ISSN- 2394-5125 VOL 08, ISSUE 02, 2021

A PRODUCT SEARCH SYSTEM THAT LEADS THE USER TO THE MOST PROFITABLE PRODUCT

Zahra Fasihi[1](*), Mohammadreza Razzazi[2]

[1]Faculty of IT, Amirkabir University of Technology (Polytechnic), Tehran, Iran
[2]Faculty of IT, Amirkabir University of Technology (Polytechnic), Tehran, Iran
[1]za.fasihi@aut.ac.ir, [2]razzazi@aut.ac.ir

ABSTRACT: Nowadays, the variety and complexity of products, particularly electronicproducts are so high that most users prefer to consult an expert before making a purchase. Thus, an attempt was made to design a system that notifies the user of some important aspects of the commodity. In the proposed approach, with the help of data mining algorithms, an attempt is made to explore the relationships between diverse attributes of the product and save the resulting rules to be utilized in the system. The system attempts at notifying the user and leads him/her to a better purchase by revealing these rules.
By observing these principles, the user reconsiders his/her choice and repeats the search process. This continues until the user is assured that the requested product has exactly the same attributes. After this stage, from the attributes of the product that the user has requested and related attributes that have been discovered as association rules, a structure will be created as "*Relation graph*", then "*utility graph*" is dynamically created for the user and by using the Analytic Hierarchy Process (AHP) with a few changes, Finally the most suitable and profitable commodities are suggested for the purchase.

KEYWORDS: Analytic Hierarchy Process; Multi attribute search; Product search;Multi-Criteriarecommender systems.

1 INTRODUCTION

One of the most important demands of customers in e-commerce is to find the right product in terms of their needs, quality and the price. Hence, a tool to help the user to find appropriate goods is very important.

What the user needs most is to know what each attribute of the product is and what effect it has on other attributes and whether there is a product that can meet all of their requirements.

One way to solve this problem is to use probabilities. In that sense, when several conflicting attributes are given as input, the product that has a greater number of attributes or is more probable to meet the user's expectations can be suggested. The problem with this approach is the user's unawareness of the "confliction".

In the proposed approach, the user is directed toward a product that provides him/her with the most satisfaction and benefit. Thus, through the discovery of rules among the attributes of products and storing them, the user becomes aware of these hidden relationships among attributes of products and can reconsider the attributes if deemed necessary. With these rules, the "utility graph" is then dynamically created for the user and according to that, the product that provides the user with the most benefits and is more likely to be purchased is identified and suggested.

In fact, the proposed system guides the user and somehow plays the role of an "advisor" in purchasing goods.

2 Background

In some systems, users can explicitly provide their general preferences on the multi-attribute content of items that can be used by various searching and filtering techniques to find the most relevant items [2]. In some recent researches, implicit features are also included.

For example, [3] is a system which suggests hotels that are personalized and tailored for the given user. The system employs natural language processing and topic modeling techniques to assess the sentiment of the users' reviews and extract implicit features. The entire recommender engine contains multiple sub-systems, namely users clustering, matrix factorization module, and the hybrid recommender system. Each sub-system contributes to the final composite set of recommendations through covering a specific aspect of the problem. Another example is [4], which presents a novel multi-criteria recommendation, based on the idea of clustering

ISSN- 2394-5125 VOL 08, ISSUE 02, 2021

users in "preference lattices" according to their criteria preferences and overall ratings of items are used for prediction.

[5] and [6] can also be classified into a new category in which features are retrieved rather than user preferences. [5], uses a fuzzy-based similarity metric to retrieve relevant products/services and their associated information from the product taxonomy file. [6], combine collaborating filtering and taxonomy of products and present a taxonomy-aware denoising autoencoder based model which incorporates taxonomy-aware side information into denoising autoencoder based recommendation models.

Some case-based recommender systems [7, 8] allow users to "critique" the recommendation results by refining their requirements as part of the interactive and iterative recommendation process, which uses various search and filtering techniques to continuously provide the user with the updated set of recommendations [2].

According to [9], these systems are divided into three major types: Natural conversational critiques systems, System-suggested critiquing systems and User-initiated critiquing systems. Some recent works are also a hybrid of these three types. For instance, [10] motivates a model in which consumers learn (update) their preference weights. When consumers learn preference weights, it may not be optimal to recommend the product with the highest option value, as in most search models, or the product most likely to be chosen, as in traditional recommendation systems.*Dzyabura* and *Hauser* in [10] believe that recommendations are improved if consumers are encouraged to search products with diverse attribute levels, products that are undervalued, or products for which recommendation-system priors differ from consumers' priors.

