Collaborative Recommendation with Energy Distance Correlation

Mun Van Dong[1], Trong Van Nguyen[2], Nhung Cam Thi Mai[3], Tu Cam Thi Tran[4], and Hiep Xuan Huynh[3](✉)

[1] Hau Giang Public Administration Service Center, Vi Thanh, Hau Giang, Vietnam
[2] Bac Lieu University, Bac Lieu, Vietnam
nvtrong@blu.edu.vn
[3] Can Tho University, Can Tho, Vietnam
{mtcnhung,hxhiep}@ctu.edu.vn
[4] Vinh Long University of Technology Education, Vinh Long, Vietnam
tuttc@vlute.edu.vn

Abstract. The recommendation systems are applied to many fields of the social life. In which, the measure of the similarity, and the measure of the distance are the core problems of the recommender systems, there are many proposals with the different approaches, it shows the characteristics of each recommendation system, commonly used measures such as: the measure Cosine, the measure Pearson, the measure Jaccard, etc. However, there have not been many studies on the energy dependence to determine the correlation of the objects in the process of building a recommendation system. In this article, we mainly focus on determining the correlation/compatibility of the energy-based objects in building a recommendation model. The experimental results are evaluated on two datasets, that are MSWeb datasets and Learning from Sets of Items 2019 datasets, the results show that the proposed model has higher accuracy than the traditional model.

Keywords: Energy distance · Energy dependence measures · Collaborative filtering · Recommendation system · Distance correlation

1 Introduction

With the continuous development of the science and technology, the amount of the information in each field has increased, so the exploitation and using of the information has been studied and they are applied by many scientists in the practice. In particular, the recommendation system [1–4] uses data based on user's feedback about the items, based on the similarity of product characteristics and based on knowledge forms. The different methods to give suggestions for the users according to the level of interest from high to low.

There are many different proposed methods to solve the recommendation problem, the choice of method, it depends on the type of the information, the learning model, predicting new products for the users [5, 6]. Basically, there are main groups/formations

C. V. Phan and T. D. Nguyen (Eds.): ICCASA 2022, LNICST 475, pp. 19–32, 2023.
https://doi.org/10.1007/978-3-031-28816-6_2

of the recommendation model that are divided as follows: Content-Based Recommender Systems [2, 4, 7, [8]; Collaborative Filtering Systems [2, 4, 7, 9]; Knowledge-Based Recommender Systems [4, 7]; Context-Based Recommender Systems [4, 7]; Hybrid Recommender System [4, 7, 10]. The most approaches to building recommendation systems today are based on a measure of similarity such as: cosine, Pearson, Jaccard, etc. Each model has its own advantages in using the measures, it is suitable for the type of information applied in the model, so there are many recommendation models that have been successfully applied in the various fields, especially in the field of e-commerce [11, 12].

In addition to the traditional similarity measures, the energy distance [13] is a measure used to determine the distance correlation between the vectors (the vectors have arbitrary size), which is a powerful tool in multivariate analysis, opens a new direction in building the recommendation systems. Currently, there are not many studies on applying this measure in building a recommendation system model. In this article, we present a collaborative filtering model that predicts for the users who are missing ratings at specific products and recommends the best products (the most relevant products) based on the distance correlation measures/energy distances [14].

The structure of the article is divided into 5 parts: Sect. 1 introduces the basic issues in the recommender system and the proposes energy measures in building a recommendation model, Part 2 presents an overview of the collaborative filtering model, Part 3 proposes a collaborative filtering model with the distance correlation, Part 4 presents the experimental results and the evaluations, and finally a conclusion presents a summary of the obtained results.

2 Collaborative Filtering

Collaborative filtering [2, 7] uses the user's ratings dataset to rate products liked by users, so it makes the missing rating predictions for the items (the unreviewed product) or recommend a specific number of products considered best to a user who needs recommendations.

