Promoting Diversity in Content Based Recommendation using Feature Weighting and LSH

Dimosthenis Beleveslis and Christos Tjortjis [0000-0001-8263-9024]

The Data Mining and Analytics research group, School of Science & Technology, International Hellenic University, 14th km Thessaloniki – Moudania, 57001 Thermi, Greece
c.tjortjis@ihu.edu.gr

Abstract. This work proposes an efficient Content-Based (CB) product recommendation methodology that promotes diversity. A heuristic CB approach incorporating feature weighting and Locality-Sensitive Hashing (LSH) is used, along with the TF-IDF method and functionality of tuning the importance of product features to adjust its logic to the needs of various e-commerce sites. The problem of efficiently producing recommendations, without compromising similarity, is addressed by approximating product similarities via the LSH technique. The methodology is evaluated on two sets with real e-commerce data. The evaluation of the proposed methodology shows that the produced recommendations can help customers to continue browsing a site by providing them with the necessary "next step". Finally, it is demonstrated that the methodology incorporates recommendation diversity which can be adjusted by tuning the appropriate feature weights.

Keywords: Content-Based (CB) Recommendation Systems (RS), eCommerce, Minhash, Locality-Sensitive Hashing (LSH), Decision Support.

1 Introduction

A Recommendation System (RS) refers to an intelligent system that produces suggestions about items to users. Its role is to predict items that are likely to be of interest to the user. They provide personalized information by learning the user's interests through his/her interaction with items. The way that these recommendations are produced depends on the specific domain and the desired results. In general, they are based on item similarities, user preferences, past purchases and actions related to items and users.
Recommended items can potentially be anything that a user is looking for, such as products, movies, songs, services etc. [25, 26, 27]. E-commerce is a domain that RS are used extensively [1, 2]. Online stores need to offer customers a similar or even better shopping experience than that of an actual store.
A customer can see product recommendations in several parts of an e-commerce site including the 'product page', showcasing a specific product, along with its detailed description and features [3]. The user can add the product to their cart or visit an alternative product page that better matches their preferences. The aim of the system is to display relevant items on such a page and help customers to continue browsing the site

by providing them with the necessary “next step”. The recommended products must be similar to the target product, by considering the importance of their features.
However, it is important to achieve diversity in recommendations because customers do not need a list of almost identical products. For example, when a user views a laptop of a specific brand and receives recommendations corresponding only to similar laptops of the same brand. In this case, the user will not be able to visit the product page of a laptop of an alternative brand by navigating through the recommendations.
Furthermore, the logic of such a RS relies on the idea of finding similar products by calculating similarities among them [4]. This is computationally heavy and time-consuming, for sites with thousands of products. Calculating the similarity between a product and other products, in order to find the most similar ones, is usually inefficient. So, it is important to apply more sophisticated methods that reduce the complexity and computational cost without compromising accuracy.
The scope of this research is to implement a Content-Based (CB) RS suitable to the ‘product page’ of an e-commerce site [5]. Specifically, we address the problem of product representation by incorporating a weighted method that offers the functionality to customize the importance of product features. In that way, the logic of the methodology is adjusted to the needs of the site. In addition, recommendation diversity is accomplished without compromising similarity. Furthermore, this work addresses the problem of efficient product similarity calculations for sites that dispose thousands of products. This problem is tackled by incorporating a Locality-Sensitive Hashing (LSH) method [6, 7]. Finally, our implemented heuristic approach is being evaluated based on real data from two e-commerce sites.
The remainder of the paper discusses related work and the data used in sections 2 and 3 respectively. The proposed methodology are detailed in section 4. Section 5 presents results and evaluation. Section 6 concludes the paper.

