Constraint-based Recommender System for Commodity Realization

Hanna Yehoshyna, and Vadim Romanuke

Original scientific article

***Abstract*—In this paper, we suggest a novel recommender system where a set of appropriate propositions is formed by measuring how user query features are close to space of all possible propositions. The system is for e-traders selling commodities. A commodity has hierarchical-structure properties which are mapped to the respective numerical scales. The scales are normalized so that a query from a potential customer and any possible proposition from the e-trader is a multidimensional point of a nonnegative unit hypercube put on the coordinate origin. The user can weight levels. The distance between the query and propositions are measured by the respective metric in the Euclidean arithmetic space. The best proposition is defined by the shortest distance. Top N propositions are defined by N shortest distances. The system does not depend on any user experience, nor on the e-trader tendency to impose one's preferences on the customer.**

***Index Terms*—recommender system, query and propositions, experience independence, neutrality support.**

I. Introduction

Recommender Systems (RSs) are software tools and techniques trying to predict preferences of a user after one makes queries for items of one's certain interest [1, 2]. In commerce, the user is a potential customer interested in buying or ordering specific commodities, goods, wares, etc. From the side of the subject which realizes commodities, an RS should serve to maximize the profit. On the other side, the RS should serve to support and improve the quality of the decisions consumers make while searching for and selecting commodities online [3].

Nowadays, the e-commerce of any branch is interested in facilitating customers to cope with information overload [4, 5]. Indeed, this is clearly motivated by struggling to derive profits from the customer activity, whereas an information-overloaded potential customer quickly gets tired of it and thus the one becomes low-active (orders and buys less). First of all, RSs are to help less experienced customers which may become confused in assessing large amounts of proposed elements. If RSs are capable of suggesting a truly valuable (demanded) set of commodities to a customer, it has a positive impact on the customer's experience that strengthens both the e-commerce profitability and customer contentment [3, 6, 7].

However, it is necessary to distinguish the motivations as to why e-service providers introduce RSs and the benefits that RS provides for the user of this RS. One of the major functions of a commercial RS is an opportunity to sell an additional set of items compared to those usually sold without any kind of recommendation [1, 3, 5, 8]. Another important function is to enable the user to select items that might be hard to find due to their unpopularity [9]. RS also enhances the user experience of the application by increasing the user satisfaction. Also RS forms a description of user preferences, collected explicitly or predicted, that can be further used for a number of other purposes, for example, improving inventory management [10].

RS provides the solution of the following tasks for the user: it helps users discover new content and find the content they were already looking for [1]. In cases when the number of items is relatively small, the user is provided with all items that can meet one's individual needs, together with additional explanations generated by RS [11]. In order to improve the user profile, RS can personalize the customer experiences by using one's feedback. In the absence of specific knowledge about the active user, RS can only provide one with the same recommendations that would be delivered to an "average" user [12]. In some cases it is necessary to recommend a sequence of educational and entertaining items (e. g., TV series, e-learning courses, musical tracks) [13].

Different types of RSs differ in terms of the subject area and knowledge, but the main difference is the recommendation algorithm used. The recommendation algorithm defines how to predict that an item is worth recommending. Other differences relate to how recommendations are collected and presented to the user in response to user requests. There are basically 4 important types of recommendation engines: content-based, collaborative filtering, social and demographic, knowledge-based (contextual) [14, 15].

Content-based filtering (CBF) algorithms learn to recommend items similar to the ones that the user liked in the past. The similarity of items is calculated based on the features associated with the compared items. This approach also extends naturally to cases where item metadata is available (e. g., movie stars, book authors, music genres, etc.) [16].

The main idea of collaborative filtering (CF) is that there is given a matrix of preferences by users for items, and these are used to predict missing preferences and recommend items with high accuracy. It recommends to the active user the items that

Manuscript received May 27, 2021; revised October 5, 2021. Date of publication October 25, 2021. Date of current version October 25, 2021.

