Collaborating personalized recommender system and content-based recommender system using TextCorpus

Srikar Amara
Department of Computer Science and Engineering
Kalasalingam Academy of Research and Education
Virudhunagar, India
srikaramara345@gmail.com

R. Raja Subramanian
Department of Computer Science and Engineering
Kalasalingam Academy of Research and Education
Virudhunagar, India
rajasubramanian.r@klu.ac.in

Abstract—**Recommender systems aim to get the relevant data, based on the user's interests. One of the key problems of the recommender systems is to maintain the dataset and to retrieve the data, which is relevant to the user. A common solution is to track the user's preferences and showing the relevant results, however, it is a complex task in terms of time and space. The user data need to be analyzed and learnt using efficient algorithms. To address this problem, we have proposed a method to format the data in the dataset using POS-taggers using NLTK framework. In this paper, we have proposed a user-profile model which uses this tagging mechanism to provide better recommendations compared to the existing state-of-the-art recommender techniques.**

Keywords—Recommendation,Recommendersystems, NLTK framework, user-profile model, personalized recommender

I. Introduction

The recommender systems are playing a major role in people's lives, academic research and in day-to-day activities. The Recommender system is an information filtering system that filters the information depending on the user preferences or the user's activity. Generally, recommender systems rely on the crude data provided by the user during the initial stage of the recommendation and later on, the application tracks the user's activity and later it provides the personalized recommendations. The data retrieval and the acquisition is a task of prime importance. To make this process efficient we have proposed the inclusion of text corpus in the dataset. Deployment of the data from the dataset can benefit the auto-taggers. This model initially gets the crude data of the user. Depending on the activity of the user, the profile is updated and the data is recommended to the user.

Corpus is the collection of data. Text corpus is used most commonly in Annotations. An example for this annotation is parts-of-speech taggers (POS taggers). Taggers are commonly used in many applications and blogs for the allocation of the data. The most commonly used websites for the taggers include Instagram, Facebook, and many social websites. These are primarily used for categorization of the data and for searching purposes. In this paper, we have introduced taggers in the dataset and for the user's profile.

Recommender systems have a set of tools that help the user in decision making. These tools recommend/suggest data to the user depending on the user's activity. Many applications use these recommender systems, some of the applications include Spotify, google, YouTube, Facebook, and Netflix.

The main reason that many conglomerates use recommendations is for the revenue. Statistics have shown that the usage of the recommendation systems increased the sales of the company drastically when compared to the previous sales without recommenders. There are 4 types of recommender systems available: i. content-based ii. collaborative iii. personalized and iv. hybrid recommender systems. Content-based filtering (CPF) is analogous to cognitive filtering, which provides recommendations by predicting the influential factors of the user. Let's take an example such as movie application. In this movie application, the recommendation is done based on the activity of the user such as most views and likes of the user. Whereas in the Collaborative filtering, as the name indicates, this filtering is done based on collaboration between the users. Based on the similar user activity, i.e. the user's activity of the specific product, the recommendation is done. Personalized recommender systems are used to recommend the data to a user based on his activity. It provides recommendations to every user independently. This enhances the recommendation a level ahead which helps the users in decision making. Both content and collaborative have their weaknesses. To overcome them, hybrid recommender systems are introduced. This hybrid recommender systems is the combination of recommender systems with expandability and flexibility.

The current recommender systems follow a sequence of the process to maintain and process data. Even though the recommender systems work efficiently in all the perspectives, the retrieval and maintenance of data becomes difficult. We proposed a novel approach to categorize and maintain the data stored in the dataset by using the verb taggers. We designed a new profiling system for users that makes use of these tags provided in the dataset. Empirical evaluations are conducted and compared to the state-of-art approaches on the various baseline data, the results demonstrated that the proposed framework provides significant accuracy and efficiency for the recommendation.

II. Background And Related Work

A. LDA Topic Modelling

Topic modelling is significantly gaining attention in various text-mining areas. A topic hierarchy framework is leveraged as a principal unit in user recommendation processes. Least Drichilet Analysis (LDA) is typically used to derive the key point features out of the user preferences [39]. Researchers [7] used LDA to prescribe object covariates between the users under question and friend groups. Subsequent research [8] on LDA developed a community-leveraged recommender model, categorized as CB-MF. LDA topic modelling [2] is used for grouping

978-1-7281-5197-7/20/$31.00 ©2020 IEEE

candidates as communities. The community is intended to have common characteristics. Hierarchical modelling of topics [13] is trained at the candidate level, with user profile streamed as attributes between hierarchies. The id of the user exhibiting a sound correlation with the test user is recommended. Compared to the state-of-the-art researches, our proposed framework starts by modelling the documents leveraging the tags. User profiles, being a preliminary requirement for the efficient recommendations, built by randomized topic modelling using LDA. The profile composes latent user preferences of individual users. A scalable clustering algorithm inputting reading experience of users is applied and eventually recommend articles based on the tags allocated to the user profile by the usage statistics of the user. The framework is names as Tagger-UI-LDA model.