Given a list of m users $U = \{u_0, u_1, \ldots, u_m\}$ and a list of n items $I = \{i_0, i_1, \ldots, i_n\}$. Where $R = r_{jk}$ is the rating stored in the user rating matrix m x n where each row represents user u_j (with $1 \leq j \leq m$)) and each column represents a item i_k (with $1 \leq k \leq n$). R_{jk} represents the rating of the user u_j for the item i_k, all is shown in Table 1.

Let $u_a \in U$ be the user who needs to be suggested or the active users, and $I_a = I \backslash \{i_l \in I | r_{al} = 1\}$ is the set of the items unknown to user u_a.

The task is to predict the ratings for all items I_a or create a list (top N) of the best recommendations for u_a. The missing ranked values are predicted on each row of matrix r_a, where the missing values are estimated from other data in R, on the basis of ratings for all unknown items I_a, select the N highest predictions in ranking order.

For example, a set of 7 users $U = \{u_0, u_1, u_2, u_3, u_4, u_5, u_6\}$ and 5 items $I = \{i_0, i_1, i_2, i_3, i_4\}$. Each user gives their rating of the products on a rating scale of {?, 1, 2, 3, 4, 5}. Table 2 represents the user's rating matrix with items, where the intersection of the user (column) and item (row) is the user's rating value corresponding to that item, the cells represent "?" are products that have not been rated by users.

Table 1. The general matrix of users and items.

	$\mathbf{i_0}$	$\mathbf{i_1}$	$\mathbf{i_{...}}$	$\mathbf{i_{n-1}}$	$\mathbf{i_n}$
$\mathbf{u_0}$	$r_{0,0}$	$r_{0,1}$	$r_{0,.}$	$r_{0,n-1}$	$r_{0,n}$
$\mathbf{u_1}$	$r_{1,0}$	$r_{1,1}$	$r_{1,.}$	$r_{1,n-1}$	$r_{1,n}$
$\mathbf{u_{...}}$	$r_{..,0}$	$r_{...,1}$	$r_{...,.}$	$r_{..,n-1}$	$r_{..,n}$
$\mathbf{u_{m-1}}$	$r_{m-1,0}$	$r_{m-1,1}$	$r_{m-1,.}$	$r_{m-1,n-1}$	$r_{m-1,n}$
$\mathbf{u_m}$	$r_{m,0}$	$r_{m,1}$	$r_{m,.}$	$r_{m,n-1}$	$r_{m,n}$

Table 2. The rating matrix for users and items.

	$\mathbf{i_0}$	$\mathbf{i_1}$	$\mathbf{i_2}$	$\mathbf{i_3}$	$\mathbf{i_4}$
$\mathbf{u_0}$	5	4	?	2	2
$\mathbf{u_1}$	5	?	4	2	0
$\mathbf{u_2}$	2	?	1	3	4
$\mathbf{u_3}$	0	0	?	4	?
$\mathbf{u_4}$	1	?	?	4	?
$\mathbf{u_5}$	?	2	1	?	?
$\mathbf{u_6}$	?	?	1	4	5

To predict missing ratings, the system recommends using distance correlation to calculate the values. The way to determine the value is to predict each pair of users against each other, for example, if we want to predict the rating value for product i_2 by user u_0 ($u_{0,2}$), we will calculate the correlation between the user pair {$u_{0,1}$; $u_{0,2}$; $u_{0,3}$; $u_{0,4}$; $u_{0,5}$; $u_{0,6}$}, selecting the best compatible value to determine the prediction for user u_0 at item i_2 (Table 3).

Table 3. The matrix predicts the ratings.

	$\mathbf{i_0}$	$\mathbf{i_1}$	$\mathbf{i_2}$	$\mathbf{i_3}$	$\mathbf{i_4}$
$\mathbf{u_0}$	5	4	3.31	2	2
$\mathbf{u_1}$	5	3.04	4	2	0
$\mathbf{u_2}$	2	3.05	1	3	4
$\mathbf{u_3}$	0	0	0.38	4	0.8
$\mathbf{u_4}$	1	1.31	2.36	4	2.64
$\mathbf{u_5}$	0.87	2	1	2.35	2.35
$\mathbf{u_6}$	4.18	3.75	1	4	5

However, in practice we don't need to predict all the rating values for a user, we just find the most suitable item to suggest to that user or suggest the user to match the item, that is called nearest neighbor (Table 4).

