A Big Data Semantic Driven Context Aware Recommendation Method

Manuel J. Barranco[1](✉), Pedro J. Sanchez[1], Jorge Castro[2], and Raciel Yera[3]

[1] University of Jaen, Jaen, Spain
{barranco,pedroj}@ujaen.es
[2] University of Granada, Granada, Spain
jcastro@decsai.ugr.es
[3] University of Ciego de Avila, Ciego de Avila, Cuba
ryerat@yandex.com

Abstract. Classical content-based recommender systems (CB) help users to find preferred items in overloaded search spaces, comparing items descriptions with user profiles. However, classical CBs do not take into account that user preferences may change over time influenced by the user context. This paper propounds to consider context-awareness (CA) in order to improve the quality of recommendations, using contextual information obtained from streams of status updates in microblogging platforms. A novel CA-CB approach is proposed, which provides context awareness recommendations based on topic detection within the current trend interest in Twitter. Finally, some guidelines for the implementation, using the Map Reduce paradigm, are given.

Keywords: Content-based · Recommender system · Context-aware recommendation · User profile contextualization · Map-reduce paradigm

1 Introduction

In recent years, information in world wide web (www) scenarios have experienced a huge increase, which causes that users need to dedicate great effort to find relevant information in www search spaces. Recommender systems (RS) are useful tools to help users in these scenarios, providing successful results in e-business [1], e-learning [2], e-tourism [3], e-commerce [4], etc.

There are different RS approaches. The most popular are collaborative filtering (CF) [5] and content-based RS (CB) [6]. CF is based on user behavior, considering that users with similar profiles could like similar items; while CB is based on item content, recommending items that are similar to such ones that the user liked in the past. There are other recommendation techniques depending on the knowledge source [7]: demographic, knowledge-based, community-based, etc.

This work is partially supported by the Spanish Ministry of Economy and Competitiveness through the Spanish National Research Project PGC2018-099402-B-I00.

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2021
C. Kahraman et al. (Eds.): INFUS 2020, AISC 1197, pp. 894–902, 2021.
https://doi.org/10.1007/978-3-030-51156-2_103

Moreover, hybrid approaches take advantage of benefits of some techniques to overcome the drawbacks of other ones. Recently, another important knowledge source in RSs is the user's context. In this way, context-aware (CA) recommendation approaches [8] are focused on context information, recommending items relevant to user needs that change over time.

In this paper, a novel context-aware content-based (CA-CB) recommendation method is proposed, which improves traditional CBs applying context awareness based on topic detection within current trend interest. There are previous works that integrate contextual information to CB recommendations [7,9,10]. Our proposal is to consider current trend interests, coming from microblogging services, such as Twitter, to build a new CA-CB recommendation method. An important step in this method is to remove noise in contextual information, caused by words that share the same lexical root. In this way, it is necessary to cluster context, identifying topics to build a contextualized user profile. Moreover, given that microblogging systems generate data at a high rate, a MapReduce approach [11] has been considered in our proposal in order to manage big data.

The rest of this paper is structured as follows: Sect. 2 provides a background of CB, CA and MapReduce, Sect. 3 presents our proposal of CA-CB recommendation method, Sect. 4 introduces some patterns for the implementation, and finally, Sect. 5 concludes the paper.

2 Preliminaries

In this section, a background of related works is included, about CB, CA and MapReduce approaches.

2.1 Content-Based Recommender Systems

Regarding the item representation, there are different CB approaches. One of the most popular is the free-text representation, where each item is described by means of unstructured data, for example, the content of a web page, a news article or a movie synopsis. Commonly, TFIDF technique [12] is applied, converting the unstructured data in data stemming words [13]. In this way, the quantity of terms is considerably reduced, unifying words with the same root, for example: recommend, recommender, recommendation, etc.

Equations 1 and 2 show the calculation of a vector of weights of terms, considering their importance in the document,

$$profile_d^{tfidf} = \{tf_{t,d} * idf_t \quad s.t. \quad t \in d\} \tag{1}$$

$$idf_t = -\log\left(\frac{|N|}{|N_t|}\right) \tag{2}$$

being $tf_{t,d}$ the number of occurrences of term t in document d, N the set of all documents and N_t the set of documents that contain the term t at least once.