B. *User profiling*

User profiling [1] is the most common type of user data acquisition and its properties. The main reason for leveraging on user profiling techniques is to find the user's interest in the topics that are being recommended. This can be found in content-based recommender systems. And this is used for personalized [3] web searches to enhance web services. We have proposed a user profile system based on verb tagging, leveraging the dynamicity of user interests and providing an exploration function in worst case scenarios in this paper.

C. *User profile modelling*

Generally, user profile modeling is aimed to represent the user's interest in the same feature space that effectively retrieves data based on the user's preferences. We aim to recommend the articles that comes on top of user interest rosters, based on the user tag whilst comparing to the data tag. Initially the data recommendation is based on the raw data like user's background or historical data.

D. *Topic Extraction*

Unsupervised learning techniques including LDA [1,2] are typically used to extract key factors from a dataset. Topics of article datasets are usually considered as key factors, evidencing the use of LDA in topic extraction. Each data is represented as a probability distribution over a few topics. A probability distribution of the words forms the topics. Leveraging UI-LDA model [6,7,9,10], a Tagger-UI-LDA model is proposed, which will maintain a dynamic user profile overtime where tags are allocated to the user profile. In the next login, the user will be recommended based on the tags allocated to the user. Extraction process is depicted below:

```
For each user $U = \{u_1, u_2, u_3, \ldots, u\}$
        choose $\theta_{\alpha} \sim Dirichlet(\alpha)$
For each topic $T = \{t_1, t_2, t_3, \ldots, t_m\}$
        choose $\phi_{t} \sim Dirichlet(\phi)$
For each article $D = \{d_1, d_2, d_3, \ldots, d_k\}$
        given the vector of users $U_D$
For each word $I = \{I_1, I_2, I_3, \ldots, I_l\}$ in the article,
conditioned on the user set $U_D$,
    choose an user $U_{Di} \sim Uniform(U_D)$.

For each article viewed category tag $C = \{c_1, c_2, c_3, \ldots, c_p\}$

        Choose $\boldsymbol{\pi}C \sim Dirichlet(\boldsymbol{\pi})$
```

Conditioned on U_{Di}, choose a topic T_{Di}. Conditioned on T_{Di}, choose a word from I.

During this process, the topic probabilities $\theta$and the selected words focused on φ and topic assignments t is used to find the interest topic. The confined probability P is given as:

$$P(W|\theta,\phi,U) = D\prod_{d=1} P(W_d|\phi,U_D) \tag{1}$$

where U is users of the corpus. From (1), it is obvious that the weighted sum of u and t provides the probability W_d, that the token exists in the preference list of the user. Among the available hyperparameters $\theta$and ϕ are randomized. In sampling and variables inference, the aim is to infer the posterior distribution of the users and the words in the data. Hidden parameters from the samples can be extracted Markov chain models leveraging Gibbs sampling. So, the scheme is based on the observation that

$$P(\theta,\emptyset|train,\gamma,\delta) = \\ \sum P(\theta,\emptyset|t,u,train,\alpha,\beta).P(t,u|train,\gamma,\delta) \tag{2}$$

We obtain the values of θ and ϕ by using a Gibbs sampler to compute the sum over t and u. This process is carried out in two-folds. Initially, an experimental estimate of $P(t,u|train,\gamma,\delta)$ is obtained. In subsequent fold, the model for the sample of particular t and $P(\theta,\emptyset|t,u,train,\alpha,\beta)$ is developed based on the Dirichlet distribution, which conjugates the multinomial as in (2).

III. Data Gathering And Preprocessing

A. *Scrapy*

Scrapy is an open-source python framework which is used to scrape the data, including image, video, text data, from the web page in a fast, simple and yet extensible way. This framework is used in the retrieval of the data sets in the experiments we have conducted. This framework is used to scrape the visible content of the web pages such as articles, headings, URLs. Unlike the other state-of-the-art frameworks, Scrapy is fast, versatile and efficient. By using this the data in the websites such as Cnet and other technical or news websites datacan be directly retrieved to the database. This data is further processed through the tag allocation process of NLTK.

