster and small news groups within the cluster. By doing this, user's cluster can be obtained, where leaf nodes denote small groups accompanied by their topic distributions, and internal nodes contain a couple of news groups, representing more general news topics. Once we generate the user's hierarchy and clusters, the probable similar users can be obtained by sequentially matching the specific user A onto each cluster, and select most similar cluster as global k-nearest users. When comparing the similarity between the topic distribution of each cluster Γ_C and the one of the UP-Tree Γ_U, we adopt the cosine similarity:

$$Sim(\Gamma_C, \Gamma_U) = \frac{\Gamma_C \cdot \Gamma_U}{||\Gamma_C|| \cdot ||\Gamma_U||} \quad (5)$$

Note that each cluster corresponds to a topic category. For simplicity, we only consider the similarity between topic distributions of each intermediate cluster centers and the target user's reading news topics.

Local: When user A compared with any user in the matching cluster, two compared object are mainly used: topics and feedback; the topic vector is easier, but the feedback is more complicated, and for simplicity, only the number of article feeds on the same topic is compared. Given a UP-Tree for user A and B, the similarity between A and B is computed as

$$Sim(\mathrm{U}_A, \mathrm{U}_B) = \frac{\alpha sim(\Gamma_A, \Gamma_B) + \beta sim(\mathrm{F}_A, \mathrm{F}_B)}{\sqrt{\alpha^2 + \beta^2}}, \quad (6)$$

Where α, β are parameters to control how we trust the corresponding components. $sim(\Gamma_A, \Gamma_B)$ is computed by Eq.(5), whereas $sim(\mathrm{F}_A, \mathrm{F}_B)$ are calculated by the Jaccard similarity.

5 EXPERIMENTS

In this section, we conduct experiments to test our proposed method on several real-world datasets. It is the same as most recommendation evaluation methodologies, we divide the datasets into a training set and a test set according to a certain proportion. The training set is used to learn and predict what other items a user would click and compare the predicted set with the actual set of clicks from the test set. In an offline process, the split is in the ratio 80% − 20% (train to test) and done for each user. More importantly, to evaluate the validity of the proposed recommender framework and the overall success of the dynamics in terms of UP-Tree model, we measure various evaluation metrics. The experiment results demonstrate the robustness of our approach.

5.1 Datasets

We compared our proposed method with other approaches on user's relevant news corpus of the Newsgroups and the user feedback datasets for articles read. The characteristics of the data sets are as follows:

- **News Corpus:** The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. To the best of my knowledge, it was originally collected by Crawler, probably for internet newspaper, though he does not explicitly mention this collection. The 20 newsgroups collection has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering. The data was organized into 20 different newsgroups, each corresponding to a different topic, such as economy, politics, entertainment, automobile and health. Meantime, various news corpuses from different publishers consist of 9,012 users on 18941 items from 2007 to 2015 year.
- **Feedback:** The feedback of user history activities includes explicit and implicit points. Implicit feedbacks has click or none click, additional user reading dwell time of each item. In addition, explicit feedbacks include comments, thumbs-up, focus etc. In order to meet the recommended application scenario of general news text, this paper uses implicit data when modeling profiles.

5.2 Evaluation

According to the utilized user profile information which does not provide the explicit rating of news items, we use the top-K recommended list to measure the accuracy of UP-TreeRec. The standard metrics used to measure the accuracy are Precision, Recall and F-Measure. P,R compute the ratio of news for which the correct prediction is obtained:

$$Recall = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \quad (7)$$

$$Precision = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \quad (8)$$

These quantities are also related to the (F_1) score, which is defined as the harmonic mean of precision and recall. in Eq.(9), where $\alpha = 1$:

$$F = \frac{(\alpha^2 + 1)P * R}{\alpha^2 (P + R)} \quad (9)$$

Also the percentage of the unattended topics predictions (PTP) metric is the percent of the correctly predicted unknown topic-related items by the UP-TreeRec to the total number of items in

the test ratings set of the active user. PTP are used also for evaluating the excavation ability and novelty for system, which are defined by the following formulae:

$$PTP = \frac{\sum_{i=1}^{U_A} |Novelty(UN_i^p)|}{N_i^p} \quad (10)$$

Here, N_i^p is the total number of predicted items for user u_i, U_A is the total number of the active users and $Novelty(UN_i^p)$means number of topic-related items that have never been concerned but preferred by users.