This set of vectors defines a term space, more specifically, a matrix of weights of terms (columns) for each document (rows). However, the term space can be excessively wide and sparse. Latent Semantic Analysis (LSA) [6] is a technique commonly used to overcome this problem. In LSA, the term-document matrix is factorized with Singular Value Decomposition (SVD) to reduce it to orthogonal dimensions and keep the f most relevant singular values.

$$TFIDF_{(|D|\times|T|)} = U_{(|D|\times f)} * s_{(f)} * V^{t}_{(f\times|T|)} \tag{3}$$

At this point, we have a reduced feature space that replaces the wide term space, where each document has a new profile expressed in the new feature space. Then, user profiles are built through a linear combination of document profiles of items that users liked in the past [14,15] and finally, the system recommends those items more similar to the user profile.

2.2 Context-Aware Recommender Systems

User's context is another source of information which can be considered in recommendations, in order to provide more suitable results. F. Ricci [16] appointed that users' circumstances have an important influence in the users' behavior while any decision making activity. Therefore, CA recommendations will be more accurate and interesting for users, adapted to their context.

Traditional RSs try to approximate a function R applied to a two-dimensional space $R : User \times Item \rightarrow Rating$ in order to make predictions. In contrast, in CA RSs, the function is applied to a three-dimensional space $User \times Item \times Context$.

Depending on the moment when context is considered, three classes of CA approaches can be distinguished [17]: (i) pre-filtering, when the system selects only information relative to the current context, (ii) post-filtering, if item predictions are modified regarding the specific context of the users, filtering out the items which are not according to the context, and (iii) contextual modeling, when the contextual information is integrated in the recommendation model.

The purpose of the pre-filtering and post-filtering approaches is to reduce the problem of the CA three-dimensional recommendation function to a two-dimensional one, to solve it with traditional RS. However, the third CA approach suggests to integrate the contextual information in the recommendation model.

2.3 MapReduce Paradigm

Scalability is one of the main challenges to deal when facing big data problems. MapReduce [11] is a paradigm that makes possible to process big data in an scalable way, becoming one of the most popular paradigms for parallelization in general purpose applications. There are two main operations: map and reduce:

- Map: this operation takes an input given as a set of <key, value> pairs, transform it in another set of <key2, value2> pairs, group all pairs that have the same key and redistributes the work.

– Reduce: this operation combines pairs by key, applies some aggregation function and produces a smaller pair set.

Apache Spark [18] is one of the most popular MapReduce frameworks, suggested to implement the proposed method. It offers a set of in-memory primitives based on Resilient Distributed Datasets (RDDs), a structure that stores data in such a way that later computations can be easily parallelized in distributed machines. RDDs allow to cache or redistribute intermediate results, which enables the design of data processing pipelines.

3 A Recommendation Method with Context Awareness Based on Topic Detection in Current Trend Interest

In this section, the proposal of a CA-CB recommendation method is introduced. It fits into the CA approach of contextual modeling, because it integrates contextual information in the model that is used to provide recommendations. The aim of this method is to build a contextualized user profile, taking into account user preferences and the current context.

The scheme of the proposal is shown in Fig. 1. It consists of five phases, which are described in the next subsections.

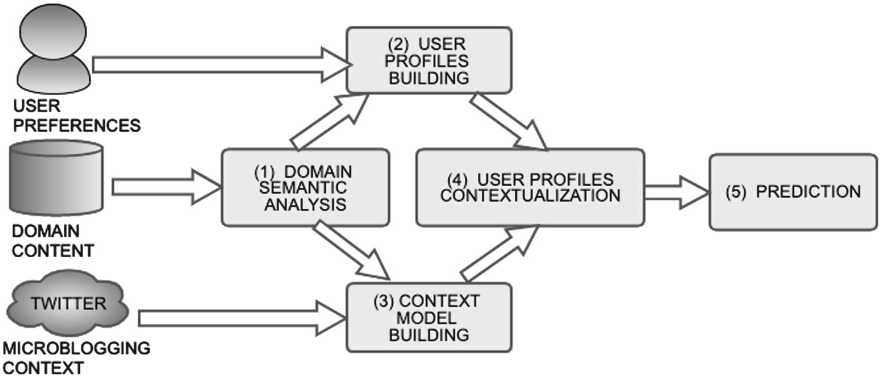
USER PREFERENCES
DOMAIN CONTENT
TWITTER
MICROBLOGGING CONTEXT
(2) USER PROFILES BUILDING
(1) DOMAIN SEMANTIC ANALYSIS
(4) USER PROFILES CONTEXTUALIZATION
(5) PREDICTION
(3) CONTEXT MODEL BUILDING