Among new System-suggested critiquing systems, we can mention [11], in which we learn the user expectations by different learning-to-rank methods. These methods include Pointwise Ranking, Pairwise Ranking and Listwise Ranking. *Yong* have applied utility-based methods to multi-criteria recommendations.

Another example of the hybrid solutions is [12], which proposes an approach that extends in experience-based critiquing (EBC), an effort to improve its efficiency over both standard critiquing and incremental critiquing.

Multi-criteria ratings

According to [2], some researches tried to design a total order on items and obtained a single global optimal solution for each user, whereas others took one of the possible partial orders of the items and found multiple (Pareto optimal) solutions. For example, the approach by FrancescoLolliet al. [13] could fall into the second category. ItStarts from a linear optimisation model aimed at searching for the most discriminating vector of weights, three quadratic variants are proposed subsequently to overcome the issues arising from the linear model. An iterative quadratic optimisation model is proposed to fit the real setting in which the application should operate, where the eliciting procedure must be launched iteratively and converge over time to the vector of weights. Finally, three experiments are performed to confirm the effectiveness and the differences between the proposed models. Another model [14], which is called "Bayesian probabilistic tensor factorization for multicriteria (BPTF-MC)," predicts the overall rating and the rating from each viewpoint simultaneously.It does this by using multicriteria latent features as additional factors. Manouselis and Costopoulou [15] propose a method that calculates total utility U either by summing the k predicted partial utilities on each criterion (Uc) or by weighting the predicted ratings that the user would give on each criterion c by the user's importance weights (Wc).

Based on a new categorization method [16], Multi-criteria recommendation techniques can be classified by their utility function into two categories: *memory-based* and *model-based* techniques. In memory-based techniques, the similarity can be computed from the multi-criteria ratings in two ways: the first approach aggregates traditional similarities values that are calculated separately on each criterion into a single similarity using one of the following aggregation methods (average [17], worst-case [17], and weighted sum of individual similarities [18]). The second approach calculates the distance between multi-criteria ratings directly using multidimensional distance metrics (Euclidean, Manhattan, and Chebyshev distance metrics) [17]. Model-based approaches build a predictive model in order to predict unknown ratings, such as the Aggregation-function-based approach [17] coming from the assumption that an item's overall rating is not independent of other ratings but, rather, there exists a relation between the overall rating and the multi-criteria ratings; Probabilistic Modeling [19], Support Vector Regression (SVR) [20], Multilinear Singular Value Decomposition (MSVD) [21], and Genetic Algorithm (GA) [22].

One solution for rating (can be categorized as the first approach mentioned abow), is using AHP which is used in this paper. AHP was proposed by Saaty (1977, 1980) to model the subjective decision-making processes based on multiple attributes in a hierarchical system [23]. The first level indicates the goal for the specific decision problem which is "buy or not" in our field. For example, in [24] to recommend a product while considering it as multi criteria, a hierarchical structure is used which can meet "and"/"or" conditions.

ISSN- 2394-5125 VOL 08, ISSUE 02, 2021

The main focus of our study is edges between nodes in the criteria tree that are not based on the *include* relationship like [24] but based on *effect*. However, our edges can also have the *include* relationship and the solution is the same.

3 Description of the Proposed System

The architecture of our system is shown in Figure 1. As it can be seen, the system consists of two components, each component presented in details as follows:

- Discovering the relationships[1] between the attributes of a product and creating the “relation graph”.
- Creating the customer “utility graph” and comparing each item with it and selecting the best and most profitable goods.

