Context-Based and Collaboration-Based Product Recommendation Approaches for a Clothes Online Sale System

Hai Thanh Nguyen, Vi Hung Ngo, and Tran Thanh Dien

Abstract E-commerce systems have developed remarkably and provided a considerable profit for commercial companies and groups. Customers also benefit from such systems. However, the rapidly increasing volume and complexity of data lead customers to find that it is a challenge to find suitable products for their interests. Numerous product recommended methods have been researched and developed to support users when they visit E-commerce websites. This study proposes a recommendation system for a Clothes Online Sale system based on analyzing context-based and collaboration-based methods. Each type was divided into memory-based and model-based approaches. The results give the same product, but the cosine distance of the Word2vec + IDF algorithm is the lowest. We have also deployed algorithms including the K-nearest neighbor's algorithm (KNN), singular value decomposition (SVD), non-negative matrix factorization (NMF), and matrix factorization (MF) for the comparison. The method is evaluated on Amazon women's clothing, including 50,046 samples and six features. We proposed a content-based memory-based method using Word2vec + IDF and a collaboration-based model-based method using the SVD algorithm with the result of RSME as 1.268 to deploy on the sales system.

Keywords Recommendation system · Content-based · Amazon · Collaboration-based · E-commerce website · Clothes

H. T. Nguyen (✉) · V. H. Ngo · T. T. Dien
College of Information and Communication Technology, Can Tho University, Can Tho 900000, Vietnam
e-mail: nthai.cit@ctu.edu.vn

T. T. Dien
e-mail: thanhdien@ctu.edu.vn

T. D. L. Nguyen and J. Lu (eds.), *Machine Learning and Mechanics Based Soft Computing Applications*, Studies in Computational Intelligence 1068,
https://doi.org/10.1007/978-981-19-6450-3_6

1 Introduction

E-commerce websites have been trendy and are growing day by day. E-commerce site managers always want to increase sales while customers want convenience when buying, so the recommendation system is developed and increasingly optimized for E-commerce sites. Thence, we consolidate studies and analyze key types of recommendations, such as content-based and collaboration-based recommendations. Our goal is to synthesize system methods that recommend appropriate products to customers using popular algorithms so that E-commerce site administrators can easily select and implement a recommendation system for their website. In addition, we have compared different algorithms in the same product recommendation method to help people make good choices.

This study introduces a content-based memory-based method using Word2vec + IDF and a collaboration-based model-based method with the SVD algorithm and obtains an RSME of 1.268. In the next sections of the paper, we present some research related in Sect. 2. Section 3 includes our steps to perform the recommendations for an E-commerce website. We also exhibit the results and compare the performance of various methods in Sect. 4. Finally, Sect. 5 is the conclusion and direction of development.

2 Related Work

Many different types of recommendations have been studied [1] as well as the development of various algorithms within those types of recommendations [2, 3]. Moreover, every kind of recommendation has its unique method and evaluation [4], and it develops in many directions, such as building the framework of recommendation system [5], perennial customer information [6]. Typically, the recommended system will be divided using data such as content-based [7] and collaboration-based [8], or even a combination of the two [9].

2.1 Related Work on Text Vectorization

The authors in [10] studied to convert text to vector using TF and IDF, and both give good performance. However, the formula seems simple. Later on, many other methods appear to compare efficiency. The authors in [11] studied the quality of the vector representations of words derived by different models over a set of syntactic and semantic linguistic tasks. They observed that it is possible to train high-quality magnetic vectors using a simple model architecture compared to standard neural network models (straight and repetitive).

2.2 Related Work on the Algorithm for Matrix Factorization

We combine the ability to predict supplementary information and the ability to simulate nonlinear relationships of a multi-layered neural network in the problem of predicting user ratings (explicit) by using the architecture based on Neural Collaborative Filtering and the equation in the article "Neural Collaborative Filtering" [12]. The authors in [13] gave a non-negative matrix factorization(NMF) to solve the problems. We see this as an applicable method for the situation we are studying. Again mentioned the method of improving the accuracy, dividing the matrix into many parts, so the Singular Value Decomposition (SVD) method appears in an article [14]. The question here is whether these methods, which are more efficient when integrating into a particular sales system, are the startup sales system we have built to study.