Fig. 1. General scheme of the proposal.

3.1 Domain Semantic Analysis

The domain D consists of a set of documents that describe items to be recommended. Initially, the terms that characterize documents are the words that appear in them. However, this set of terms have to be stemmed using the Porter Stemmer algorithm [13], in order to unify words that have the same lexical root. Once terms are stemmed, the TFIDF document profiles, $profile_d^{TFIDF}$, are built according to Eq. 1.

Next, the matrix of TFIDF document profiles is processed by the technique LSA to reduce its dimensionality. Applying a singular value decomposition, the initial matrix, $TFIDF$, consisting of documents, d, described in the term space, is decomposed in a reduced matrix of documents described in the feature space, U, a singular value vector, s, and a matrix of terms, t, described in the feature space, V (see Eq. 3). In this way, we obtain the profile of both terms and documents in the feature space, which compose the domain semantic model:

$$profile_d^{LSA} = \{u_{t,1}, \ldots, u_{t,f}\} \tag{4}$$

$$profile_t^{LSA} = \{v_{t,1}, \ldots, v_{t,f}\} \tag{5}$$

3.2 User Profiles Building

The previous phase has built a model consisting of terms and documents profiles. In addition, user profiles must be available in the same feature space, so that documents and user profiles are comparable. The system holds a set of ratings, $R = \{r_{u,d}\}$, that shows, for each user, the items in which he/she has expressed some interest: commenting, voting, buying, etc.:

$$R_u = \{d \quad s.t. \quad r_{u,d} \in R\} \tag{6}$$

Given that documents profiles are available in the feature space (see Eq. 4), we can generate user profiles in the same space, aggregating document profiles of preferred items:

$$profile_u^{LSA} = \sum_{d \in R_u} profile_d^{LSA} = \{\sum_{d \in R_u} profile_{d,1}^{LSA}, \ldots, \sum_{d \in R_u} profile_{d,f}^{LSA}\} \tag{7}$$

3.3 Context Model Building

In this phase, the context model is built to take into account the current trend interests in the recommendation process: documents more similar to current issues will be more relevant. The proposal is to use the status updates of a popular microblogging service, Twitter, as the source of current trend interests. From these status updates, the system generates a context model that will be used to transform the user profile into a contextualized user profile.

Firstly, terms that appear in status updates must be stemmed, and after that, the system filters out stemmed terms that do not appear in the semantic model generated in the first phase (see Eq. 5). Next, a clustering of the filtered stemmed terms takes place, so that, each cluster, c_i, determines a context topic, avoiding topics with very similar meaning. We propose a fuzzy c-means clustering algorithm [19] that groups the terms using their feature vector, $profile_t^{LSA}$, calculating distances based on cosine correlation coefficient. After that, context topic profiles are generated in the feature space, aggregating the profiles of the terms included in each cluster (see Eq. 8).

$$profile_{c_i}^{LSA} = \sum_{t \in c_i} profile_t^{LSA} = \{\sum_{t \in c_i} profile_{t,1}^{LSA}, \ldots, \sum_{t \in c_i} profile_{t,f}^{LSA}\} \tag{8}$$

3.4 Contextualizing User Profiles

Once we have user profiles and context topic profiles, our aim is to combine them to provide contextualized and personalized recommendations. Regarding this purpose, firstly, given a target user, the system selects those topics most similar to the user's profile. Our proposal is to calculate the cosine coefficient between context topics and the user's profile, selecting the c_j that has a greater coefficient (see Eq. 9).