B. *Natural Language Tool Kit (NLTK)*

NLTK is a platform that works with the help of human language leveraging features like sentence tokenizer and POS taggers [5]. Python has an initial tokenizer but NLTK is more reliable and versatile. It will tokenize the sentences preprocessed with the removal of unnecessary prepositions and connectors {'this', 'is', 'a', 'token'}. The list of taggers is as follows CC: coordinating, FW: Foreign word, NN: Noun.

Therefore, this NLTK framework [14] is used to format the data in the dataset using POS tagger and that formatted data is used further in recommendation. The process of tokenization and pos tagging is shown in the Fig. 1 below.

The tokenization process takes place in several stages.The crude initial data set is processed through NLTK framework and is tokenized into tokens and the data is further processed to the final data set. The final data sets consists of the articles with the specific tags(tokens) used for the recommendation and the user profile.

In the final Dataset each article consists of the specific tag for the article.This data is used to display to the user and to give the recommendations. Initially the data with the tags which matches with the user's profile is shown and later on the data tags are updated to the user profile which therfore given as recommendations to the user during future logins. The tokenization process is as mentioned in the fig. 1.

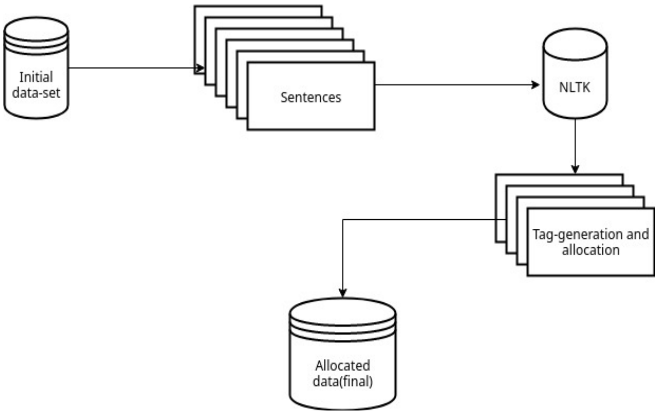

Fig. 1 .Tag-allocation using NLTK

IV. Gibbs Sampling

The process of Gibbs sampling employed in proposed framework is described in Algorithm 1. Unknown new articles are represented in the random mixtures of latent topics. Each topic is characterized by the distribution of words and tags allocated to them. The topic distribution is represented as shown in the vector $\{<t_1,w_1,c_1>,<t_2,w_2,c_2>,\dots\}$, where each entry represents the word and the corresponding weight and tags.

A. Dynamic model of the User Profile

Generally a personalized recommender system need to construct the user profile, to learn user's interest. Traditionally user's interests can be tracked by the user's history. Our goal is to dynamically build user interest features and add tags to the user profile. After applying Tagger-UI-LDA model, a special user profile is designed to construct the model from user's explicit feedback and use the model and to add tags to the user profile. Based on the tags and usage, the model will dynamically adapt to user interest changes or the user profile.

B. Construction of UP-Tree

To capture user's reading interests, the user profile is given the basic tags during the sign-up process. Decision tree models serves efficient for categorizing user interests based on user behaviours. The interests serve as nodes and behaviours form the attributes, ultimately forming the user profile.

C. T-UP-Tree algorithm

The proposed T-UP-Tree (Tagger-User Profile-Tree) algorithm is a novel approach for profiling, which enhances the recommendation for the user in a significant manner. The working application of the algorithm is as shown in Fig. 2.

Initially, the recommendations are provided by the crude data of the user, such as the background historical data. The initial tag list of the user is set to zero. If the user clicks on the article the article-topic count increases. As the sum of the article-topic is increased (Wa(K)). The tag count of the above articles is increased Cm(K).