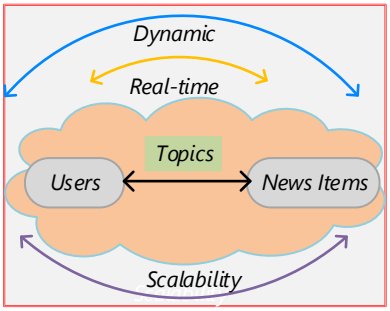

Figure 5: Three-metrics for the UP-TreeRec system

In addition to the above traditional recommendation metrics, it is also necessary to focus on measuring the dynamic characteristics of UP-TreeRec, that is, whether it can capture changes in user's interests in real time and whether it is scalable in mass news recommendation. As shown in Figure 5, in the context of LDA-based topic distribution associated with user-item, the recommendation model mainly measures three metrics: dynamic, real-time and scalability. These metrics need to rely on an assumption. Due to this paper is mainly directed at user implicit behavior, if the user clicked to read the recommended news content and stayed the long time, which shows that the user satisfaction has been to a certain extent. The formula for rate of user clicked:

$$dwT_i^u = \frac{(overTime - clickedTime)\phi^u}{IC_i}, \quad (11)$$

Where IC_i is the number of words contained in article and ϕ^u represents the normal average number of readings per second. Hence, the dwell time of user *u* reading article item *i* can help in a realistic implicit feedback scenario, which indirectly reflects how much user likes. Assume $dwT_i^u \geq 1/2$, indicates that user was interested in this news items.

5.3 Results

We conducted the experiments based on the evaluation metrics given in the previous section to obtain experimental results for analysis. In our experiments, we use the following state-of-the-art methods as baselines:

- **PRemiSE** [41] is a state-of-the-art and personalized news recommendation framework via implicit Social Experts. Compared with this paper, what they have in common is based on implicit feedback data and recommendations in the field of news.
- **LOGO** [42], in which the long-term and short-term reading preferences of users are seamlessly integrated when recommending news items to individual online users. In this paper, we also combine long-term and short-term interest models.
- **Content-based** [6] method is use of user profiles or item descriptions for recommendation.
- **Hybrid-based** [7] method, the goal is to combine the best of both worlds to create an even more robust recommender system. Our work is essentially a hybrid recommendation approach.

Table 2: Accuracy and Novelty metrics results

Top-K	*Metric*	*Precision*	*Recall*	*F1*	*PTP*
10	UP-TreeRec	0.323	0.140	0.195	0.2
	PRemiSE	0.350	0.184	0.241	0
	LOGO	0.210	0.248	0.215	0.1
20	UP-TreeRec	0.368	0.286	0.322	0.3
	PRemiSE	0.335	0.163	0.219	0.15
	LOGO	0.274	0.362	0.305	0.05
30	UP-TreeRec	0.312	0.445	0.367	0.233
	PRemiSE	0.339	0.175	0.231	0.17
	LOGO	0.314	0.402	0.341	0.147
50	UP-TreeRec	0.383	0.566	0.457	0.38
	PRemiSE	0.352	0.181	0.236	0.22
	LOGO	0.291	0.488	0.371	0.29
100	UP-TreeRec	0.446	0.817	0.577	0.27
	PRemiSE	0.345	0.192	0.247	0.11
	LOGO	0.339	0.546	0.418	0.18

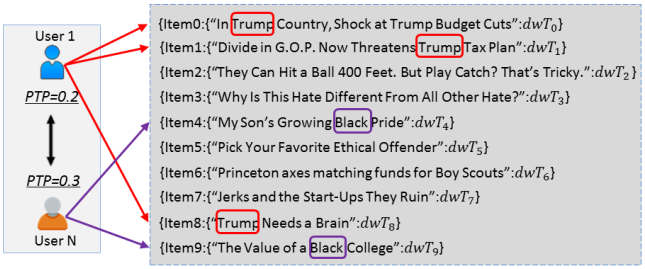

Figure 6: PTP modeling results at K=10

Table 2 shows the calculated values of accuracy and novelty metrics for the UP-TreeRec and other state-of-the-art methods for different top-K values {10, 20, 30, 50, and 100}. We plotted the results of metrics calculation for three methods according to the Precision, Recall, F1-Measure and PTP are depicted in **Fig. 9**.