2 Related Work

Zhang et al. addressed the problem of high dimensionality, traditionally faced by RS [8]. E-commerce platforms with very big number of items and users need to incorporate large volume of data in the recommendation process. This makes the generation of real time recommendations inefficient. They incorporated LSH techniques in a Collaborative Filtering (CF) approach in order to reduce time complexity. Minhash hashing method was used for binary data and simhash for real-valued data. Similar candidate pair identification was performed through LSH to increase efficiency of similarity computing, the most time-consuming task for traditional CF RS. By conducting experiments on synthetic and real-world datasets, it is shown that LSH can approximately preserve similarities of data whilst significantly reducing data dimensions.
Aytekin et al. use LSH to introduce a RS that can scale with increasing amounts of data [9]. They improved the LSH based recommender algorithms and systematically evaluated LSH in neighborhood-based CF. By experimenting with real datasets, they presented algorithms that have better execution time than standard LSH-based

applications, whilst preserving prediction accuracy and producing recommendations with diversity.

Debnath et al. proposed a hybridization of CB and CF recommendation, which incorporates product attribute weighting [10]. They argued that human judgment of similarity between two items often gives different weights to different attributes and that RS need to consider this. The weights refer to the importance of each product attribute to customers and are estimated from a set of linear regression equations obtained from a social network graph, which captures human judgment about similarity of items. The proposed methodology is compared with CB methods that consider the importance of different products features as equal. The evaluation is based on IMDB recommendations as benchmark. The proposed method outperforms simple methods; hence, the effectiveness of feature weighting is demonstrated.

Barranco et al. proposed a feature weighting method to improve the CB filtering in cases of multi-valued item features [11]. They argued that a user considers some features as more important than others. This represents an implicit feature weighting, which can be subjective. The weight of each feature is computed according to: (i) the entropy or amount of information provided (the more entropy the higher the weight), and (ii) the correlation between items chosen by the user in the past and the values of some features of the set of items.

K. Bradley argued that recommendation diversity is important, and that traditional CB RS suffer from potentially poor diversity [23]. He also questioned the blind faith in the similarity assumption. A diversity preserving similarity-based retrieval algorithm is proposed. Multiple strategies for retrieving k items from a collection, each focusing on a different way of increasing diversity is presented. Finally, he concluded that an over emphasis on diversity can result in a corresponding drop in similarity.

3 Data

Two datasets were used to evaluate the proposed RS. The first one was created by scraping a real e-commerce site and comprises information for thousands of products. The second one was created by preprocessing an existing dataset that has been used for similar purposes in the past [27]. These datasets are described next.

3.1 Bestprice dataset

Bestprice is a new dataset created for the purposes of this research. Data were gathered by scraping BestPrice.gr, a commercial site, where a customer can compare product prices across numerous e-shops [12]. A huge variety of product categories and subcategories is available. We used a subset consisting of data regarding 29.541 products that belong to 6 main technology categories, each comprising several subcategories, 22 in total. Additionally, there are 515 different product brands. A product title is also available for each item, along with a set of 10 product recommendations for each product, provided by the site in each product page, which were used for evaluation.

The dataset was preprocessed in order to assign a textual representation to each product. This text holds information that characterizes each product and is produced by concatenating its title, category, subcategory and brand. Each of these features were preprocessed before being added to the final text. First, the textual data of each feature were transformed to lowercase. Then categories, subcategories and brands were discriminated from other words by adding the suffixes '_cat', '_subcat' and '_brand' respectively. This helps to assign different weights to these specific words. Specific symbols that do not add any value were removed from the title. Finally, a string that consists of the above textual features was created for each product. This set of strings was used to create the TF-IDF matrix [13]. The information available and the final textual representation for a product is presented in table 1.

Table 1. Product Details for Bestprice dataset.

Title	Category	Subcategory	Brand	Text
Omega Ice Box	Laptop_pc	Laptop_bases	Omega	omega_brand ice box laptop_pc_cat laptop_bases_subcat

3.2 Retailrocket dataset

Retailrocket dataset is a dataset published by Retail Rocket that has been used in RS [14]. The data were collected from a real-world e-commerce website. We selected a subset of 28.241 products that belong to 6 main categories and 37 subcategories to produce a dataset with customer sessions. A session refers to a group of user interactions with an e-shop that take place within a given time frame. For example, a session may contain the information that a customer first viewed several products, added some to their cart and finally purchased them.
This session-based dataset was used to determine the product pages that users visit after viewing each product. An important aspect we considered was the number of consecutive events related with each other. We call this window size. Window size equal to 1 means that a product that a user viewed is linked only to the product that they visited next. Respectively, window size equal to 2 means that a product that a user viewed is linked to the next two products they visited. Furthermore, a representative text, that consists of the title, category and subcategory, has been assigned to each product. This set of texts was used in order to calculate the TF-IDF matrix.