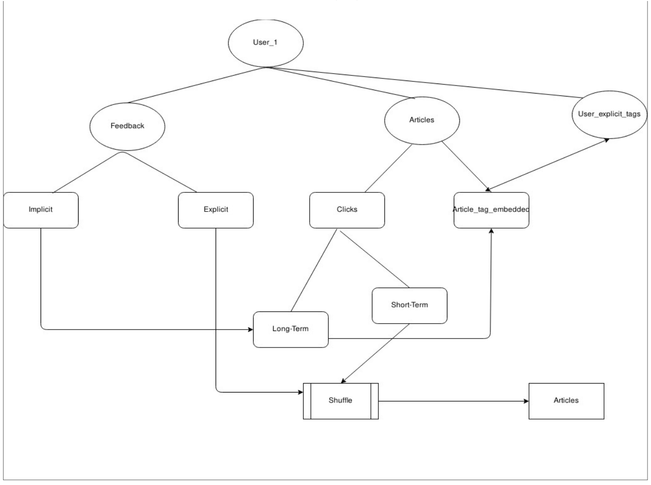

Fig. 2 T-UP-tree model and the workflow of the user profile tag allocation

Algorithm-1: T-UP-Tree

Input: Word vectors {w} ,Dataset, User profile, User profile tags, article tags

Output: Updated user profile tags, multinomial parameters.

```
//Initialization
Initialize the nm(k),n(m),nk(t),nk to zero
//Get the initial data of the user profile tag data and the articles dataset
get the initial tag data{t} from the user profile.
Get the articles related to {t} and show the data on top of the list
//User interaction
    for all articles a ∈ [ 1, A ] do
        for all words w ∈[1, W a ] in article m do
            Sample topic index Za, w = k ~Mult[1k]
            Increment article topic count:wa(k)+=1
                Increment article-topic sum: wa+=1.
              Increment article-tag count:c a(k)+=1
        end for
    end for
 //Gibbs sampling over a time period,
 //Assigning the articles maximum tag count to the user profile
 while unfinished do
        for all articles a ∈ [1, A] do
        //Allocate the maximum article topic count tags to the user profile tags(New user preferences)
            If article-topic count (c a(K)) is max
                    article-tag→ user profile tag[ ]
        end for
 end while
```

V. EXPERIMENTATION AND RESULTS

Experiments are conducted on the datasets and the grouping of data is done by the verb and PoS taggers. The tree depicting the grouping of sample data in the data set is shown in fig. 2. The data retrieval is done efficiently using UP-tree and thus the faster retrieval of data is achieved by the profile taggers and data (article) taggers. In our experiments we used the following methods as baselines: (i) Content-based [4] method; uses user profiles or item descriptions to make recommendations, (ii) PRemiSE [12]; a framework for efficient calibration of user interest to new data. The proposed framework is based on indirect feedback news recommendations in personalized conditions.

Since the existing works considered content recommended by users and not the user's interest overtime, in the proposed framework, the data in the dataset is categorized and the tags are allocated to the user profile, which made it significantly easier to organize the data and provides faster data retrieval. Fig. 3 represents the transformation of the user profile-tag overtime. As the clicks for a specific article increases, the tags of the similar articles which have been visited more will be added to the user's profile.

Another problem that we have faced with the recommender systems is the data set formatting/ categorization. We are using NLTK for the POS tag allocation to the articles. The fig. 4. represents the simulated dataset in a T-SNE visualization.

The precision of data retrieval using T-UP-Tree algorithm is significantly increased when compared to content-based or the personalized recommendation algorithms. The comparative analysis is depicted in Fig. 5. It is evident that the proposed algorithm performs better than the personal and content-based recommender systems.