H. Yehoshyna is with the Department of Information Technologies, Odessa Polytechnic State University, Shevchenko av., 1, 65044 (e-mail: yehoshyna.h.a@op.edu.ua). V. Romanuke is with the Department of Technical Cybernetics and Information Technologies, Odessa National Maritime University, Mechnikova str., 34, Odessa, Ukraine, 65029 (e-mail: romanukevadimv@gmail.com).

Digital Object Identifier (DOI): 10.24138/jcomss-2021-0102

1845-6421/12/2021-0102 © 2021 CCIS

other users with similar tastes liked in the past. The similarity in taste of two users is calculated based on the similarity in the rating history of the users [17, 18].

Social and demographic recommenders suggest items based on the demographic profile of the user. The assumption is that different recommendations should be generated for different demographic niches. In addition, recommenders suggest items that are liked by friends, friends of friends, and demographically-similar people [19, 20].

Knowledge-based (KB) systems recommend items based on specific domain knowledge about how certain item features meet user needs and preferences and, ultimately, how the item is useful for the user requirements. Contextual recommendation algorithms recommend items that match the user's current context. In these systems, a similarity function estimates how much the user needs (problem description) match the recommendations (solutions of the problem) [21, 22].

There also hybrid recommender systems. These RSs are based on the combination of the abovementioned techniques. This approach often combines collaborative and content-based algorithms [23, 24].

As there always is a floating-uncertain gap between an RS and user experience, this study suggests an approach which completely removes the experience dependency. We suggest a novel RS trying to more rigorously interpret what the user requests, rather than impose e-trader interests on the user as a potential customer. The rigor is methodologically ensured by 0-1-normalization and subsequent measurement of the distance between the RS propositions and the user query.

The paper proceeds as follows. The motivation of our study and problem statement on its basis are briefly presented in Section II. Section III formalizes our approach to building an RS for commodity realization. In Section IV it is explained how we map item features to numerical scale. The results are further recapitulated in Section V. In this Section we also give a simple example of how an RS of hierarchical structure works. Besides, a visualization of a run of the RS is presented. Our approach is discussed in Section VI, where we settle bounds to potential ambiguities. The main findings of the study conclude the paper in Section VII.

II. Problem Statement

The relevance of recommendations generated by CF is possible only when the history of previous user interactions with the RS is of large amount. Besides, the current user (for whom a recommendation is formed) must have actively participated in building this history. Moreover, the CF-based RSs presume that the service is frequently used [17, 18]. Otherwise, the matrix of ratings is badly sparse [18].

RSs using CBF algorithms generate recommendations based purely on data of a current user. This approach requires a preliminary description of the user's preferences. The description is available only either by the history of the user's requests (queries) or by preliminarily questioning the user [25].

Both the approaches propose poor recommendations if the user's data are of small amount. CF and CBF fit best for objects which are of a simple structure and requested (and bought) frequently. Therefore, the e-commerce services proposing commodities of a more complex structure, which, in addition, have poorer history (due to rarer requests or/and purchasing) avoid using CF and CBF algorithms [26].

KB methods use information about users and the field-of-request knowledge, but not ratings from the history of interactions. The user may be prompted to give a series of rules or guidelines on what the results should look like, or an example of an item. The cold-start problem does not affect KB recommenders since requirements are directly elicited from a recommendation session [27].

There are two main approaches to constructing systems of such a class. These are case-based and constraint-based recommenders differing in how a decision is formed [1, 21]. The case-based approach uses similarity metrics for comparing the current item (query, request, etc.) to the instances from the history of previous interactions [21]. Constraint-based approach uses predefined knowledge databases containing explicit rules (filter constraints) on how to correlate the user's request with the element's characteristics [26]. If the object corresponding to the user's requirements (demands) is not found, both the approaches use a method of determining a minimal set of changes in the requirements so that a solution could be found [28].