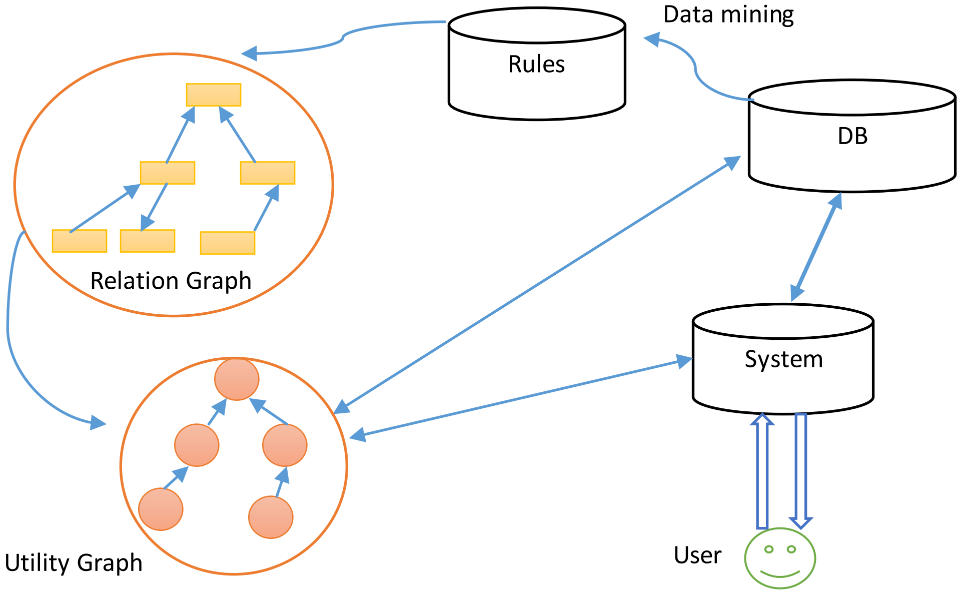

Figure 1. System architecture

An attempt is initially made to explore the relationships between different attributes of a product and save the result which would be utilized in the system. Thus, by presenting these results to the user, the system attempts to notify the user to make a better purchase. These relationships can be recorded in the system by an expert or be done with the help of data mining algorithms. However, a product’s variety of attributes should be considered in advance. Initially, we examine the variety of attributes of a typical product, a washing machine:

- Numerical and quantitative attributes such as price and users’ rating.
- Grouped or non-ordinal attributes such as company, color, country of production.
- Grouped-ordinal attributes such as energy consumption category.

3.1 Declarations

In accordance with any of attribute types, a method must be presented to calculate the relationships between attributes. In order to discover these relationships, a hybrid method was used [25]. These relationships actually consist of "association rules"([26]) plus hierarchical relationships or sub-relationships that will be exemplified later.

To increase the precision and accuracy of the discovered rules, revisions on rules are made with the supervision of an expert (an expert can add rules that are correct and uncovered in data mining due to lack of information), these rules must be stored in an ontology or graph. Figure 2 illustrates an example of this graph.

[1]These relationships are introduced in detail in "Declarations" section.

ISSN- 2394-5125 VOL 08, ISSUE 02, 2021

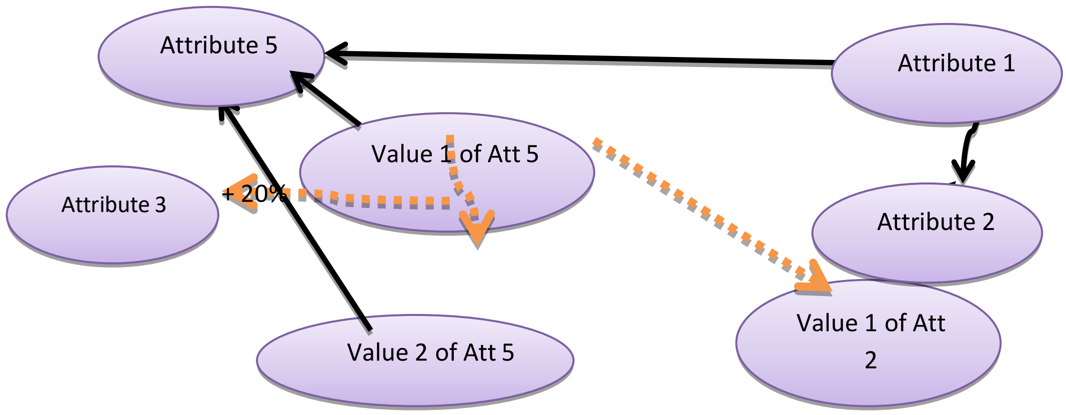

Figure 2.An example of relation graph between product attributions

The relation graph is defined as below:

G = (V, E)

E = {E1∪ E2 ∪ E3}

V: {attributes of a product ∪ (possible values of an attribute | V is a non-quantitative attribute)}

E1: {dotted line between two nodes}

E2: {dotted line between an edge and a node}

E3: {solid line}

|E|: number of discovered rules or relationships

|V|: number of attributes or their offspring which are connected to the edges or rules.

In figure 2, possible values of an attribute can be added to the graph only if it is a non-quantitative attribute like “Made-In” or “color”.