$$c_j = \underset{c_i}{\operatorname{argmax}} \quad cosine(profile_u^{LSA}, profile_{c_i}^{LSA}) \tag{9}$$

Next, the user's profile and the selected context topic's profile are combined, providing the user's contextualized profile. We apply a convex combination (see Eq. 10): the greater the value of α, the more importance of the user's profile over the selected context topic's profile. In this way, the user's profile has been transformed to a contextualized profile, adapted to both user's preferences and context.

$$profile_{C,u}^{LSA} = \alpha * profile_u^{LSA} + (1 - \alpha) * profile_{c_j}^{LSA} \tag{10}$$

3.5 Prediction

In this phase, for each document that describes an item to be recommended, we make a prediction of suitability, considering the user's contextualized profile. Finally, the system will recommend a list of the most suitable items, described by the top N documents, sorted by $p_{u,d}$.

$$p_{u,d} = profile_{C,u}^{LSA} * \left(profile_d^{LSA}\right)^T \tag{11}$$

4 MapReduce Implementation

In this section, some guidelines for the implementation with MapReduce are provided. The phases of the proposed method can be implemented, according to the MapReduce paradigm, using one or more Map operations and one or more Reduce operations, that will be distributed and performed in a parallelized way by the workers, typically, the nodes in a computer cluster (see Sect. 2.3).

Next, we include some MapReduce patterns for implementing the consecutive phases of the proposed method:

1. Domain semantic analysis.
   - Map (stemming):$<d,\{w\}> \rightarrow <d,\{t\}>$.
   - Map: $<d,t> \rightarrow <(d,t),1>$
   - Reduce (sum): $\{<(d,t),1>\} \rightarrow <(d,t),tf>$.
   - Map: $<(d,t),tf> \rightarrow <t,1>$
   - Reduce (sum): $\{<t,1>\} \rightarrow <t,idf>$.

   For SVD algorithms, see [20,21].
2. Build user profiles.
   - Map: $<u,\{profile_d^{LSA}|d \in R_u\}> \rightarrow <(u,x \in featurespace),\{profile_{d,x}^{LSA}\}>$.
   - Reduce (sum): $\{<(u,x),profile_{d,x}^{LSA}>\} \rightarrow <(u,x),profile_{u,x}^{LSA}>$.
3. Build context model.
   - Map (stemming): $<d,\{w\}> \rightarrow <d,\{t\}>$.

   For c-means algorithms see [22].
   - Map: $<c_i,\{profile_t^{LSA}|t \in c_i\}> \rightarrow <(c_i,x),\{profile_{t,x}^{LSA}\}>$.
   - Reduce (sum): $\{<(c_i,x),profile_{t,x}^{LSA}>\} \rightarrow <(c_i,x),profile_{c_i,x}^{LSA}>$.
4. Contextualize user profiles.
   - Map: $<u,(profile_u^{LSA},\{profile_{c_i}^{LSA}\})> \rightarrow <u,(c_i,\cos(profile_u^{LSA}, profile_{c_i}^{LSA}))>$.
   - Reduce (argmax): $\{<u,(c_i,\cos(profile_u^{LSA},profile_{c_i}^{LSA}))>\} \rightarrow <u,c_j>$.
5. Prediction:
   - Map: $<u,\left(profile_{C,u}^{LSA},profile_d^{LSA}\right)> \rightarrow <u,\Big(d,profile_{C,u}^{LSA} * \left(profile_d^{LSA}\right)^T\Big)>$.
   - Reduce (top N): $\{<u,\Big(d,profile_{C,u}^{LSA} * \left(profile_d^{LSA}\right)^T\Big)>\} \rightarrow <u,\{d_1, ..., d_N\}>$.

5 Conclusions and Further Work

In this paper, we have studied the integration of contextual information in a CB, providing more suitable recommendations when considering that global interests change over time and can affect user preferences. A CA-CB recommendation method has been proposed, which builds a contextualized user profile, using the status updates that a microblogging service (Twitter) provides. The context is made up of a big set of words, which are clustered using a fuzzy c-means algorithm, in order to obtain a meaningful set of topics. In addition, given the large size of the data, a MapReduce approach has been considered to achieve better performance.