4 Methodology

The heuristic approach for the proposed CB RS is based on feature weighting and LSH. The methodology consists of three parts. The first part refers to the method that is used in order to represent the set of products as a weighted matrix. The second is the weighted Minhash method that is used to create a compressed representation of each product by its Minhash signature and to approximate the Jaccard similarity [22] of two sets. The last part is the efficient production of recommendations based on LSH. The

implementation of the methodology based on the Bestprice dataset is presented in the following three subsections and in Figure 1.

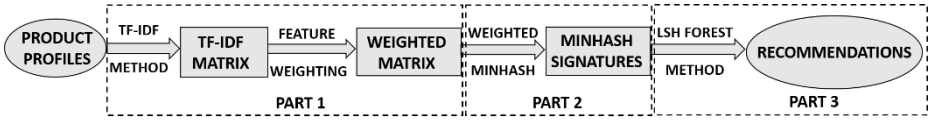

Fig. 1. Methodology diagram

4.1 Weighted method

The first part involves creating a TF-IDF matrix for the set of 29.541 products. Typically, different features carry different amount of information. A word in a product title may be more informative than the rest, and this must be considered when calculating similarity. In our case, each product is represented by a text that is the concatenation of the product title and three specific features, as described in section 3.1. The terms that correspond to the three specific product features are initially considered to be simple terms. These features are the category, subcategory and brand of each product. The total number of products is 29.541 and the respective corpus consists of 35.405 unique terms. The target is to create a matrix that represents how important each term is for each product. A weight is assigned to each unique term for each product text. This weight corresponds to the significance of the term for each specific product. The more texts a term appears in, the less informative it is considered to be; thus it gets a smaller weight. Respectively, the significance of a term in a product text increases with the occurrences of it in that specific text. In particular, the TF-IDF matrix [13] has been created by using the corresponding functions of the 'scikit-learn' python library [15]. Furthermore, specific terms of the corpus have been given extra weight. These terms correspond to the three product features that are parts of each product text. The target is to have a functionality with which we can easily adjust the importance of each feature in the calculation of product similarities and enhance recommendation diversity, without compromising similarity or efficiency.

Assuming a products' textual representation that consists of N terms and that n of them correspond to specific product features, we assign a weight w_i (w_1,w_2,...,w_N) to each of the N terms by applying the TF-IDF method. Then we adjust the importance of those n product features by multiplying their weights by small constants, while the rest of the weights remain the same. By multiplying a feature's weight with a constant greater than 1, this is considered as more important during the calculation of product similarities. The multiple scenarios that were implemented are presented in section 5.1.

4.2 Weighted Minhash

The second part of the methodology aims to create a compressed representation for each product by applying the Minhash approach. Specifically, a method that incorporates real number weights has been used since the set of products is represented by a TF-IDF matrix. The weighted Minhash algorithm [16, 17, 18, 19] that is available in

'datasketch' python library [20] is used. In practice, a Minhash signature has been created for each product based on the corresponding TF-IDF array. Hence, each product is represented by a much smaller array than before. Specifically, while the TF-IDF array consists of 35.405 elements, the Minhash signature consists of only 128 elements. The length of the signature corresponds to the 'sample_size' parameter that can be adjusted accordingly as by increasing the number of samples, a better accuracy is accomplished, at the expense of slower speed.

4.3 LSH Forest and Recommendations

In the third phase, the methodology uses the Minhash signatures in order to approximate the Jaccard similarity between products by applying the LSH approach. In particular, the LSH approach is used in order to find the most similar products to each product based on their Minhash signatures. In particular, we search for the k-top similar products that correspond to k recommendations. For this reason, a variation of LSH that is known as LSH Forest [22] is used. Minhash LSH Forest, uses the Minhash representation of the query product and returns the k-top matching products that have the approximately highest Jaccard similarities with the query product. Hence, it is not necessary to pre-define a specific threshold for the Jaccard similarity score. In this way, we produced k recommendations for each of the 29.541 products in the Bestprice dataset.