The dotted lines indicate the impact of an attribute on the others (association rules).

The solid lines represent the relationship between “parent” and “child” or “composed of”. For example, “security” is an attribute and “air bag” is its child.

In addition, in this graph, one edge may impact an attribute, meaning that the change is from one attribute or amount to another. For example, if the user chooses value 2 of attribute 5 instead of value 1 of the same attribute, attribute 3 increases by 20%.

So, in relation graph we have three types of edges:

The solid edge represents the parent and child hierarchical relationship (like consumption which consists of water, electricity, etc.) or an attribute with its possible values (like color which could be white, black, … based on our data) we call this kind of edge as "*Hierarchical edge*".

The dotted edge that connects the two nodes indicates the impact of a numerical (or non-numerical) attribute on another (like the clothing capacity which impacts the price at the rate of 1.2) we call this edge "*Association edge*".

The dotted edge that connects an edge with a node indicates the impact of changing the value of an attribute on another (like price change, if the color changes from white to silver) we call this edge "*Transition effect edge*".

3.2 Creating product utility graph for the customer

So far, we have come to the conclusion that the user or customer demands his or her requested item along with one or more desirable attributes. After extracting these attributes from user queries, it is time to guide the user. That is, by searching in the relation graph of a product, we must see whether the demanded attribute has influenced other attributes or not, and if it has, inform him/her of this relationship. To achieve this goal, we refer to the relation graph, find the user-specific attributes, and search for the edges. By finding *Association edge* and *Transition effect edge* and then presenting them to the user, he/she may reconsider his/her search, or disregard it. This process continues until the customer is certain of his/her considered attributes and without looking back to modify the inquiry, he/she looks for the product. At this moment, according to the user's demanded attributes and the relation graph, our system enters the next phase and seeks to find the most suitable and profitable product for the customer. In the following phase, finding the suitable product for the costumer turns to a decision-making issue with multi attributes or the Multi Attributes Utility Theory (MAUT) [27].

Similar to [24], we consider the benefit or score of each item or attribute as below:

(Here, $\sum_{i=1}^{n} W_i$does not need to be 1.)

$U(X) = \sum_{i=1}^{n} W_i \times U_i(X_i)$

Wi is the weight of the significance of the*i*th attributefor the userand Xiis the scoreof the *i*th attribute that would be defining in the following.

Scoring method

In this section, we want to describe our method for scoring each node and finally each product. Initially, the utility graph is constructed based on the relation graph between the product attributes (figure 2).

AS it was mentioned, in figure 2, we have three types of edges. In constructing our suggested utility graph, the first and second edges are easily used, but the third edge is slightly transformed to the second edge and then used.

Thus, in constructing the utility graph, the intended user attributes are considered, and all the offspring attributes (nodes having type *Hierarchical edge* in figure 2) are scaled to the last level (the level that there are no input edges to the nodes anymore.). Then, we include other attributes that impact the user-specific attributes (nodes having type *Association edge* in figure 2) and scale them to the last level. The user-specific attributesalong with all scaled attributes are added to the utility graph.

We add the entire scaled path along with the attributes in between to the utility graph (along with the impact rate of that attribute). If the sum of these rates is not 1, no problem occurs because normalization takes place.

To convert a third type edge to a second type, the following is applied. As it is known, the third type edge is as follows:

$A \rightarrow B \Rightarrow C \rightarrow xC$

This means that by changing the attribute *A* to *B* (They are children or values of another attribute, having type *Hierarchical edge* in Fig. 2), the attribute *C* changes with the rate of *x*. Based on "Propositional Logic", equation above is equivalent to:

$B \rightarrow xC$

Thus, one of the attributes is eliminated and the other attributes are relatively solved.

The equation above, similar to the second type edge (*Association edge*), is as follows:

$B \xrightarrow{x} C$

The equation above means attribute B will change value of attribute C with the rate of x. Note that if the new rule already existed in our discovered rules, we would no longer add it.

Regarding the price attribute, we consider two subsets or nodes that are called "value" and "real price". The subset of the value or usefulness of the attributes found in the database impacts the value of the products (potential price), and the real price is the same as the actual price of the product (real value in database), and the goal is in fact to introduce a product whose potential price is high compared to its real price.

At this stage, our graph has been completely constructed, thus, we have to choose a method to score nodes. As stated above, the AHP uses the weighted average (existing in MAUT) to score nodes. The proposed method is useful for general scoring of nodes, but in scoring, other issues should also be considered such as nodes in which the sum of weights of their subset or children edges is less than one or negative.