Table 4. The correlation of user pairs.

$\mathbf{u_i}$	$\mathbf{u_0}$
$\mathbf{u_1}$	0.71
$\mathbf{u_2}$	0.24
$\mathbf{u_3}$	0.94
$\mathbf{u_4}$	0.86
$\mathbf{u_5}$	0.38
$\mathbf{u_6}$	0.12

Assuming u_0 is the user to suggest, we will determine the similarity of user u_0 with previous users, choose $k = 2$ then the nearest neighbor of u_0 $\{u_3, u_4\}$.

3 Recommendation with Distance Correlation

3.1 Distance Covariance

Distance covariance measures the dependence between random vectors with an arbitrary dimension, these dimensions are not necessarily equal [15, 17] and they are adjustable [16], specifically:

Assume there are the random observation vector samples $(X_i, Y_i) \in R$ (i = 1, 2, ….n).

Convention:

$$a_{ij} = \left|X_i - X_j\right| (i, j = 1, 2, \ldots, n)$$

$$a_{i.} = \sum\nolimits_{k=1}^{n} a_{ik} \quad a_{.j} = \sum\nolimits_{k=1}^{n} a_{kj} \quad \bar{a}_i = \bar{a}_{i.} = \frac{1}{n} a_{i.}$$

$$a_{..} = \sum\nolimits_{i,j=1}^{n} a_{ij} \quad \bar{a} = \frac{1}{n^2} \sum\nolimits_{i,j=1}^{n} a_{ij}$$

$$b_{ij} = \left|Y_i - Y_j\right| (i, j = 1, 2, \ldots, n)$$

$$b_{i.} = \sum\nolimits_{k=1}^{n} b_{ik} \quad b_{.j} = \sum\nolimits_{k=1}^{n} b_{kj}$$

$$b_{..} = \sum\nolimits_{i,j=1}^{n} b_{ij} \quad \bar{b} = \frac{1}{n^2} \sum\nolimits_{i,j=1}^{n} b_{ij}$$

$$A_{i,j} = a_{ij} - \overline{\mathrm{a}}_i - \overline{\mathrm{a}}_j + \overline{\mathrm{a}} \quad B_{i,j} = b_{ij} - \overline{\mathrm{b}}_i - \overline{\mathrm{b}}_j + \overline{\mathrm{b}}$$

$$\boldsymbol{A}^*_{i,j} = \begin{cases} \frac{n}{n-1}\left(\mathbf{A}_{i,j} - \frac{\mathbf{a}_{ij}}{n}\right), & i \neq j \\ \frac{n}{n-1}\left(\overline{a}_i - \overline{a}\right), & i = j \end{cases}$$

$$\boldsymbol{B}^*_{i,j} = \begin{cases} \frac{n}{n-1}\left(\mathbf{B}_{i,j} - \frac{\mathbf{b}_{ij}}{n}\right), & i \neq j \\ \frac{n}{n-1}\left(\overline{b}_i - \overline{b}\right), & i = j \end{cases}$$

Then, the modified distance covariance statistical formula is [16]:

$$v^*_n(\mathrm{X}, \mathrm{Y}) = \frac{1}{n(n-3)}\left\{\sum\nolimits_{i,j=1}^{n} A^*_{i,j}B^*_{i,j} - \frac{n}{n-2}\sum\nolimits_{i=1}^{n} A^*_{i,i}B^*_{i,i}\right\} \tag{1}$$

With $n \geq 3$.

3.2 Distance Correlation

The statistics on distance correlation dcorT [14] were proposed by G. J. Székely, M. L. Rizzo, and N.K. Bakirov in 2013.

The statistical operation of distance correlation dcorT is formed base on the transformed parameter t of dCor [14] of G.J. Szekely, M.L. Rizzo, N.K. Bakirov. The advantage of dcorT checks the independence between vectors (the size of the vector is arbitrary), this approach is applied to replace the traditional analysis.