3 The Proposed Method

Our recommendation system is divided into a context-based method that uses data about each user's context and behavior and the collaboration-based method to use another user's behavior data as the primary user. Each category evolved in two directions: memory-based (memory-based is the use of direct, untrained data) and model-based (model-based is the use of trained data). These four types all use the following architectures:

In Figure 1, memory-based recommendations will stop at the algorithm selection process and then move directly to the top-N product process.

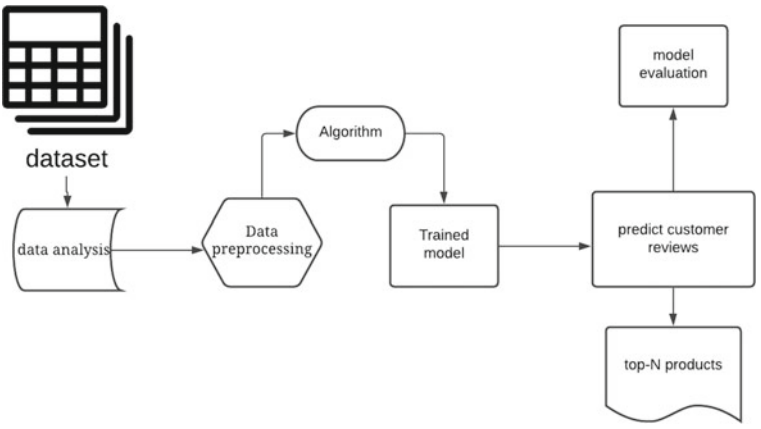

Fig. 1 Processes in the recommended system study

3.1 Data Description

Our method is evaluated on two datasets from the website Kaggle [15] with the following description. The first dataset includes 28,395 samples and seven features on products. The set of considered features contains data on Amazon's standard identification number, title the title of the product, the brand, color information of apparel, and the type of the apparel, the URL of the image (for small image) to show the product image, and the URL of the large image. The other is about product evaluation with 50,046 samples and six features, a collection of customer reviews for a product. The second dataset contains 6584 products and 6670 customers. Features include: ASIN (Amazon standard identification number), title (title of the product), review_userId (identification number of reviewer), review_score (rating score), review_summary (summary comment section), review_text (comment section content). Our data analysis mainly looks at the top-rated products or the most appreciated products, the statistics of the features, and customer opinion through product reviews.

3.2 Data Preprocessing

In dealing with **loss data**, we removed rows because there are many unique types and a little bit of data loss.

Remove **duplicate title data**: Because we use the title to suggest products with similar titles in the contextual suggestion, we have to deal with each word in the title duplicate. Therefore, the data will be duplicated as follows: Products with almost the same title only differ in size, and products with nearly the same title vary only by color.

Remove Stop Words in title data: Stop Words are some words partially or completely ignored by search engines. Words like: "a, a, a, of, or, many, etc... ". These words have nothing to do with the content and meaning of the article.

Handling special characters such as.,?! -_ + \ [] { } " ;: '= /: We will proceed to remove all these special characters. Because these special characters affect the comparison of the filter.

Text vectorization: We use different equations to find the best one**,** those equations include **IDF** (Inverse Document Frequency), **TF-IDF** (Term Frequency-Inverse Document Frequency).

Word2vec [11] is an unsupervised learning model trained from a large corpus. The dimension of Word2vec is much smaller than one-hot encoding, where the dimension is $N \times D$, where N is the number of documents, and D is the number of word embedding. Word2vec represents each separate word with a list of specific numbers called vectors.