KB of constraint-based RS (CBRS) commonly includes five elements: two sets of parameters and three sets of constraints [29, 30]. The parameter sets relate to the user query features and features of objects (services) provided by the application. The constraint sets include constraints to those features and a bank of filters (rules) that describe interrelationship between user demands and object features. A solution to a given recommendation task is a complete assignment to the variables of the parameter sets such that this assignment is consistent with the constraints [29].

Typical approaches to solve a recommendation task are constraint satisfaction algorithms and conjunctive database queries. Solutions for constraint satisfaction problems are calculated on the basis of search algorithms that use different combinations of backtracking and constraint propagation. Solutions to conjunctive queries are calculated on the basis of database queries that try to retrieve items which fulfill all of the defined customer requirements [1].

Obviously, RSs are a valuable tool making communication and connection between the e-commerce (e-trader) and customer tighter and more reliable. However, there are a few demerits causing some controversial results of the RS application for commodity realization. First, tying a new-customer's recommendations to the experience (interaction history) of other customers may be biased. The matter is that such tying erases individuality (by diminishing individual requests or preferences). This can be partially rectified by the new-customer's registration. Nevertheless, registering in an RS system is a boring process which makes delays and thus defers possible transactions [12, 16]. For instance, if a customer wants to buy (or order) a simple commodity (e. g., in grocery e-commerce), the registration is a series of actions which are never present in real-world markets. This is an indeed distracting and diverting procedure that eventually causes refusal from using the RS. Second, an RS may generate too importunate recommendations (just like Netflix, Spotify, YouTube, etc., do) because they are projected based on conditions from e-commerce owners [3, 5]. Generally speaking, this violates the neutrality. The neutrality violation is confirmed also by the commonly recognized fact of that the e-commerce wants a customer to buy a commodity regardless of whether the one really needs it or not. These demerits badly diminish the RS user benefits declared by e-service providers.

Hence, an open question is whether RSs could be improved for commodity realization. A variant of CBRS is to be considered in this way. The improvement is intended to remove the experience-tie dependency along with the neutrality violation. For this, a method of straightforwardly comparing commodity items should be suggested. The method must not depend on any user experience, nor on the e-trader tendency to impose one's preferences on the customer. But, henceforth, it is necessary to distinguish the user experience from the quality of experience (QoE). The user experience in the RS application field reflects experience of a user within the subject area and knowledge (e. g., purpose, characteristics of selected commodities, their compatibility with other commodities, specificities of usage, etc.). In this case, QoE reflects a degree of the customer satisfaction with services. All the more, the main tasks of RS are about to increase the user fidelity and increase the user satisfaction [1]. The fulfilling of these tasks provides an increase in the QoE level.

III. A Novel CBRS Framework

A framework of the CBRS for commodity realization can be described as follows. We presume that a commodity to be realized has its properties of a hierarchical structure. The root property is a denomination of the commodity. The commodity denomination can be called the root category. The next in the hierarchy is a subcategory which is slave to the root category. This subcategory can be called the second level category. For instance, if a user/client is trying to buy wine, then the type/sort of wine is the root category, and the second level category implies an additional property of the selected sort such as agedness, dryness, sweetness, etc. The third level category is another additional property which can be selected. The third level is slave to the second level. The lowest possible level is commonly the price. Thus, if the commodity has maximum M nonnumeric properties (not including its price), then the CBRS framework consists of $M+2$ hierarchical levels (including the root and lowest levels).