The adjusted distance covariance statistical formula [16] is:

$$R^*_n(\mathrm{X}, \mathrm{Y}) = \frac{v^*_n(\mathrm{X}, \mathrm{Y})}{\sqrt{v^*_n(\mathrm{X}, \mathrm{X})v^*_n(\mathrm{Y}, \mathrm{Y})}} \tag{2}$$

With:

X, Y are random vectors.

$v^*_n(\mathrm{X}, \mathrm{Y})$, $v^*_n(\mathrm{X}, \mathrm{X})$, $v^*_n(\mathrm{Y}, \mathrm{Y})$ are modified distance covariance statistics.

3.3 Recommendation

The calculation to determine the rating matrix similarity using the distance correlation statistic dcorT [14] is built as follows:

Algorithm1.1. recommendation with distance correlation

Input: User set U, item set I, and the rating matrix of the users U for the items I

New user (ux)

Output: Recommend items $I_{ux}=\{i_1, i_2,\ldots,i_k\}$ for ux users with item by preference level from high to low;

Begin

Step 1: Determine k nearest neighbors of new user u_x

For user $u_i \in U_n$ perform

<Calculate the similarity of each pair of users u_i, u_x using the distance correlation in energy>

<Sort the list of the users in order of similarity from highest to lowest>

<Select a list of k users closest to user u_x>

Step 2: Predictions and recommend the items to the users u_x

Step 3: Evaluation of the proposed model

End.

In this section, we build a collaborative filtering model with user-based energy distance correlation. This is a method of analyzing data that evaluates users' ratings based on the favorite items of many individuals. If two users have the same interests, they will like similar items. Therefore, the first works is to fill in the missing predictions of the users by finding the k users with the most similar preferences (nearest neighbors) and then perform calculations to rank their ratings for these users go to the best most relevant recommendations (Fig. 1).

The determination of the k users with the closest preferences is done using the dcorT statistic [14], which tests the energy distance correlation between the ratings of each user.

This collaborative filtering model is built with input data of two variables: (1) The matrix of the user reviews of the previous products; (2) the user needs to be referred to the item.

3.4 Training and Testing for Model

To evaluate the models, we split the dataset into 2 parts: the training data set called "Train" and test them on some test data called "Test".

In this article, we use the k-fold cross-validation method [22] to divide the dataset to evaluate the recommender model. The k-fold cross-validation method is a method that divides the data into a number of parts (k parts of the same size), performs k evaluations, each time it evaluates the system, it takes one part as the test training set, the remaining

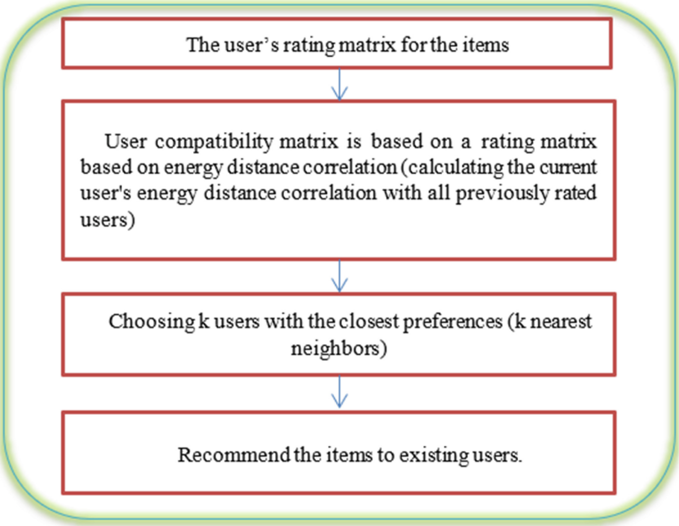

Fig. 1. Collaborative filtering model with distance correlation.

parts (k-1) serve as the training set and we proceed similarly, the result is determined by the average of the evaluations. With this method, make sure each user thing appears in the test suite at least once, then we can measure more accurately.

3.5 Evaluating of the Model

The overall performance of the model is evaluated by several evaluation metrics, we can evaluate the rating based on the confusion matrix and ROC curve (Table 5):

Table 5. Evaluation of the recommendation system.