We deploy Word2vec and IDF, as shown in Eq. 1. IDF calculates the titles of the product to find the weight of each word. When using the Word2vec model, synonyms will multiply by the IDF weight of each corresponding word.

3.3 Algorithm and Trained Model and Predict Customer Reviews

In the case of memory-based recommendations, we skip the model training and move straight to the customer rating prediction process or the top-N product. In memory-based context-based recommendations, we used cosine distance to calculate the distance of the vector of products other than the vector of the proposed product. Evaluating the two vectors by "cosine distance" has the following equation [16]:

$$\text{cosine distance} = 1 - \cos(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{AB}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} \mathbf{A}_i \mathbf{B}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{A}_i)^2}\sqrt{\sum_{i=1}^{n} (\mathbf{B}_i)^2}} \quad (1)$$

In Eq. 1, A_i and B_i are components of vectors A and B, respectively. After applying Eq. 1, we have sorted the list of products according to the products with the smallest distance, and we have the list of top N similar products. While calculating the cosine distance, we can calculate other product features such as brand, product_type_name, and color by converting the vector as shown in the Bag of Words section and blending it. Then, we get a different characteristic spacing and title characteristic spacing.

$$\text{dist} = (W_1 * \text{titleDist} + w_2 * \text{extraDist})/(w_1 + w_2) \quad (2)$$

In Eq. 2, w_2 is the weight of title, w_2 is the weight of extra features, titleDist is title difference measured by distance, extraDist is additional features difference measured by the distance, and dist is the combination of these two parameters.

In memory-based collaboration-based recommendations [17], we also studied and built a user-item matrix, then calculated the correlation between users to form the human correlation matrix. User is referred to as user-based. Similar to products, we calculate the correlation between products to form a product correlation matrix called item-based. The correlation equation is as follows:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}} \quad (3)$$

In Eq. 3, r = correlation coefficient, x_i = value of the variable x in the sample, $\overline{x}$ = mean value of variable x, y_i = value of the variable y in the sample, $\overline{y}$ = mean value of variable y. After calculating the similarity between users or between products, we can predict the rating of user u on the product i using the following formula:

$$\widehat{r_{ui}} = \overline{r_u} + \frac{\sum_{u' \in K_u} \text{sim}(u, u').(r_{u'i} - \overline{r_{u'}})}{\sum_{u' \in K_u} |\text{sim}(u, u')|} \quad (4)$$

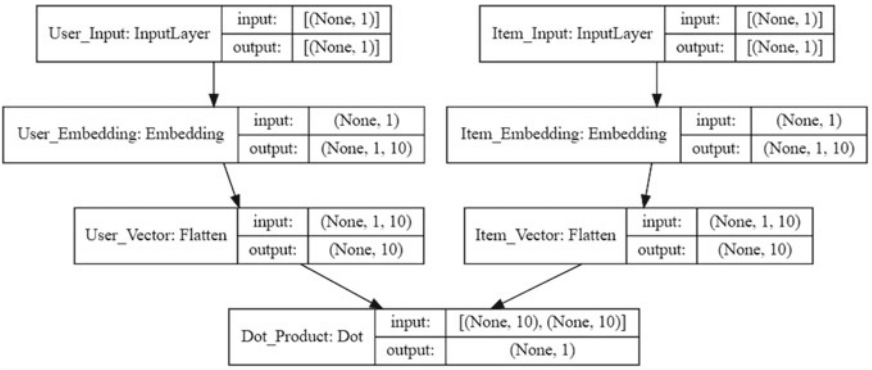

Fig. 2 Deep learning architecture 1

In Eq. 4, $\widehat{r_{ui}}$ is the prediction for user u on product i, sim(u, u') similarity between user u and u', K_u is the number of users whose proximity is near user u.