The first (root) category is a set of N_1 items

$$h_1\left(1\right), h_1\left(2\right), ..., h_1\left(N_1\right),$$

or, briefly,

$$I_1 = \left\{h_1\left(i\right)\right\}_{i=1}^{N_1}. \tag{1}$$

Every item in set (1) has its set of items slave to this item. Thus, item $h_1\left(1\right)$ has a set of slave $N_2(1)$ items

$$h_2\left(1,\ 1\right), h_2\left(1,\ 2\right), ..., h_2\left(1,\ N_2(1)\right);$$

item $h_1\left(2\right)$ has a set of slave $N_2(2)$ items

$$h_2\left(2,\ 1\right), h_2\left(2,\ 2\right), ..., h_2\left(2,\ N_2(2)\right); ...;$$

item $h_1\left(N_1\right)$ has a set of slave $N_2(N_1)$ items

$$h_2\left(N_1,\ 1\right), h_2\left(N_1,\ 2\right), ..., h_2\left(N_1,\ N_2(N_1)\right).$$

Similarly to sets

$$I_2\left(i\right) = \left\{h_2\left(i,\ j(i)\right)\right\}_{j(i)=1}^{N_2(i)} \text{ by } i = \overline{1,\ N_1} \tag{2}$$

slave to set (1), every item in set (2) has its set of items slave to this item. Thus, item $h_2\left(1,\ 1\right)$ has a set of slave $N_3(1,\ 1)$ items

$$h_3\left(1,\ 1,\ 1\right), h_3\left(1,\ 1,\ 2\right), ..., h_3\left(1,\ 1,\ N_3(1,\ 1)\right); ...;$$

item $h_2\left(N_1,\ N_2(N_1)\right)$ has a set of slave $N_3(N_1,\ N_2(N_1))$ items. Thus, sets

$$\begin{gathered} I_3\left(i,\ j(i)\right) = \\ = \left\{h_3\left(i,\ j(i),\ k(i,\ j(i))\right)\right\}_{k(i,\ j(i))=1}^{N_3(i,\ j(i))} \\ \text{by } i = \overline{1,\ N_1} \text{ and } j(i) = \overline{1,\ N_2(i)} \end{gathered} \tag{3}$$

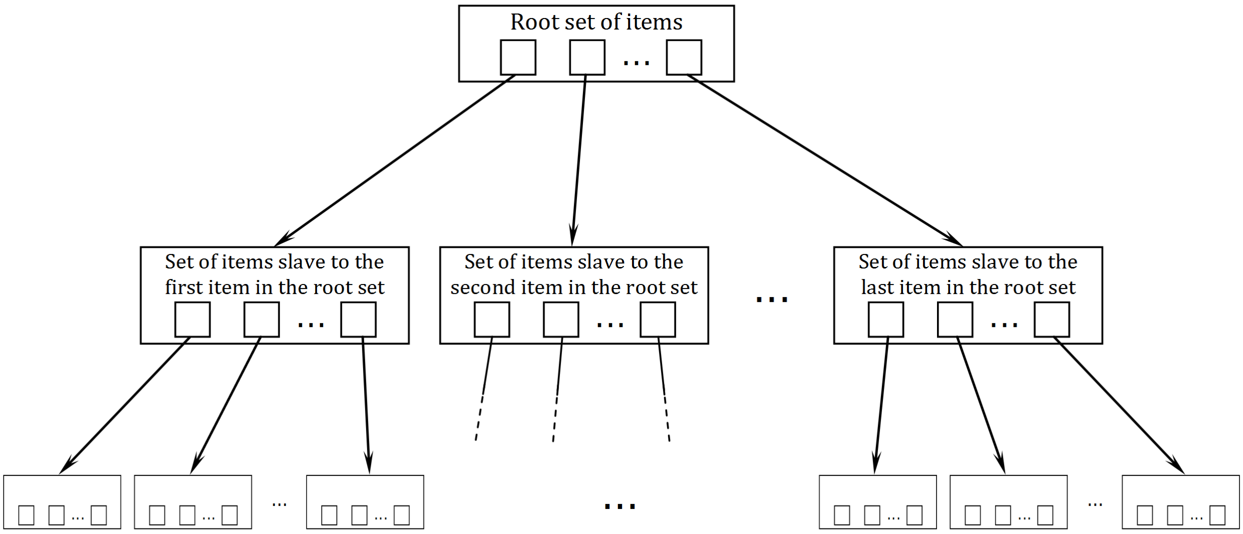

Root set of items
Set of items slave to the first item in the root set
Set of items slave to the second item in the root set
Set of items slave to the last item in the root set

Fig. 1. The CBRS framework hierarchical structure on the application side

are slave to sets (2). Shown in Figure 1, bunches of items at deeper levels (the fourth slave to the third, the fifth slave to the fourth, and so on) are constructed analogously.