Initially, we consider two modes for the nodes:

1. Nodes in which a greater value is desirable, such as power.
2. Nodes in which less value is desirable, such as energy or fuel consumption.

The first group of nodes is abbreviated as *HVHD*[2], and the second group of nodes, *LVHD*[3].

---

[2] Higher Value Higher Desirable

[3] Low Value Higher Desirable

ISSN- 2394-5125 VOL 08, ISSUE 02, 2021

For both groups, we initially find set of entries which it's absolute value of weights among all subsets of entries, is maximum and call it "*MaxW"*. Therefore, according to the relation below, we calculate the score of each node or attribute *i* for the product *j*:

First situation: if attribute*i*is a Grouped or non-ordinalone (like "red" as a color):

$Scoreij$ = 1, If product *j* has the attribute *i* and $Scoreij$ = 0, If product *j* doesn't have attribute *i.*

Second situation:Assume "n" is number of products of the same type. Letbe thevalue of attribute *i* of product *j*:

$$Score_{ij} = \frac{V_{ij}}{Max(|V_{ik}|)\ k=1\ to\ n}$$

In the above formula, if "*i*" is a LVHD attribute, multiply Score by -1. If the user has specified the desired value, Max Vik would be desired value, and whenever Score ij is more than 1, we use 1 instead.

If attribute "*i*" was not numeric and was Grouped-ordinal (type 3), we use the level instead of the value (according to the scoring function).

Third situation: If product*j*does not have a value in the database for attribute*i*.

$$Score_{ij} = \frac{\sum_{k=1}^{k=d}(Score_{kj} * W_{ki})}{\sum_{k=1}^{k=d}(Max(|Score_{kj}|) * W_{ki})k \mid W_{ki} \in MaxW\ and\ j = 1\ to\ n}$$

If "*i*" is a LVHD attribute, multiply Score by -1.

In the equations mentioned above is the score of product *j* in attribute *i, n* is the number of products, *k* is the attribute having a relationship with attribute *i* and affecting it (*i*'s child), *d* is the number of attributes affecting attribute *i* and is the weight of the relation between attributes *k* and *I*, also as it was mentioned, *MaxW* is a subset of edges that maximize the absolute total of weights.

Regarding the price which also has a real value in addition to the value obtained by the rules (potential value), we must consider the weight of both groups, that is, the weight of the rules and the weight of the real value. This weight can be considered arbitrarily by the system administrator, or the absolute total of weights in *MaxW* (called "*MaxTotal"*) can be considered for the node of the rules and 1- *MaxTotal* for the real value.

3.3 Running the method by an example

Suppose that after data mining and recording the rules, the relation graph is obtained according to figure 3.

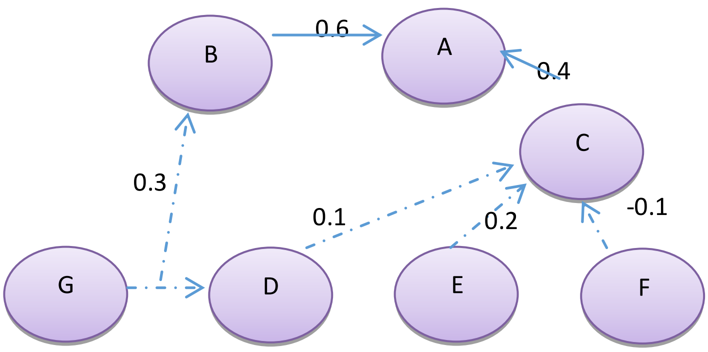

Figure 3.Example of relation graph

Suppose that attribute "A" is important to the user and requests it at maximum value. As mentioned above, to create the utility graph, we take attribute "A" from the product relation graph and then follow all the solid and dotted edges entering it, and add each node in the path, so that no edges remain. We use the edge weighs 0.3 to guide the user, and if the user has chosen the attribute D or G, by displaying this edge as a rule, it is used for awareness and guidance or advice to the user. Also, as previously explained, this edge is put forth as the following rule:

$D \xrightarrow{0.3} B$

So, our utility graph would be as follows:

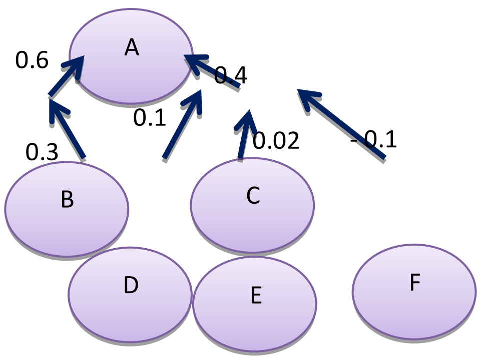

Figure 4.Utility graph

Suppose our products are in accordance with Table 1.