In model-based collaboration-based recommendations, we divided the sample data set into two parts to train and evaluate the model. We used algorithms: K-nearest neighbor (KNN), SVD, and NMF. The algorithms NMF and SVD have same the following equation:

$$\widehat{r_{ui}} = \mu + b_u + b_i + {q_i}^T p_u \tag{5}$$

In Eq. 5, $\widehat{r_{ui}}$ is the score which the customer u like the product i, while μ is the global average of all user ratings across all products. b_i is the deviation of the product (mean value of products relative to the global mean), and b_u is the user's deviation (the average of the users close to the global mean). If customer u is undefined, b_u is expected, and the factors p_u are assumed to be 0. The same applies to product i with b_i and q_i. To expand our research, we have referenced article [12] and proposed two more deep learning types for this problem. The architecture of those two deep learning types is as follows:

Figure 2 shows this architecture mainly after embedding the input layer and then calculating those two vectors' scalar product. The dot layer is the layer used to combine two layers of user input and product input by calculating the scalar product corresponding to each user of the embedded layer and the embedded layer's product.

In Fig. 3, we did not use the scalar product, we used the join layer to join, and the dense layer called the sigmoid function to compute the input vector. The output at each layer is equal to the product of the input vector with the weight matrix, and each element of the output vector is applied the nonlinear operator σ

$$y = \sigma(wx) \tag{6}$$

σ is a space matrix $\mathbb{R}_{m_x n}$ whose elements are applied nonlinear operator σ. The activation function σ of the classes is a **sigmoid function**.

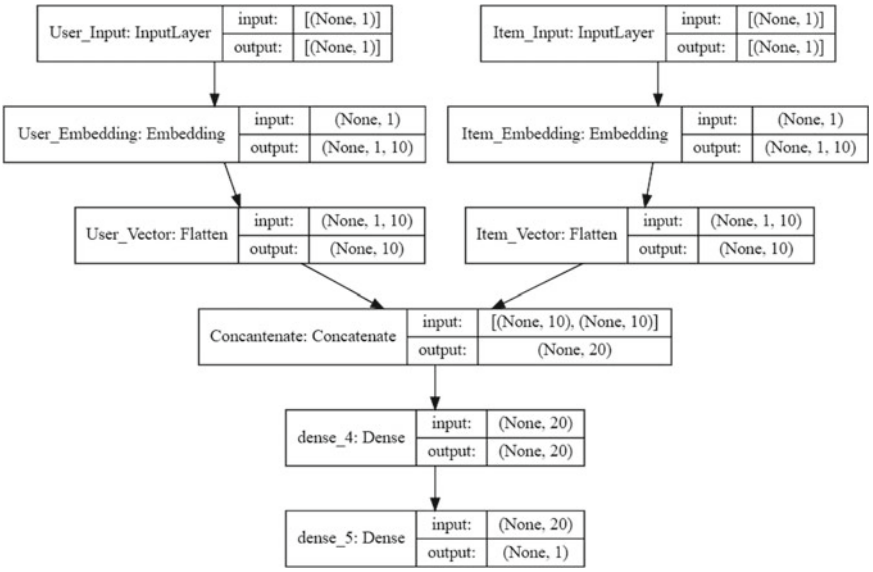

Fig. 3 Deep learning architecture 2

3.4 Model Evaluation

Evaluate the model by two main measures: RMSE and MAE. The mean squared error (MSE) of an estimate is the mean of the error square. In other words, it is the difference between the values predicted by the model and the actual value. We take the root of MSE to get RMSE. The mean absolute error (MAE) is a method of measuring the difference between two continuous variables. The MAE is calculated as follows:

$$\text{MAE} = (\frac{1}{n}) \sum_{i=1}^{n} \left| Y_i - \widehat{Y}_i \right| \tag{7}$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \Sigma_{i=1}^{n} \left(Y_i - \widehat{Y}_i \right)^2} \tag{8}$$

In Eqs. 7 and 8, n is the number of elements in the set, Y_i is worth real evaluation, $\widehat{Y}_i$ is the value of the prediction.