In fact, we suggest a CBRS framework of a fractal type (Figure 1). A query (request) from a user is an $(M+2)$-dimensional point in $\mathbb{R}^{M+2}$. Denote it by

$$\mathbf{Q} = [q_k]_{1\times(M+2)} ,$$

where q_k is a value of a constraint imposed on an item of the k-th level. Values of $\mathbf{Q}$ are obtained within unit interval $[0;\ 1]$ by two-stage transformation from the user query, where all sets at every level are unified (Figure 2). In addition, these sets can be supplemented by other potential items not available from the application side. First, if an item at a level is descriptive (e. g., "red wine"), it is mapped to a numerical scale. This stage is omitted if the item is already given as a number (like at the price level). Then every numerical scale is normalized to interval $[0;\ 1]$ by dividing by the greatest value at the level. Denote this two-stage mapping by ψ. If the levels have their (priority) weights $\{w_k\}_{k=1}^{M+2}$, where

$$w_k > 0 \text{ by } k = \overline{1,\ M+2}, \tag{4}$$

then

$$\bar{w}_l = \frac{w_l}{\max\limits_{k=\overline{1,\ M+2}} w_k} \text{ by } l = \overline{1,\ M+2}, \tag{5}$$

and every scale is multiplied by $\bar{w}_l$ by $l = \overline{1,\ M+2}$.

A proposition (as an answer to the query) is an $(M+2)$-dimensional point in $\mathbb{R}^{M+2}$. Denote it by

$$\mathbf{P} = [p_k]_{1\times(M+2)} ,$$

where

$$p_1 \in \{\bar{w}_1 \cdot \psi(I_1)\} ,$$

$$p_2 \in \{\bar{w}_2 \cdot \psi(I_2(i))\} ,$$

$$p_3 \in \{\bar{w}_3 \cdot \psi(I_3(i,\ j(i)))\} , ...,$$

or, briefly,

$$p_k \in \{\bar{w}_k \cdot \psi(I_k)\} \text{ by } k = \overline{1,\ M+2}. \tag{6}$$

Denote by P a space of all possible propositions in the CBRS. The best proposition is

$$\mathbf{P}^* \in \arg\min_{\mathbf{P}\in P} \|\mathbf{Q} - \mathbf{P}\| =$$

$$= \arg\min_{\mathbf{P}\in P} \sqrt{\sum_{k=1}^{M+2} (q_k - p_k)^2}. \tag{7}$$

Obviously, the best proposition by (7) is the nearest neighbor to query $\mathbf{Q}$ within space P [31, 32]. To find top N propositions ($N \in \mathbb{N}$), we throw out the found proposition by (7), i. e.,

$$P^{(\text{obs})} = P,\ \ P = P^{(\text{obs})} \backslash \{\mathbf{P}^*\} , \tag{8}$$

and solve problem (7) again. Thus, operations (7) and (8) are successively repeated for N times.