The values of Precision, Recall and F1-measure comparison indicate that the accuracy of the proposed method is better than the other two method. For example, starting from top K = 20, the F1 values of proposed method has improved significantly relative to others, in which PremiSE always shows a steady trend between 0.2 and 0.3, LOGO maintains the same growth trend as UP-TreeRec, but there is a significant gap between them. The values of PTP metric in Figure 8 indicate that the recommended list of the method can solve the problem of news topic diversity for user selection in comparison to the two method. In Figure 6, the top 10 items obtained by the training of the model, the topic-related keyword that $Useri$ was interested in include "Trump", but $Useri$

clicked on the item where the $Userj$ was interested in the topic-related keyword "Black". Therefore, $Useri$'s PTP = 0.3 and $Userj$'s PTP = 0.2. In short, these curves (solid lines) in Figure 8 show that the proposed method has a better performance in predicting the future activities correctly and the accuracy increases whenever the length of recommendation list increases. At the same time, it also shows that the other method cannot recommend other users' favorite items. The recommendation type is not diversity and novelty, and the hybrid-based recommendation of this paper solves this problem very well.

The existing methods only consider contents recommended by the user, and do not consider that the user's interest may change over time. In this paper, the data sets are grouped according to the chronological order of year, month, and day. Then, the data in the same time group is further divided according to the user, so that the data is finally divided into test set units. A subset of datasets that are not time-differentiated are also used as training sets. After the two groups dataset have finished model training, take a user's profile information for comparison (as shown in Figure 7 for a user's profile topic distribution), and find that the index of time phase in the user's related news topics changes, corresponding in the Figure 7 for V_A, V_E and V_C, the recommended items will also adapt to their changes. The results show that UP-Tree model can well capture the changes of users' interests and has the dynamic of news recommendation.

Similarly, considering the Shuffle module function in the UP-Tree model, UP-Tree also needs to measure scalability and real-time performance under different data sizes: the criteria for a good recommender system is that even under large data processing conditions, it can guarantee good recommendation accuracy and recommendation timeliness.

As is evident in Figure 9, we computed the averaged F-measure of recommendation results for different data size. Before 1G, the F1 value fluctuation was not very obvious, but as the scale increased to 5G, the F1 value increased significantly. This demonstrates that the proposed model can adapt well to the requirements of big data processing under high throughput, and the larger the scale, the higher the accuracy.

Real-time performance of the three methods on different data size are shown in Figure 10.