	Practical class	
Prediction class	True Positive (TP)	False Positive (FP)
	False Negative (FN)	True Negative (TN)

In there:

+True Positive (TP): These are the recommended items that have been purchased.

+False Positive (FP): These are suggested but not purchased items.

+False Negative (FN): These are not recommended items purchased.

+True Negative (TN): These are not recommended items that have not been purchased.

Precision: A measure/index of the model accuracy that is the ratio of correctly suggested items (TP) to total suggested items (TP + FP).

$$\textbf{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FP}} \tag{3}$$

Recall: A measure/index of model accuracy, it is the ratio of correctly suggested items (TP) to the total number of useful suggestions (TP + FN).

$$\textbf{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}} \tag{4}$$

Receiver Operating Characteristic (ROC):

+True Positive Rate (TPR): This is the percentage of purchased items that were recommended. It is the number of TP divided by the number of items purchased (TP + FN).

+False Positive Rate (FPR): This is the percentage of non-purchased items that were recommended. It is FP divided by zero of purchased items (FP + TN).

Area Under the Curve (AUC) to quantify model accuracy based on calculating the area under the curve. The area AUC is the area from the horizontal axis that is bounded by the curve. If the ROC is closer to the left corner, the AUC area will be larger, the accuracy of the model will be higher.

4 Experiment

4.1 Data

The MSWeb [18] and Learning from Sets of Items 2019 [19] datasets are used to evaluated for the propose model.

MSWeb is a sampled dataset of anonymous users visiting www.microsoft.com on a one-week timeframe in February 1998. This dataset is sampled from 32710 anonymous users visiting over 285 original web address. User ratings in this dataset are binaryRatingMatrix.

Learning from Sets of Items 2019 is a user-rated dataset of movies rated on [phimmoi.org] (https://movielens.org/) from February to April 2016 with 45897 ratings of 854 person for 13012 movies, rated movies from 1 to 5, real data type is realRatingMatrix.

4.2 Tool

To implement the experiment, we have updated and added the recommended packages arules_1.7-3 [21] and recommenderlab_1.0.0 [22] in R language.

In addition, we used the recommended package energy_1.7-10 [20] to build a user-based collaborative filtering model, writing a function to measure the distance correlation in RStudio, and we named the function is ebms_energy_dcorT (in R language).

4.3 Scenario 1: Recommendation on Learning from Sets of Items 2019

To compare the performance of the built model, we test two models with three measures: Collaborative filtering model on users with the ebms_energy_dcorT measure we built; Additive filtering model on UBCF users with measures of cosine and pearson respectively on the Learning from Sets of Items 2019 dataset.

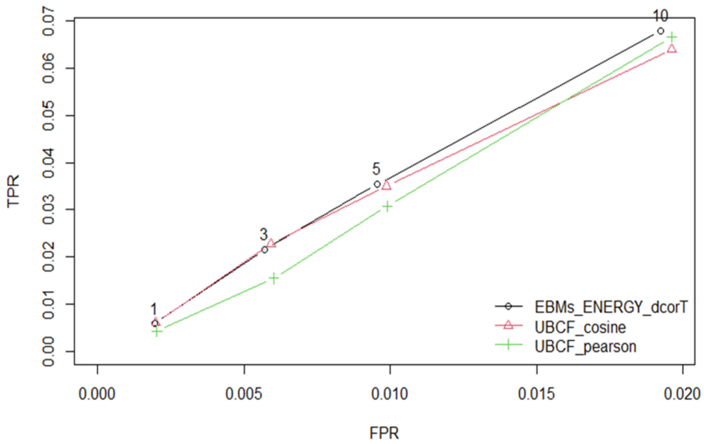

Fig. 2. The graph of ROC curve on Learning from Sets of Items 2019 with 3 measures.