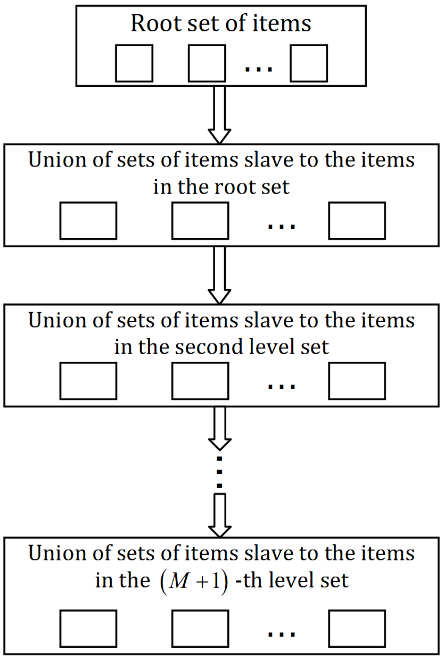

Root set of items
...
Union of sets of items slave to the items in the root set
...
Union of sets of items slave to the items in the second level set
...
Union of sets of items slave to the items in the $(M+1)$ -th level set
...

Fig. 2. The hierarchical structure for the user query

IV. Numerical Scale Mapping

Mapping descriptive items to numerical scales is a fundamental part in the suggested CBRS. Initially, a user query is a commodity denomination and a set of $M+1$ its properties. Every potential (and sometimes not available) commodity denomination and other descriptive items are evaluated by a set of experts. Thus they are mapped to their respective numerical scales whose ranges do not matter. The scales are nonnegative.

It is clear that items like price, volume, quantity are already given within their respective ranges. If a numerical item in a user query falls outside its (nominal) range, the range is straightforwardly widened. However, the ranges are most likely to be constrained so that a user would not request for unrealistically low price, too small or big volume, quantity, etc.

All the $M+2$ numerical scales are normalized to interval $[0;\ 1]$ by the greatest-value (maximum) division. Thus any units of measurements are removed. Finally, all available items, comprising space P of all possible propositions, and query $\mathbf{Q}$ become numeric. While the suggested CBRS serves, only user query is mapped to vector $\mathbf{Q}$.

V. Results

The 0-1-normalization is fulfilled for space P just once, unless the e-trader changes its commodity assortment or/and stock. The user query is 0-1-normalized every time a new object is requested. In general, proper functioning of such a CBRS requires only processing a current user query, rather than exploring the user interaction history or the rating of the respective request. Owing to the numerical scale mapping and 0-1-normalization, the descriptive item constraints are not parsed or linguistically analyzed but are computationally processed. Thus, compared to other CBRSs, the novel CBRS is built and serves far faster. Moreover, the novel CBRS cannot have anything close to the specific issue of that the RS cannot draw any inferences for users or items about which it has not yet gathered sufficient information [1, 2, 4, 18]. Therefore, the cold-start problem does not affect this CBRS.

As an example, consider a toy CBRS with three levels. This CBRS is intended to help both an e-trader and consumer in selling and buying, respectively, 3 sorts of the alcoholic beverage (say, beer, wine, and whiskey). This is the root level, wherein