Table 1. Example of products

Name	NodeD	NodeE	Node F
P1	1	30	0
P2	1	24	1
P3	0	35	1

In order to calculate the score for all three products of node "C", first we will count *MaxW* (here {EC, DC}). Node "C" is the kind not having a value in the database,so we calculate it from the second formula (assuming the attribute as HVHD type).

$$Score_{CP1}\frac{(0*-0.1)\ +\ (30*0.02)\ +\ (1*0.1)}{(35*0.02)+(1*0.1)}=0.87$$

$$Score_{CP2}\frac{(1*-0.1)+(24*0.02)+(1*0.1)}{(35*0.02)+(1*0.1)}=0.6$$

$$Score_{CP1}\frac{(1*-0.1)+(35*0.02)+(0*0.1)}{(35*0.02)+(1*0.1)}=0.75$$

The score of the product "*P1*" in attribute "*C*" is the highest.

4 Assessment

The proposed system is dependent on user satisfaction or dissatisfaction more than anything else. For this purpose, a benchmark has been used to measure customer satisfaction after and before using this system. For this overall evaluation, the t-test has been used with SPSS software. The details of this assessment are described below.

4.1 Assessment assumptions

In implementing this test, the following assumptions are made:

The dependent variable in this test is the users ranking of the system after using it, before using it and through the normal search, which contains numerical values from 0 to 5.

The independent variables or groups in this test are the normal search system (similar to Digikala [1]) and the proposed system (our test type is Paired t-test).

The number of participants in this assessment is 50.

The hypothesis in this test is that the degree of user satisfaction with system guidance and proposed commodities is more than the exact and normal searching system.

The zero hypothesis in this test is that user satisfaction rate has not changed or has decreased.

ISSN- 2394-5125 VOL 08, ISSUE 02, 2021

4.2 Results

After collecting user information after and before using the system, the results are as follows:

According to Figure 5, the mean of the second row1, which is the average rating after using the system, has increased. However, we should consider whether this increase is significant or random.
The value of "sig (2-tailed)" indicates meaningfulness of the test:
sig. = 0.290/2 = 0.145
Given that 0.145 is not less than the first type error of 0.05, we cannot completely reject the zero hypothesis, that is, our problem is still open.

T-Test

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	1	3.7400	50	1.02639	.14515
	2	3.9000	50	1.08327	.15320

Paired Samples Correlations

		N	Correlation	Sig.
Pair 1	1 & 2	50	.499	.000

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	1 - 2	-.16000	1.05676	.14945	-.46033	.14033	-1.071	49	.290

Figure 5.Results of T-Test on the samples

In the next stage, according to the relatively high standard deviation in the ranking average, we classify the participants and then repeat the test again. In this way, the participating users are asked to enter their knowledge level regarding the searched product in the form of "large", "moderate", and "low" values.

After this stage, another variable is added to our t-test, which is the level of the user's knowledge that we call "Knowledge Level". This variable is set with three values of 1, 2, and 3, which in turn represents the low, medium, and high level of knowledge. Afterwards, the evaluation is done in the presence of this new variable once again.

The following results are achieved for the first category of people who have little knowledge.

T-Test

```
[DataSet1] C:\Users\Ya Ali\Desktop\test1.sav
```

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	1	3.9063	16	1.06800	.26700
	2	4.5313	16	.61830	.15457

Paired Samples Correlations

		N	Correlation	Sig.
Pair 1	1 & 2	16	.282	.289

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	1 - 2	-.62500	1.07238	.26810	-1.19643	-.05357	-2.331	1	.034

Figure 6.Results of T-Test in the first category of users

According to the amount of sig. (0.034), the zero hypothesis is rejected and the assumption hypothesis is correct. In the first category of users, or those who are less knowledgeable of the requested product or its market, the average user satisfaction of the system has increased.

Figure 7, shows the evaluation results for the third category. This category of users has enough information about the product. As it is observed in this category, the average user satisfaction of the system has reduced compared to a simple and fast search of the product, and the assumption hypothesis is that the average satisfaction is less if the system is used. According to the P-Value (0.038), the zero hypothesis is false and the assumption hypothesis is correct.