5 Evaluation

The evaluation of the RS consists of two parts. The first part is based on the Bestprice dataset that was created by scraping a real e-commerce site and is presented in paragraph 3.1. The aim is to compare the recommendations of our methodology with those that were available on the site, concerning certain aspects. Specifically, we compare the diversity of the recommendations and we examine whether we can achieve different results by tuning the weights of the product features. The second part is based on the Retailrocket dataset that consists of real sessions as described in paragraph 3.2. The aim is to examine whether the recommendations that the proposed methodology produces would help users navigate an e-shop and prevent them from leaving the site without finding products that meet their preferences.

5.1 Recommendation diversity

In this part of the evaluation process, the recommendation diversity of our methodology is examined. We evaluate the recommendations that our methodology produced for the Bestprice dataset. Finally, we compare the recommendations that our methodology produced with those that were scraped from the respective site. The dataset is presented in section 3.1. A subset of 1.182 products has been used in the evaluation process.
Having applied the methodology that is presented in detail in section 4, we have produced a set of ten recommendations for each product page. This number matches the number of recommendations that are available for each product in the Bestprice dataset.

The experiment was conducted for 11 scenarios that are presented in table 2. In each scenario we consider the importance of the product features (brand, category and subcategory) to be different, by tuning the corresponding weights (W_brand, W_cat and W_subcat). The average number of different brands, categories and subcategories in those sets of 10 recommendations is presented in table 2.

Table 2. Average # of brands, categories, subcategories in the sets of 10 recommendations

	W_brand	W_cat	W_subcat	Brands	Categories	Subcategories
Scenario1	0.5	1	1	3.07	1.16	1.30
Scenario 2	1	1	1	2.54	1.16	1.30
Scenario 3	1.5	1	1	2.16	1.16	1.31
Scenario 4	1	1	0.5	2.47	1.21	1.40
Scenario 5	1	1	1.5	2.58	1.14	1.25
Scenario 6	1	0.5	1	2.51	1.20	1.33
Scenario 7	1	1.5	1	2.63	1.12	1.29
Scenario 8	0.5	0.5	1	2.94	1.20	1.40
Scenario 9	1.5	1.5	1	2.19	1.14	1.26
Scenario 10	0.5	1	0.5	3.01	1.20	1.32
Scenario 11	1.5	1	1.5	2.23	1.13	1.29
BestPrice	-	-	-	5.16	1.0	1.0

In the first 3 scenarios, we see that the number of different brands decreases by increasing the corresponding weight. This means that considering the product brand as more important in the similarity calculations, results in recommendations that include more of the same brand for each product page. Respectively, in scenarios 4 and 5, the number of different subcategories decreases by the increase of the corresponding weight. The same happens for the product category feature in scenarios 6 and 7. Furthermore, in the rest of the scenarios, the weights of different combinations of the product features are tuned. The characteristics of the recommended products are affected by the weights that are assigned to each product feature.

In general, by observing table 2, we see that the methodology incorporates diversity as it does not recommend products that are almost the same with each other. Specifically, products from more than one brand are present in each set of 10 recommendations. Also, by assigning the appropriate weights, we have product recommendations from more than one subcategory. There are also cases in which the recommended products belong to more than one category.

The recommendation diversity is very important for the quality of the system [23]. In CB RS, diversity can be as important as similarity [24]. Similarity assures that the recommended products are similar to the target product. Diversity means that the recommended products are not very similar to each other. The importance of the recommendation diversity can be explained through the following example. Assume that the target product is a Dell laptop and the system recommends 10 different Dell laptops. This might probably mean that the recommended products are very similar to the target product. However, the user will not have the option to move to a laptop of a different brand

through the recommendations. Similar problems will occur in a case when all the recommended products are of the same subcategory or category.
In the last row of table 2, we see the respective statistics in the product recommendations that were scraped from the site (BestPrice.gr). It is obvious that there is diversity in the recommendations concerning the brand feature. There is an average of more than 5 different brands in each set of 10 product recommendations. However, there is no diversity in the cases of subcategory and category. The system recommends products that belong only to the category and subcategory of the target product. As a result, a user does not have the option to move to a product of a similar subcategory through clicking one of the recommendations.