Experimental results in Fig. 2 shows that the predictive accuracy of the model with the ebms_energy_dcorT measure is higher than that of the UBCF model with the cosine and pearson measures, the area under the AUC curve of our proposed model is the largest. The prediction accuracy of UBCF_cosine is the lowest about 0.062, while the prediction accuracy of EBMS_ENERGY_dcorT is the highest, about 0.068 (Fig. 3).

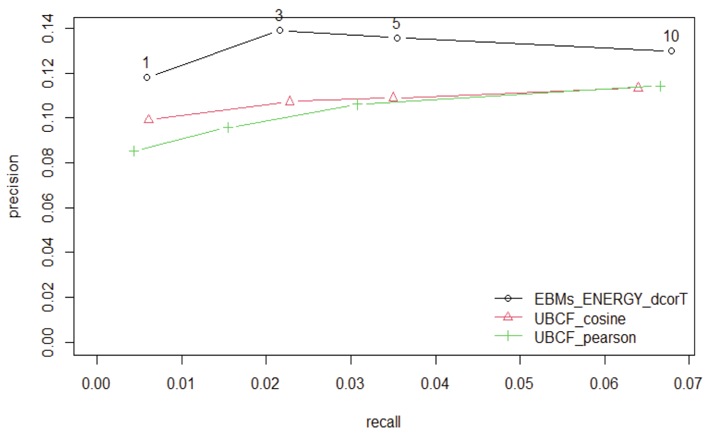

Fig. 3. The recall chart on Learning from Sets of Items 2019 with three measures.

The precision evaluation performance of the model based on the ebms_energy_dcorT tool has a higher rate than the UBCF-cosine and UBCF-pearson models shown in Table 6. With n = 1 precision index of $ The lowest UBCF_pearson (0.08521739) and the highest $EBMs_ENERGY_dcorT (0.1182609). Similar to n = {3, 5, 10}, the Precision of ebms_energy_dcorT is higher than $UBCF_cosine and $UBCF_pearson.

Table 6. Recommendation table with independent bias.

	Precision	Recall	TPR	FPR	n
$EBMs_ENERGY_dcorT	0.1182609	0.005926583	0.005926583	0.001950615	1
$UBCF_cosine	0.09913043	0.006070018	0.006070018	0.001995051	1
$UBCF_pearson	0.08521739	0.004339777	0.004339777	0.002027737	1
$EBMs_ENERGY_dcorT	0.1391304	0.021649753	0.021649753	0.005712064	3
$UBCF_cosine	0.10724638	0.022752706	0.022752706	0.005934277	3
$UBCF_pearson	0.09565217	0.015465308	0.015465308	0.006012859	3
$EBMs_ENERGY_dcorT	0.1356522	0.035455768	0.035455768	0.009555129	5
$UBCF_cosine	0.10886957	0.034982341	0.034982341	0.009866516	5
$UBCF_pearson	0.10608696	0.030807933	0.030807933	0.009899105	5
$EBMs_ENERGY_dcorT	0.1299130	0.067911814	0.067911814	0.019240201	10
$UBCF_cosine	0.11339130	0.063980242	0.063980242	0.019616993	10
$UBCF_pearson	0.11426087	0.066532646	0.066532646	0.019602998	10

4.4 Scenario 2: Recommendation on MSWeb Dataset

Similar to the above, we test two models with three measures: Collaborative filtering model on users with ebms_energy_dcorT measure built by us; Additive filtering model on UBCF users with cosine and pearson measures on MSWeb dataset, respectively.

Experimental results in Fig. 4 show that the predictive accuracy of the model with ebms_energy_dcorT is higher than that of the UBCF model with cosine and pearson measures, the area under the AUC curve of our proposed model. is the largest. Similar to scenario 1, in this scenario Table 6, the prediction accuracy of UBCF_cosine is the lowest about 0.08, while the prediction accuracy of EBMS_ENERGY_dcorT is the highest, about 0.28 (Fig. 5).

According to Table 7, it has once again confirmed the accuracy of the EBMs_ENERGY_dcorT model with n = {1, 3, 5, 10} respectively, which has a higher rate than the UBCF models- cosine and UBCF-pearson. This indicator is shown on the highest deviation at n = 5, $UBCF_cosine has the lowest Precision (0.08163934), while $EBMs_ENERGY_dcorT has the highest (0.4550820).