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	1	3.5000	7	.70711	.26726
	2	2.5000	7	1.04083	.39340

Paired Samples Correlations

		N	Correlation	Sig.
Pair 1	1 & 2	7	.396	.379

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	1 - 2	1.00000	1.00000	.37796	.07515	1.92485	2.646	6	.038

Figure 7.Results of the third category of users

In the second category of users, it is not possible to come to a definite conclusion; however, the average satisfaction of these people has generally increased.

Of course, if we remove the unconventional data from the samples, which consist of 3 users who give the system a low score outside the confidence interval of 95%, the evaluation results are as follows. According to the P-value (0.0275), it can be concluded that the average satisfaction of individuals has generally increased and our hypothesis is proven.

ISSN- 2394-5125 VOL 08, ISSUE 02, 2021

Paired Samples Test

	Paired Differences					t	df	Sig. (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference: Lower	95% Confidence Interval of the Difference: Upper			
Pair 1 1 - 2	-.30851	1.07619	.15698	-.62449	.00747	-1.965	46	.055

Figure 8. Results of 95% confidence interval

5 Conclusion

In this article, an attempt was made to design and implement a system to direct users in selecting more beneficial products. The innovation of this system is that it primarily guides the user with the help of the rules, which were discovered about the impacts of various attributes on each other so that the user is aware of the product attributes and its impact on the others. After this stage, the product utility graph is dynamically constructed and the effective rules impacting each other are used to construct this graph. Finally, a product is offered to the user that has a balance of the sum of requested user attributes and desirability.

In the construction of the utility graph, AHP has been used differently and with regard to the negative edges and also the abnormal coefficients (the total weight of the edges is not equal to one).

Based on the results of the evaluation done, users will be divided into two groups in accordance to such a system. Those who welcome the system because they have enough opportunity to examine the discovered principles and are not knowledgeable of the product, and the second category refers to those who were not interested in the system due to incompatibility of the proposed product with all the demanded attributes or the waste of time using the system. It seems that if people in the second category who are dissatisfied with the product offer, while receiving the proposed product, see the positive attributes of that product as principles, their satisfaction will increase. That is, in this case it becomes fully understood why this product is more profitable than the others.

Another factor that may have a positive impact on user satisfaction is an increase in the readability of the rules presented to the user. For example, the user does not know what (capacity$\xrightarrow{+0.2}$price) means. By improving the readability of these principles, the user's understanding and satisfaction of the system will increase.

6 Future studies

It seems that measuring the impact rate of one particular attribute on another is of great significance. This rate is either determined by experts or by obtaining the average of rates. It is better to calculate this rate more accurately in the future.

In Section 3, a correlation coefficient is used which almost linearly detects the effects of the attributes on each other. Identifying the impacts of attributes on each other using a nonlinear method would yield a positive study. For example, it is observed that as the price increases, user ranking for a product increases, but this trend is accurate up to a certain price. Afterwards, the user's ranking will decrease with increasing prices.

References

[1] www.digikala.com

[2] Adomavicius, G., Manouselis, N., Kwon, Y. Multi-Criteria Recommender Systems. Recommender Systems Handbook, Springer, Boston, MA, **(2015)**, pp. 847-880.

[3] Ebadi, A., Krzyzak, A. "A Hybrid Multi-Criteria Hotel Recommender System Using Explicit and Implicit Feedbacks", International Journal of Computer, Electrical, Automation, Control and Information Engineering,**(2018),** pp. 1450-1458.

[4] Liu, L., Mehandjiev, N., Xu, D.L. "Multi-Criteria Service Recommendation Based on User Criteria Preferences", Proceedings of the fifth ACM conference on Recommender systems,**(2011)**, pp. 77-84, Chicago.

[5] Prateeti Mohapatra, Anupama Ray, Gargi Dasgupta, “FuTSe: a fuzzy taxonomy service to facilitate product search”, CoDS-COMAD '18: Proceedings of the ACM India Joint International Conference on Data Science and Management of Data,**(2018)**, pp. 220–229.

[6] Chunhong Zhang, Tiantian Li, Zhibin Ren, Zheng Hu, Yang Ji., “Taxonomy-aware collaborative denoising autoencoder for personalized recommendation”, Applied Intelligence volume 49,**(2019)**, 2101–2118.