4 Experimental Results

The main presentation compares the algorithms and then compares the recommended methods.

4.1 Text Vectorization Methods

The process steps in this form are mainly about calculating the cosine distance of the products other than the input product, and we will reorder which products have the smallest cosine distance Eq. 1 relative to the input product, which is also a list of similar products. We test to find similar products with the title "*women's unique 100 kinds of cotton special Olympics world games 2015 white size L*" for all three algorithms Bag of Words, TF-IDF, IDF. All three algorithms return the same products "*women's new 100 kinds of cotton special Olympics world games 2015 red size*" but different in distance calculation. Table 1 shows that the most similar product results are the same but different indicators in that the cosine distance using Word2vec+IDF is somewhat smaller.

Table 1 Cosine distance results on the same product

	Womens new 100 cotton special olympics world games 2015 red size
Bag of Words	0.18181818181818166
TF-IDF	0.12748023262076247
IDF	0.11689768372442932
Word2vec	0.05564171
Word2vec+IDF	0.03939116

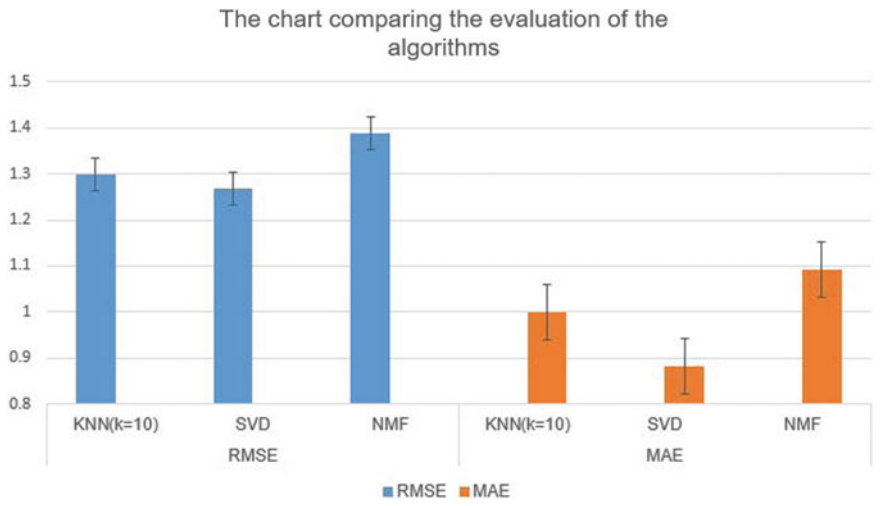

Fig. 4 Average performance comparison KNN, SVD, NMF in RMSE and MAE. The standard deviations are shown in error bars

4.2 Recommended Methods

We can use the machine learning model in the content-based recommended method, but because we comply with the method's rules based on each user's data, the data is greatly lacking. Besides, many users only rate very few products, this user data is very little, so when training the model, the results are not satisfactory. However, we still research to recognize the weaknesses and strengths of this method. We will present it in the comparison of methods. In the collaboration-based recommended method, we compare three algorithms to train the model: **KNN, SVD, and NMF** with fivefold cross-validation on the dataset.

From Fig. 4, we see that the SVD algorithm is the best algorithm for RMSE and MAE results, but we have to pass the corresponding parameter to get this result. We will choose the RMSE measure for SVD comparison with two deep learning structures, all three algorithms divide the data sample set (50,046 samples and 3 features) with a training rate of 0.8, a test rate of 0.2, and the number of epoch is 50, N_factor(the number of feature vectors) = 10.