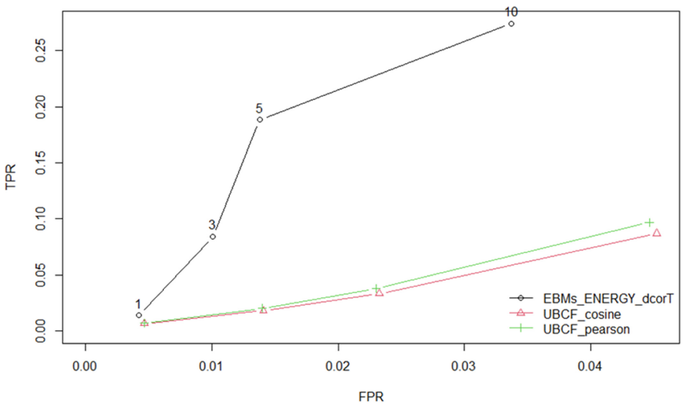

Fig. 4. The graph of ROC curves on MSWeb dataset with three measures.

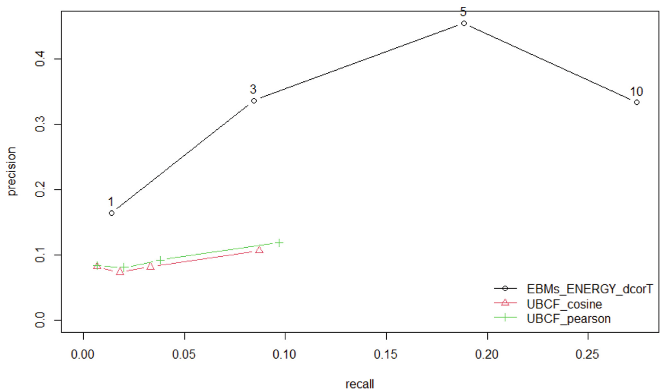

Fig. 5. The recall chart on MSWeb dataset with three measures.

Table 7. Recommendation table with independent bias.

	Precision	Recall	TPR	FPR	n
$EBMs_ENERGY_dcorT	0.1639344	0.01373053	0.01373053	0.00423476	1
$UBCF_cosine	0.08196721	0.00659319	0.00659319	0.004647951	1
$UBCF_pearson	0.08360656	0.006767633	0.006767633	0.004639796	1
$EBMs_ENERGY_dcorT	0.3366120	0.08419550	0.08419550	0.01007666	3
$UBCF_cosine	0.07322404	0.01791563	0.01791563	0.014079057	3
$UBCF_pearson	0.08087432	0.020058621	0.020058621	0.013963848	3
$EBMs_ENERGY_dcorT	0.4550820	0.18833600	0.18833600	0.01378538	5

(*continued*)

Table 7. (*continued*)

	Precision	Recall	TPR	FPR	n
$UBCF_cosine	0.08163934	0.03314451	0.03314451	0.023250781	5
$UBCF_pearson	0.09213115	0.037918181	0.037918181	0.022987034	5
$EBMs_ENERGY_dcorT	0.3337705	0.27411552	0.27411552	0.03371074	10
$UBCF_cosine	0.10672131	0.08693352	0.08693352	0.045228540	10
$UBCF_pearson	0.11852459	0.096851547	0.096851547	0.044632079	10

5 Discussion

With two scenarios, the experiment results show that with two different datasets (MSWeb (binaryRatingMatrix) and Learning from Sets of Items 2019 (realRatingMatrix), The precision rating of the proposed model is higher than the precision rating of the traditional model, even the number of evaluations (and) are changed. The proposed model works well on datasets with different data types.

With the type of dataset is realRatingMatrix, the results show that the precision of the model is higher than the precision of the model with the type of dataset to be binaryRatingMatrix in Table 8. Precision of the MSWeb and Learning from Sets of Items 2019 dataset.

Table 8. Precision of the MSWeb and Learning from Sets of Items 2019 dataset.