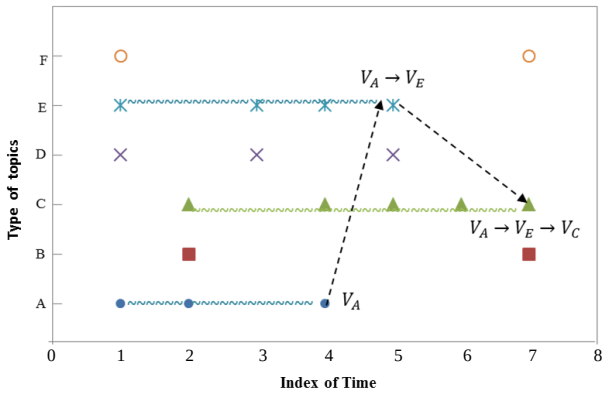

Figure 7: Capturing interest-shift behavior of UP-TreeRec over time

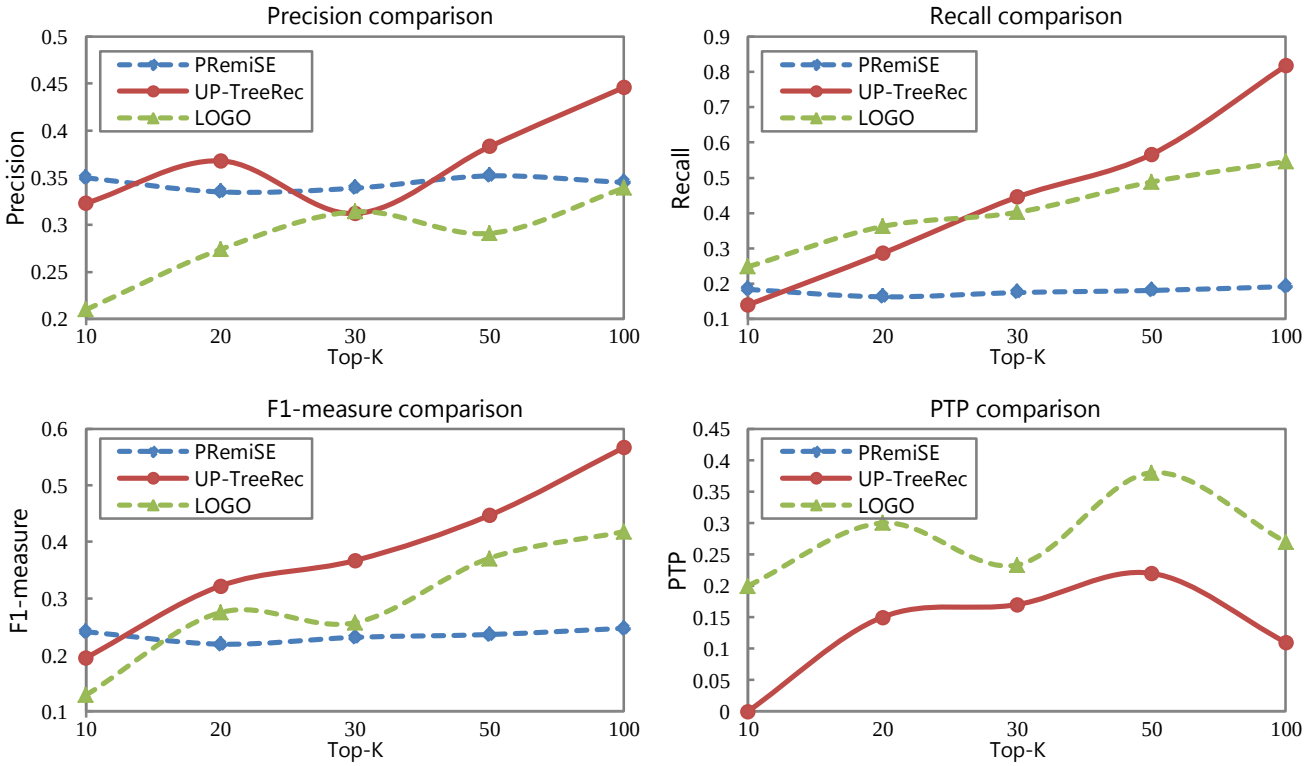

Figure 8: Comparison of accuracy metrics for UP-TreeRec

We can observe that UP-TreeRec is comparative with Content-based and hybrid with CF-CB method in data size 100k, 100M, 500M, 1G and 5G. According to the results, in the case of a large data set size, UP-Tree is obviously shorter than the other two methods, and it is consistent with scalability. The reason why UP-TreeRec can outperform other methods in predicting big data size problems, including new users and new items? The proposed model adopts the parallel mode of Map-Reduce in calculation, and the efficiency of off-line processing of user's profile data is higher. This is the main reason that is superior to other methods.

6 Conclusion

This paper proposes a hybrid recommendation framework called UP-TreeRec, which integrates content-based and collaborative filtering for news recommendation. Following this framework, we first exploit UI-LDA model to bridge the user interests and news topics. We then design the UP-Tree model, which leverage decision tree approach to dynamically capture user preferences. Next, we utilize clustering based multidimensional similarity computation method effectively select the nearest neighbor of UP-Tree. The extensive experiments we have conducted validated the effectiveness of our UP-TreeRec framework. Besides, this research sheds new light on the usage of user profile for news recommendation. We plan to further improve our recommendation service continuously in the future and apply this approach to more application scenarios.