Table 2 shows that architecture 2 of deep learning has a smaller measure than SVD and proves that the algorithm is better. The standard is to choose the right recommendation method for a newly opened sales system to split deals by new product, new user, both new product and new user, to choose the method that could predict in the case. In Table 3, we can see that in the context-based approach, the memory usage method works well in case the sales system is just starting up. In collaboration-based methods, we see that using memory to calculate the correlation takes time. Using the deep learning model is unpredictable due to the use of embedded

Table 2 Measure comparison between SVD and deep learning algorithms

	SVD	Deep learning architecture 1	Deep learning architecture 2
RMSE	1.2680594415447108	2.1875152570896508	1.2665433273283626

Table 3 Method for new product, new user

		New product	New user	Both new product and new user
Content-based	Memory	Still works	Still works	Still works
	Model	New data training is required	New data training is required	New data training is required
Collaborate-based	Memory	Still works time-consuming	Still works time-consuming	Time-consuming to rebuild corr-matrix
	Model	Still works but not with deep learning	Still works but not with deep learning	Still works but not with deep learning

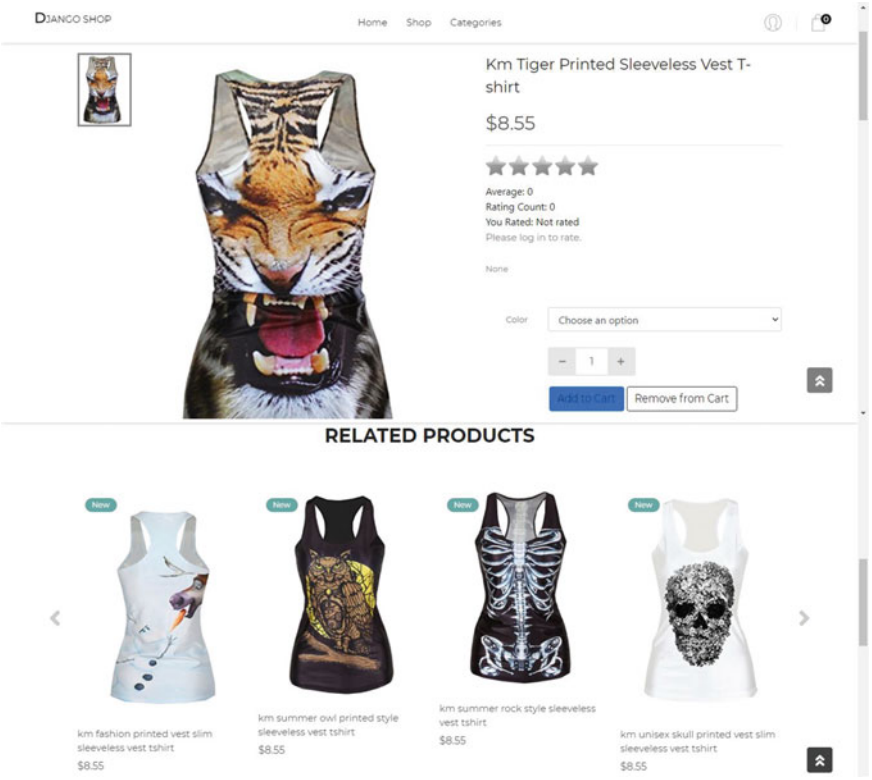

Fig. 5 Recommended product interface when deployed on a sales system

layers, but solving SVD techniques is still predictable. Finally, we implemented the referral system into the sales web system. Figure 5 shows the input product interface and shows the proposed output product interface (similar to the input product).

5 Conclusion

Regular E-commerce sites usually have basic information about the product, so in this study, we propose a method based on content, using memory, using Word2vec + IDF algorithm for text vectorization to find similar products through the product's contextual information because of the lowest cosine distance (0.03939116). We propose a model-based, collaborative-based approach that uses the SVD algorithm for the launch sales system to predict ranking scores. SVD model is predictable, with RMSE being 1.2681 for new users and new products, so it is suitable for E-commerce startups and E-commerce sites with big data. In the future, we should apply the A/B testing method (also known as split testing) to compare according to customers'

feelings to increase convenience for customers. A/B testing is a process in which two versions (A and B) are compared together in a defined environment/situation, thereby evaluating which version is more efficient.