5.2 Session based

By applying our methodology, we produced a set of ten recommendations for each product page in the Retailrocket dataset. The aim is to determine whether the recommendations would help customers navigate through the e-commerce site, by offering them the required next step in order to move from a product page to another.
For this purpose, we compare the recommendations with the product pages that users visited after viewing each single product. The test was conducted for multiple scenarios in which the importance of specific product features was tuned. These product features are the product category and subcategory. In addition, each scenario was tested for a window size up to 3. Table 3 presents the percentage of cases that at least one recommended product matches with one product view, for the scenarios where subcategory weights were tuned.
It turns out that the percentage of cases that at least one recommended product matches the real product views is between 94% and 97%, regardless the feature importance. The percentage increases with the increase of the window size. Similar results were accomplished by tuning the category weights. Hence, in most of the cases, the recommendations that our methodology produces would potentially help users move to a similar product page by clicking one of the recommended products.

Table 3. Percentage of cases that at least one recommended product matches with views.

Window	Subcategory Weight			
Size	0.5	1	1.5	2.0
1	94.74 %	94.79 %	94.84 %	94.86 %
2	96.90 %	97.01 %	96.98 %	97.03 %
3	97.68 %	97.73 %	97.68 %	97.73 %

6 Conclusions

In this paper we proposed a heuristic approach for CB RS based on feature weighting and LSH with the aim to promote recommendation diversity and make similarity

calculations efficient. The proposed methodology creates a textual representation for each item based on its features. This is used in order to create a weighted representation of the set of items based on TF-IDF scheme. The importance of specific features is adjusted according to our needs by tuning the corresponding weights. Furthermore, the methodology incorporates the weighted Minhash algorithm in order to create a compressed representation of each item. Finally, the Minhash signatures are used by LSH forest in order to make the similarity calculations efficient and produce the recommendations.

Based on the evaluation of our proposed RS and the results that were produced by conducting several experiments, interesting conclusions were drawn. Regarding the recommendation diversity, results show that the characteristics of the produced recommendations can easily be adjusted to the desired results by tuning the product feature weights appropriately. The methodology incorporates recommendation diversity without compromising similarity. This aspect turns out to be very important because most of the CB approaches lack diversity [23]. Considering the product brand as less important in the similarity calculations, results in recommendations that include products of different brands. Similar results can be achieved by tuning other product features. In that way, customers see recommendations that are similar to the target product, but not very similar to each other.

Furthermore, our approach incorporates a hashing method that makes the product similarity calculations much faster and efficient than traditional systems. The representation of product profiles as compressed signatures, by applying the Minhash method, turns out to be effective. Finally, the approximation of product similarities by applying the LSH method results in a system that can handle thousands of products efficiently without compromising similarity. In that way, our proposed approach outperforms against others, as it combines recommendation diversity with calculation efficiency.

Regarding the quality of the produced recommendations, results show that in a very high percentage (94-97%) of cases, at least one recommended product matches with one product view that was generated based on real customer actions. Hence, our proposed approach was shown to offer recommendations that could potentially help users move to a similar product page instead of leaving the e-commerce site.

References

1. Schafer, J.B., Konstan, J.A., Riedl, J. Recommender Systems in E-Commerce. In ACM Conference on Electronic Commerce, pp. 158–166 (1999).
2. Karimova, F. A Survey of e-Commerce Recommender Systems. European Scientific Journal. vol 12. No. 34 ISSN:1857- 7881. (2016).
3. Han, E.-H., Karypis, G. Feature-based recommendation system. 14th ACM international conference on Information and knowledge management, pp. 446-452. (2005).
4. Charikar, M.S. Similarity estimation techniques from rounding algorithms. Annual ACM Symposium on Theory of Computing, pp. 380-388. (2002).
5. Lops P., de Gemmis, M., Semeraro, G. Content-based Recommender Systems: State of the Art and Trends. In: Ricci F., Rokach L., Shapira B., Kantor P. (eds) Recommender Systems Handbook. Springer, Boston, MA (2011).