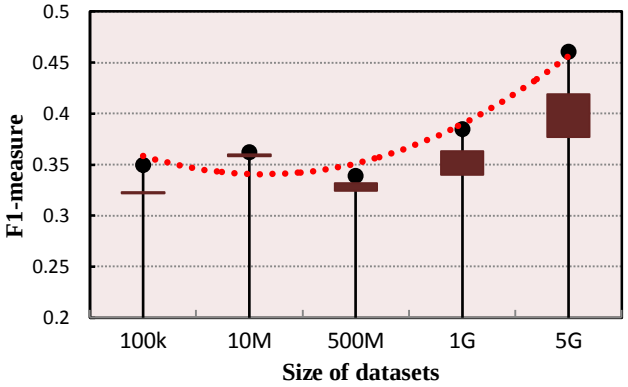

Figure 9: Average accuracy of UP-TreeRec in different data size

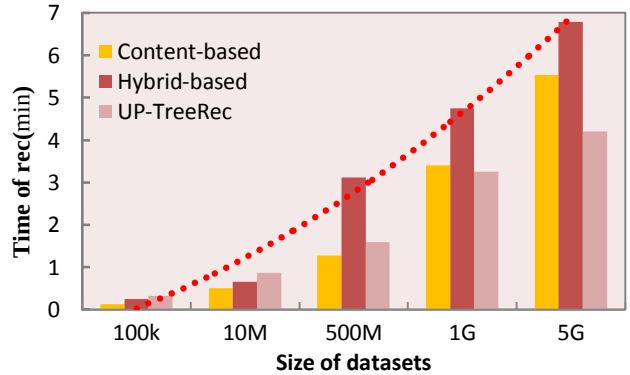

Figure 10: Comparison of three methods for rec time in different size

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable and constructive comments.

REFERENCES

[1] Hongwei Wang, Fuzheng zhang, Xing Xie, and Minyi Guo. 2018. DKN: deep knowledge-Aware network for news recommendation. In *Proceedings of the 27th international conference on World Wide Web*. ACM.

[2] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *Proceedings of the 23th international conference on Knowledge Discovery and Data Mining*. ACM, 1933–1942.

[3] Gabriella Kazai, Iskander Yusof, and Daoud Clarke. 2016. Personalised news and blog recommendations based on user location, facebook and twitter user profiling. In *Proceedings of the 39th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1129–1132.

[4] Trapit Bansal, Mrinal Das, and Chiranjib Bhattacharyya. 2015. Content driven user profiling for comment-worthy recommendations of news and blog articles. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM.

[5] Jeong-Woo Son, A Kim, Seong-Bae Park, et al. 2013. A location-based news article recommendation with explicit localized semantic analysis. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 293–302.

[6] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*. ACM, 31–40.

[7] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World Wide Web*. ACM, 661–670.

[8] Owen Phelan, Kevin McCarthy, and Barry Smyth. 2009. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*. ACM, 385–388.

[9] Pasquale Lops, Marco.de Gemmis, and Giovanni. Semeraro. Content-based recommender systems: State of the art and trends. 2011. In *Recommender Systems Handbook*. Springer.

[10] Erheng Zhong, Nathan Liu, Yue Shi, and Suju Rajan. 2015. Building Discriminative User Profiles for Large-scale Content Recommendation. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1933-1942.

[11] Chong Wang, David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 448–456.

[12] L.Li, D.Wang, T.Li, and D.Knox, B.Padmanabhan.2011.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 , ACM, 125–134.

[13] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3: 993–1022.

[14] Ke Xu, Yi Cai, Huaqing Ming, Xushen Zheng, Haoran Xie, and Tak-Lam Wong. 2017. UIS-LDA: A user Recommendation based on social connections and interests of users in uni-directional social networks. In: *Proceedings of the International Conference on Web Intelligence*, ACM, 260-265.

[15] Zhen-wu Wang, Xiao-hui Han, and Hao-ming Tian. 2017. A novel friend recommendation algorithm based on intimacy and LDA model. In Proceedings of the 9th International Conference on Information Management and Engineering, ACM, 128-132

[16] Gang Zhao, Mong Li Lee, Wynne Hsu, Wei Chen, and Haoji Hu. 2013. Community based user recommendation in unidirectional social networks. In *Proceedings of International Conference on Information and Knowledge Management*, ACM, 189–198.

[17] Marco Pennacchiotti, Siva Gurumurthy. 2011. Investigating topic models for social media user recommendation. In *Proceedings of International Conference on World Wide Web*, ACM, 101-102.

[18] Chen Li, Pu Peal. (2004). Survey of preference elicitation methods. Technical Report EPFL-REPORT-52659, EPFL.

[19] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure. Ontological user proling in recommender systems. 2004. *ACM Transactions on Information and Systems*, 22(1):54-88.

[20] G. I. Webb, M. J. Pazzani, and D. Billsus. 2001. Machine learning for user modeling. *User Modeling and User Adapted Interaction*, 11:19-29.

[21] I. Zukerman, D. W. Albrecht. 2001. Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction*, 11:5-18.