References

1. Thu, T. N. M., & Hien, P. X. (2016). Evaluation methods for recommender systems (in Vietnamese). *Can Tho University Journal of Science., 42,* 18–27. https://doi.org/10.22144/ctu.jvn.2016.023.
2. Bao, L. H. Q., Dat, Q. N. , & Nghe, N. T. (2015). Model of combination of fisheries and the law of the number of east for ranking in the recommendation system (in Vietnamese). *Can Tho University Journal of Science IT*, 1–8.
3. Dung, N. H., & Nghe, N. T. (2014). Product recommendation system in online sales using collaborative filtering techniques (in Vietnamese). *Can Tho University Journal of Science, 31*, 36–51.
4. Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. *Recommendation systems: Principles, methods and evaluation*. https://www.sciencedirect.com/science/article/pii/S1110866515000341b0175.
5. Abumalloh, R. A., Ibrahim, O., & Nilashi, M. (2020). Loyalty of young female Arabic customers towards recommendation agents: A new model for B2C E-commerce. *Technology in Society, 61*, 101253. https://doi.org/10.1016/j.techsoc.2020.101253
6. Tareq, S. U., Habibullah Noor, Md., & Bepery, C. (2019). The framework of dynamic recommendation system for e-shopping. *International Journal Information Technology, 12*, 135–140. https://doi.org/10.1007/s41870-019-00388-6
7. Beleveslis, D., & Tjortjis, C. (2020) Promoting diversity in content based recommendation using feature weighting and LSH. In: I. Maglogiannis, L. Iliadis, & E. Pimenidis (Eds.), *Artificial intelligence applications and innovations. AIAI 2020. IFIP Advances in information and communication technology* (Vol. 583). Springer. https://doi.org/10.1007/978-3-030-49161-1_38
8. Singh, M. K., & Rishi, O. M. (2020). Event driven recommendation system for E-commerce using Knowledge based collaborative filtering technique. *Scalable Computing: Practice and Experience, 21*(3), 369–378. https://doi.org/10.12694:/scpe.v21i3.1709, ISSN 1895-1767
9. Cai, X., Hu, Z., Zhao, P., Zhang, W., & Chen, J. (2020). A hybrid recommendation system with many objective evolutionary algorithm. *Expert Systems with Applications, 159*, 113648. https://doi.org/10.1016/j.eswa.2020.113648
10. Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28*(1), 11–21.
11. Mikolov, T., et al. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
12. He, X., Liao, L., & Zhang, H. (2017). Neural collaborative filtering. In *WWW '17: Proceedings of the 26th International Conference on World Wide Web* April 2017, pp. 173–182. https://doi.org/10.1145/3038912.3052569
13. Hernando, A., & Bobadilla, J. (2016). FernandoOrtega-A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. *Knowledge-Based Systems, 97*(1), 188–202.
14. Li, C., & Yang, C. (2016). The research based on the Matrix Factorization recommendation algorithms. In *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*.
15. Amazon (May 1996–July 2014). Apparels review. https://www.kaggle.com/thekenjin/amazonapparelsdata

16. Wen, H., Ding, G., Liu, C., & Wang, J. (2014). Matrix factorization meets cosine similarity: Addressing sparsity problem in collaborative filtering recommender system. In: L. Chen, Y. Jia, T. Sellis, & G. Liu, (Eds.), *Web technologies and applications* (Vol. 8709). APWeb 2014. Lecture Notes in Computer Science. Springer. https://doi.org/10.1007/978-3-319-11116-2_27
17. Stephen, S. C., Xie, H., & Rai, S. (2017). Measures of similarity in memory-based collaborative filtering recommender system: A comparison. In *Proceedings of the 4th Multidisciplinary International Social Networks Conference (MISNC '17). Association for Computing Machinery*, New York, USA, Article 32, 1–8. https://doi.org/10.1145/3092090.3092105