[7] Burke, R., “Knowledge-based recommender systems”, Encyclopedia of Library and Information Systems, **(2000)**, pp. 75–186.

[8] Reilly, J., McCarthy, K., McGinty, L., Smyth, B., “Incremental critiquing. Knowledge-Based Systems”,18(4-5),**(2005)**, pp. 143–151.

[9] Chen, Li, Pu, Pearl, “Critiquing-based recommenders: survey and emerging trends”, User Modeling and User-Adapted Interaction,**(2012)**, pp. 125-150.

[10] Daria Dzyabura, John R. Hauser, “Recommending Products When Consumers Learn Their Preference Weights”, Marketing Science (Volume 38),**(2019)**, pp. 417-441.

[11] Yong Zheng, “Utility-Based Multi-Criteria Recommender Systems”, SAC '19: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing,**(2019)**, pp. 2529–2531.

[12] Salem, Yasser, Hong, Jun, “History-Aware Critiquing-Based Conversational Recommendation” Proceedings of the 22nd International Conference on World Wide Web,**(2013)**, pp. 63-64, Rio de Janeiro, Brazil.

[13] FrancescoLolli·EliaBalugani·AlessioIshizaka·RitaGamberini·Maria AngelaButturi·SamueleMarinello·BiancaRimini, “On the elicitation of criteria weights in PROMETHEE-based ranking methods for a mobile application”,Expert Systems with Applications,**(2019)**, pp. 217–227.

[14] Hiroki M., Satoshi, O.,Masahito, K.,“Bayesian probabilistic tensor factorization for recommendation and rating aggregation with multicriteria evaluation data”,Expert Systems with Applications, **(2019)**, pp. 1-8.

[15] Manouselis, N., Costopoulou, C., “Experimental analysis of design choices in multiattribute utility collaborative filtering”, International Journal of Pattern Recognition and Artificia l Intelligence, 21(2),(**2007**), pp. 311–332.

[16] NourNassar, Assef Jafar, Yasser Rahhal, “A novel deep multi-criteriacollaborative filtering model for recommendation system”,Knowledge-BasedSystems(Volume 187),**(2020)**, Available at https://doi.org/10.1016/j.knosys.2019.06.019.

[17] Adomavicius G., Kwon Y., “New recommendation techniques for multicriteria rating systems”, IEEEIntell. Syst., 22,(**2007**), pp. 48-55.

[18] Tang T.Y., Mccalla G., “The pedagogical value of papers: A collaborative-filtering based paper recommender”, J. Digit. Inf., 10, (**2009**).

[19] Sahoo N., Krishnan R., Duncan G., Callan J., “The halo effect in multicomponent ratings and its implications for recommender systems: The case of yahoo! Movies Inf”, Syst. Res.,(**2012**), pp. 231-246.

[20] Drucker H., Burges C.J.C., Kaufman L., Smola A., Vapnik V.N.,“Support vector regression machines”, Adv. Neural Inf. Process. Syst.,(**1997**), pp. 155-161.

[21] Hassan M., Hamada M., “A neural networks approach for improving the accuracy of multi-criteria recommender systems”, Appl. Sci., (**2017**), p. 868.

[22] Hassan M., Hamada M., “Genetic algorithm approaches for improving prediction accuracy of multi-criteria recommender systems”, Int. J. Comput. Intell. Syst.,(**2018**), pp. 146-162.

[23] Tzeng, Gwo-Hshiung, Huang, Jih-Jeng, “Multiple Attribute Decision Making: methods and applications”, Chapman and Hall/CRC, (**2011**).

[24] Schmitt, C., Dengler, D., Bauer, M., “THE MAUT MACHINE: An Adaptive Recommender System”, Hannover, Germany, (**2002**), Available at:https://pdfs.semanticscholar.org/63eb/722e49481ac6a1878cc7b65e6aa5ec9f6112.pdf.

[25] Razzazi Mohammadreza, Fasihi Zahra, “Discover the relationships between product attributes by mining association rules”, In 7th International Conference on Information and Knowledge Technology (IKT2015), (**2015**), Available at: Urmia Iran. https://www.civilica.com/Paper-ICIKT07-ICIKT07_151.html

[26] R.Agrawal, T.Imielinski, A.Swami, “Mining Association rules Between Sets of Items in Large Databases”, Proceedings of ACM SIGMOD, (**1993**).

[27] Von Winterfeldt, D., Edwards, W., “Decision analysis and behavioral research”, Cambridge University Press, (**1986**).