[22] Kazunari Sugiyama, Kenji Hatano, and Masatoshi.Yoshikawa. 2004. Adaptive web search based on user prole constructed without any effort from users. In *Proceedings of the 13th International Conference on World Wide Web*, ACM, 675-684.

[23] Ahu Sieg, Bamshad Mobasher, and Robin Burke. 2007. Web search personalization with ontological user profiles. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, ACM, 525-534.

[24] Yuanhua Lv, Taesup Moon, Pranam Kolari, Zhaohui Zheng, Xuanhui Wang, and Yi Chang. 2011. Learning to model relatedness for news recommendation. In *Proceedings of the 20th international conference on World Wide Web*. ACM, 57–66.

[25] Xuejian Wang, Lantao Yu, Kan Ren, Guanyu Tao, Weinan Zhang, Yong Yu, and Jun Wang. 2017. Dynamic attention deep model for article recommendation by learning human editors' demonstration. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2051-2059.

[26] Michal Kompan, Mária Bielikov á 2010. Content-based news recommendation. In *Proceedings of the 11th International Conference, on E-Commerce and Web Technologies*, Springer, 61–72.

[27] Tapio Luostarinen, Oskar Kohonen. 2013. Using topic models in contentbased news recommender systems. In *Proceedings of the 19th Nordic Conference of Computational Linguistics*. Linköping University Electronic Press, 239–251.

[28] Thomas L. Griffiths, Mark Steyvers. 2004. Finding scientific topics. In *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, Suppl 1, 5228-5235.

[29] Fabian Abel, Qi Gao, Geert Jan Houben, and Ke Tao. 2011. Analyzing user modeling on Twitter for personalized news recommendations. In *Proceedings of the User Modeling, Adaption and Personalization*, Springer, 1-12.

[30] Susan Gauch, Mirco Speretta, Aravind Chandramouli and Alessandro Micarelli. 2007. User profiles for personalized information access. The Adaptive Web, Methods and Strategies of Web Personalization, DBLP, 54-89.

[31] Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2010). An introduction to decision tree modeling. *Journal of Chemometrics*, 18(6):-.

[32] Waheed, T., Bonnell, R. B., Prasher, S. O., & Paulet, E. (2006). Measuring performance in precision agriculture: cart—a decision tree approach. *Agricultural Water Management*, 84(1–2), 173-185.

[33] Xia, Z., Xu, S., Liu, N., & Zhao, Z. (2014). Hot news recommendation system from heterogeneous websites based on bayesian model. *The Scientific World Journal*, 2014, 734351.

[34] Suglia, A., Greco, C., Musto, C., Gemmis, M. D., Lops, P., & Semeraro, G. (2017). A Deep Architecture for Content-based Recommendations Exploiting Recurrent Neural Networks. *Conference on User Modeling, Adaptation and Personalization* (pp.202-211). ACM.

[35] Xing, L., Ma, D., & Ma, B. (2015). Service Recommendation Method Based on Collaborative Filtering and Random Forest. *International Conference on Management, Computer and Education Informatization*.

[36] Liu, J., Dolan, P., & Pedersen, P. R. E. (2010). Personalized news recommendation based on click behavior. In Proceedings of the 15th international conference on Intelligent user interfaces (pp. 31–40). Hong Kong, China: ACM.

[37] Cleger-Tamayo, S., Fernández-Luna, J. M., & Huete, J. F. (2012). Top-n, news recommendations in digital newspapers. Knowledge-Based Systems, 27(6), 180-189.

[38] Yu, J., & Zhu, T. (2015). *Combining long-term and short-term user interest for personalized hashtag recommendation*. Springer-Verlag New York, Inc.

[39] Kishida, K. (2005). Technical issues of cross-language information retrieval: a review. *Information Processing & Management, 41*(3), 433-455.

[40] Lin, C., Xie, R., Li, L., Huang, Z., & Li, T. (2012). PRemiSE: personalized news recommendation via implicit social experts. *ACM International Conference on Information and Knowledge Management* (Vol.254, pp.1607-1611).

[41] Li, L., Zheng, L., & Li, T. (2011). LOGO:a long-short user interest integration in personalized news recommendation. *ACM Conference on Recommender Systems, Recsys 2011, Chicago, Il, Usa, October* (Vol.16, pp.317-320)..