Further works will be oriented to apply this method in different scenarios, using Spark as the MapReduce framework to manage big data.

References

1. Lu, J., Shambour, Q., Xu, Y., Lin, Q., Zhang, G.: A web-based personalized business partner recommendation system using fuzzy semantic techniques. Comput. Intell. **29**(1), 37–69 (2013)
2. Yera Toledo, R., Caballero Mota, Y.: An e-learning collaborative filtering approach to suggest problems to solve in programming online judges. Int. J. Dist. Educ. Technol. **12**(2), 51–65 (2014)
3. Noguera, J., Barranco, M., Segura, R., Martínez, L.: A mobile 3D-GIS hybrid recommender system for tourism. Inf. Sci. **215**, 37–52 (2012)
4. Rafailidis, D., Nanopoulos, A.: Modeling users preference dynamics and side information in recommender systems. IEEE Trans. Syst. Man Cybern. Syst. **46**(6), 782–792 (2016)
5. Koren, Y., Bell, R.: Advances in Collaborative Filtering, pp. 77–118. Springer, Boston (2015)
6. de Gemmis, M., Lops, P., Musto, C., Narducci, F., Semeraro, G.: Semantics-aware content-based recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 119–159. Springer, Boston (2015)
7. De Pessemier, T., Courtois, C., Vanhecke, K., Van Damme, K., Martens, L., De Marez, L.: A user-centric evaluation of context-aware recommendations for a mobile news service. Multimed. Tools Appl. **75**(6), 3323–3351 (2016)
8. Adomavicius, G., Tuzhilin, A.: Context-Aware Recommender Systems, pp. 191–226. Springer, Boston (2015)
9. Parikh, N., Sundaresan, N.: Buzz-based recommender system. In: Proceedings of the 18th International Conference on World Wide Web, WWW 2009, pp. 1231–1232. ACM, New York (2009)
10. Ponzanelli, L.: Holistic recommender systems for software engineering. In: Companion Proceedings of the 36th International Conference on Software Engineering, ICSE Companion 2014, pp. 686–689. ACM, New York (2014)
11. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)
12. Erra, U., Senatore, S., Minnella, F., Caggianese, G.: Approximate TF-IDF based on topic extraction from massive message stream using the GPU. Inf. Sci. **292**, 143–161 (2015)
13. Porter, M.F.: An algorithm for suffix stripping. Program **14**(3), 130–137 (1980)
14. Symeonidis, P., Nanopoulos, A., Manolopoulos, Y.: Feature-weighted user model for recommender systems. In: Proceedings of the 11th International Conference on User Modeling, pp. 97–106. Springer-Verlag (2007)
15. Bambini, R., Cremonesi, P., Turrin, R.: A Recommender System for an IPTV Service Provider: a Real Large-Scale Production Environment, pp. 299–331. Springer, Boston (2011)
16. Ricci, F.: Contextualizing recommendations. In: Conjunction with the 6th ACM Conference on Recommender Systems (RecSys 2012), ACM RecSys Workshop on Context-Aware Recommender Systems (CARS 2012). ACM (2012)
17. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Recommender Systems Handbook, ch. 3, pp. 217–253. Springer, US (2011)
18. Zaharia, M., Xin, R.S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M.J., et al.: Apache spark: a unified engine for big data processing. Commun. ACM **59**(11), 56–65 (2016)

19. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: The fuzzy c-means clustering algorithm. Comput. Geosci. **10**(2–3), 191–203 (1984)
20. Ulrey, R.R., Maciejewski, A.A., Siegel, H.J.: Parallel algorithms for singular value decomposition. In: Proceedings of 8th International Parallel Processing Symposium, pp. 524–533. IEEE (1994)
21. Berry, M., Mezher, D., Philippe, B., Sameh, A.: Parallel computation of the singular value decomposition. Research Report RR-4694 (2003)
22. Ludwig, S.: MapReduce-based fuzzy c-means clustering algorithm: implementation and scalability. Int. J. Mach. Learn. Cybern. **6**, 04 (2015)