	Dataset	Precision	n
$EBMs_ENERGY_dcorT	**MSWeb**	0.1182609	1
		0.1391304	3
		0.1356522	5
		0.1299130	10
	Learning from Sets of Items 2019	0.1639344	1
		0.3366120	3
		0.4550820	5
		0.3337705	10

6 Conclusion

In this paper, we proposed a collaborative filtering model based on energy distance correlation. We used the energy distance correlation algorithm between users to recomend for user, who need of advice. In order to have a new direction in using energy in the recommender system.

To evaluate the model's effectiveness, we have experimented on the MSWeb and Learning from Sets of Items 2019 datasets with three measures: the model with the dcorT measure (EBMs_ENERGY_dcorT); UBCF user-based collabora-tive filtering model with the Cosine measures (UBCF_cosine) and Pearson measures (UBCF_pearson). The experimental results show that the proposed model is more accurate than the traditional model and the proposed model runs well on both binaryRatingMatrix and realRatingMatrix.

References

1. Woerndl, W., Schlichter, J.: Introducing context into recommender systems. In: AAAI - Workshops, pp. 138–140 (2007)
2. Felfernig, A., Jeran, M., Ninaus, G., Reinfrank, F., Reiterer, S., Stettinger, M.: Basic approaches in recommendation systems. In: Robillard, M., Maalej, W., Walker, R., Zimmermann, T. (eds.) Recommendation Systems in Software Engineering, pp. 15–38. Springer, Berlin, Heidelberg (2014). https://doi.org/10.1007/978-3-642-45135-5_2
3. Mpela, M.D., Zuva, T.: A mobile proximity job employment recommender system. In: 2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD), pp. 1–6, (2020)
4. Ricci, F., Rokach, L., Shapira, B. (eds.): Recommender Systems Handbook. Springer, Boston, MA (2015). https://doi.org/10.1007/978-1-4899-7637-6
5. Lu, J., Wu, D., Mao, M., Wang, W., Zhang, G.: Recommender system application developments: a survey. Decis. Support Syst. (2015)
6. Sindhwani, P.M.V.: Recommender systems. Commun. ACM 1–21 (2010)
7. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)
8. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72079-9_10
9. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 291–324. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72079-9_9
10. Cai, Y., Leung, H., Li, Q., Min, H., Tang, J., Li, J.: Typicality-based collaborative filtering recommendation. Knowl. Data Eng. IEEE Trans. **26**(3), 766–779 (2013)
11. Hussien, F.T.A., Rahma, A.M.S., Wahab, H.B.A.: Recommendation systems for e-commerce systems an overview. J. Phys. Conf. Ser. **1897**, 1–14. IOP Publishing (2021)
12. Tang, J., et al.: Recommendation with social dimensions. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), pp. 251–257 (2016)
13. Rizzo, M.L., Szekely, G.J.: Energy distance. WIRES Comput. Stat. **8**(1), 27–38. Wiley (2016)
14. https://github.com/mariarizzo/energy
15. Székely, G.J., Rizzo, M.L.: Brownian distance covariance. Ann. Appl. Stat. **3**(4), 1236–1265 (2009)
16. Szekely, G.J., Rizzo, M.L.: The distance correlation t-test of independence in high dimension. J. Multivar. Anal. **117**, 193–213 (2013)
17. Székely, G.J., Rizzo, M.L., Bakirov, N.K.: Measuring and testing independence by correlation of distances. Ann. Stat. **35**(6), 2769–2794 (2007)
18. Breese, J.S., David, H., Carl, M.K.: Anonymous web data from. Microsoft Research, Redmond WA, 98052–6399, USA, (1998). https://www.microsoft.com/en-in/

19. Sharma, M., Harper, F.M., Karypis, G.: Learning from sets of items in recommender systems. In: Proceedings of the ACM Transactions on Interactive Intelligent Systems (TiiS) (2019)
20. https://cran.r-project.org/web/packages/energy/index.html
21. https://cran.r-project.org/web/packages/arules/index.html
22. https://cran.r-project.org/web/packages/recommenderlab/index.html