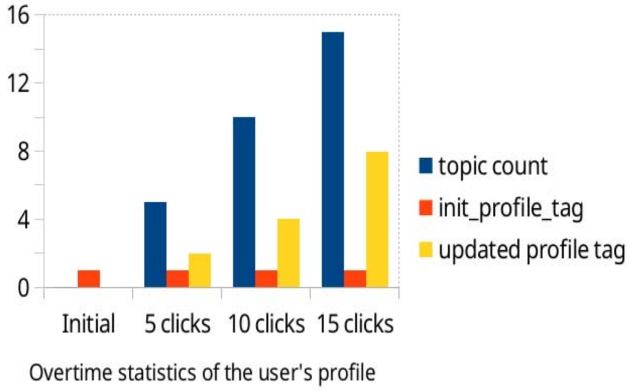

Fig. 3 Statistics of the user's profile tags

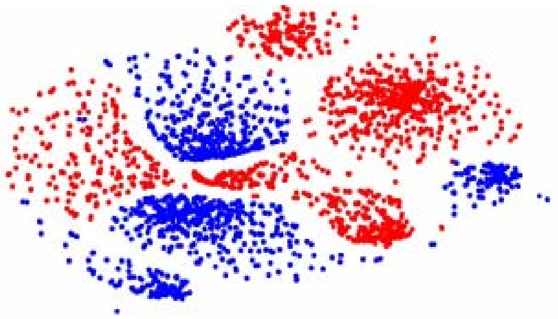

Fig.4 T-SNE visualization of the sample dataset

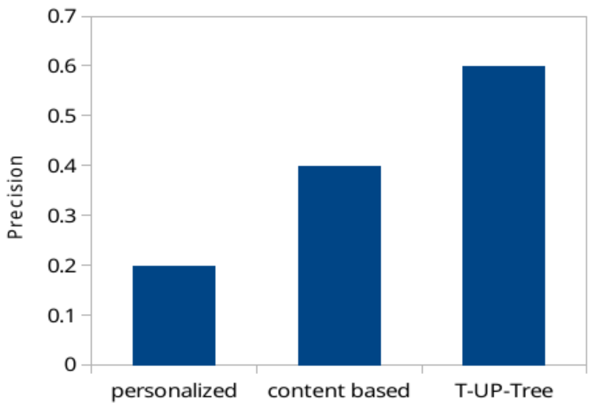

Fig .5 Precision graph for data retrieval among personalized content-based and the proposed T-UP-Tree algorithm

CONCLUSION

With many resources available on the web and the overwhelming information availability, specific information is blurred for the user. This paper presents an effective approach for efficient retrieval of data of users' interest. We have proposed the tagging system for the dataset and the new user-profile design in this paper. The tags of the user will be periodically updated and the relevant information is recommended to the user every time the user logs in. Experimental results demonstrate that this method achieves better annotation and retrieval performance than the state-of-the-art personalized and content-based recommender systems.

REFERENCES

[1] Hongwei Wang, F. Zhang, X. Xie, and M. Guo, "DKN: deep knowledge-Aware network for news recommendation." In Proc. of the 27th international conference on World Wide Web. ACM, 2018.

[2] S. Okura, Y. Tagami, S. Ono, and A. Tajima, "Embedding-based news recommendation for millions of users." In Proceedings of the 23th international conference on Knowledge Discovery and Data Mining. ACM, pp. 1933–1942, 2017,.

[3] G. Kazai, I. Yusof, and D. Clarke. "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, pp. 1129–1132, 2016.

[4] J. Liu, P. Dolan, and E. R. Pedersen. "Personalized news recommendation based on click behavior." Proc. of the International conference on Intelligent user interfaces, ACM, pp. 31–40, 2010.

[5] E. Coviello, R. Miotto, R. G. Lanckriet, "Combining content-based auto-taggers with decision-fusion", 12th International Society for Music Information Retrieval Conference, 2011.

[6] T. Bertin-Mahieux, D. Eck, F. Maillet, and P. Lamere. "Autotagger: a model for redicting social tags from acoustic features on large music databases", Journal of Music Research, vol. 37, no. 2, pp. 115–135, 2008.

[7] M. Hoffman, D. Blei, and P. Cook, "Easy as CBA: A simple probabilistic model for tagging music", Proc. ISMIR, pp. 369–374, 2009.

[8] J. Kittler. "Combining classifiers: A theoretical framework", Pattern Analysis and Applications, vol. 1, no. 1, pp. 18–27, 1998.

[9] R. Miotto, L. Barrington, and G. Lanckriet, "Improving auto-tagging by modeling semantic co-occurrences", Proc. ISMIR, pp. 297–302, 2010.

[10] S. R. Ness, A. Theocharis, G. Tzanetakis, and L.G. Martins, "Improving automatic music tag annotation using stacked

generalization of probabilistic svm outputs", Proc. ACM Multimedia, pp. 705–708, 2009.

[11] L. Li, L. Zheng, and T. Li, "LOGO:a long-short user interest integration in personalized news recommendation", ACM Conference on Recommender Systems, vol. 16, pp. 317-320, 2011.

[12] K. Xu, Y. Cai, H. Ming, X. Zheng, H. Xie, and T. Wong, "UIS-LDA: A user Recommendation based on social connections and interests of users in uni-directional social networks", Proc. International Conference on Web Intelligence, ACM, pp. 260–265, 2017.

[13] M. Pennacchiotti and S. Gurumurthy. "Investigating topic models for social media user recommendation" Proc. of International Conference on World Wide Web, ACM, pp. 101-102, 2011.

[14] https://nltk.org