6. Indyk, P., Motwani, R. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. Annual ACM Symposium on Theory of Computing, pp. 604-613. (2000).
7. Gionis, A., Indyk, P., Motwani, R. Similarity Search in High Dimensions via Hashing. 25th International Conference on Very Large Data Bases, pp. 518–529. (1999).
8. Zhang, K., Fan, S., Wang, H. (2018). An Efficient Recommender System Using Locality Sensitive Hashing. 51st Hawaii International Conference on System Sciences, (2018).
9. Aytekin, A.M., Aytekin, T. Real-time recommendation with locality sensitive hashing. J Intell Inf Syst 53, 1–26 (2019).
10. Debnath, S., Ganguly, N., Mitra, P. Feature weighting in content-based recommendation system using social network analysis. 17th international conference on World Wide Web, pp. 1041-1042. (2008).
11. Barranco, M.J., Martínez, L. A Method for Weighting Multi-valued Features in Content-Based Filtering. In: García-Pedrajas N., Herrera F., Fyfe C., Benítez J.M., Ali M. (eds) Trends in Applied Intelligent Systems. IEA/AIE 2010. Lecture Notes in Computer Science, vol 6098. Springer, Berlin, Heidelberg (2010).
12. Bestprice Homepage, https://www.bestprice.gr/, last accessed 2020/02/21.
13. Ramos, J. Using TF-IDF to determine word relevance in document queries. Department of Computer Science, Rutgers University (2003).
14. Retailrocket dataset, https://www.kaggle.com/retailrocket/ecommerce-dataset, last accessed 2020/02/21.
15. scikit-learn, Machine Learning in Python, scikit-learn.org
16. Wu, W., Li, B., Chen, L., Gao, J., Zhang, C. A Review for Weighted Minhash Algorithms. (2018).
17. Wu, W., Li, B., Chen, L., Zhang, C. Consistent Weighted Sampling Made More Practical. 26th International Conference on World Wide Web, pp. 1035-1043. (2017).
18. Manasse, M. & Mcsherry, F. & Talwar, K. (2007). Consistent weighted sampling. Technical Report. (2010).
19. Ioffe, S. Improved Consistent Sampling, Weighted Minhash and L1 Sketching. IEEE International Conference on Data Mining, pp. 246-255. (2010).
20. datasketch 1.5.0, http://ekzhu.com/datasketch/, last accessed 2020/02/21.
21. Rajaraman, A., Leskovec, J., Ullman, J. Mining of Massive Datasets. Cambridge University Press, USA (2014).
22. Bawa, M., Condie, T., Ganesan, P. LSH forest: Self-tuning indexes for similarity search. 14th international conference on World Wide Web, pp. 651-660. (2005).
23. Bradley, K. Improving Recommendation Diversity. National Conference in Artificial Intelligence and Cognitive Science, Maynooth, Ireland, pp. 75–84. (2001).
24. Smyth, B., McClave, P. Similarity vs. Diversity. In: Aha D.W., Watson I. (eds) Case-Based Reasoning Research and Development. ICCBR 2001. Lecture Notes in Computer Science, vol 2080. Springer, Berlin, Heidelberg (2001).
25. Gerogiannis, V., Karageorgos, A., Liu, L., Tjortjis, C. Personalised Fuzzy Recommendation for High Involvement Products. IEEE SMC, pp. 4884-4890. (2013).
26. Nalmpantis, O., Tjortjis, C. The 50/50 Recommender: A Method Incorporating Personality into Movie Recommender Systems. 8th EANN, pp. 498–507. (2017)
27. Schoinas, I., Tjortjis, C. MuSIF: A Product Recommendation System Based on Multi-source Implicit Feedback. 15th AIAI, pp. 660–672. (2019)