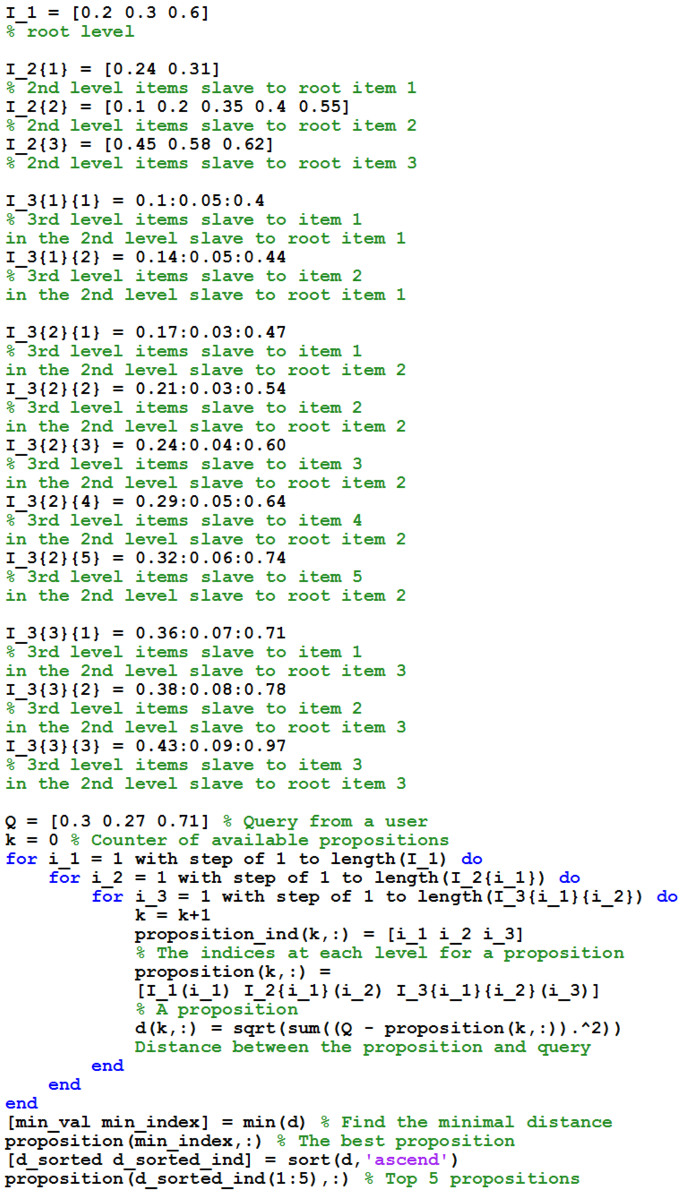
$$p_1 \in \{0.2,\ 0.3,\ 0.6\}\,.$$

The beer is sold in 2 sorts, the wine has 5 sorts, and the whiskey is sold in 3 sorts. This is the second level. Each sort of beer has 7 various prices. The 5 sorts of wine have 11, 12, 10, 8, and 8 prices, respectively. The first 2 sorts of whiskey have 6 prices, and the third sort has 7 prices. To find the best proposition along with top 5 propositions in this example, a Matlab-based pseudocode in Figure 3 is used, where the number in-between two colons is a step for a vector of normalized items. While this CBRS serves, only the query from a user is changed. A visualization of the toy CBRS example is presented in Figure 4, where the possible propositions are marked as squares, the user's query is a circle, and the top 5 propositions are circled (the closer proposition to the query is, the bigger circle is).

It is worth noting that in this example the commodity denomination is the alcoholic beverage, whereas a user queries for beer (due to $q_1 = 0.3$ and $p_1 = 0.3$ means beer). This is done for the purpose of that a user could query for anything else, not merely beer, wine, or whiskey. If, say, a user selects tequila (in the user root category), then, obviously,

$$q_1 \notin \{0.2,\ 0.3,\ 0.6\}\,.$$

Although the e-trader does not sell tequila, queries for tequila may be useful to register them and see later the respective statistics. If the amount of such queries is comparable to that of the available alcoholic beverages, then it is reasonable for the e-trader to start selling tequila along with other root items. Therefore, an additional utility can be built in the CBRS for registering and storing all the user requests. For instance, the storage may be monthly processed and analyzed with a purpose to detect the most requested items or objects. Later on, these top-requested items/objects may supplement the commodity assortment or/and stock of the e-trader. Null-requested objects may be successively expelled from the assortment/stock.



```
I_1 = [0.2 0.3 0.6]
% root level

I_2{1} = [0.24 0.31]
% 2nd level items slave to root item 1
I_2{2} = [0.1 0.2 0.35 0.4 0.55]
% 2nd level items slave to root item 2
I_2{3} = [0.45 0.58 0.62]
% 2nd level items slave to root item 3

I_3{1}{1} = 0.1:0.05:0.4
% 3rd level items slave to item 1
in the 2nd level slave to root item 1
I_3{1}{2} = 0.14:0.05:0.44
% 3rd level items slave to item 2
in the 2nd level slave to root item 1

I_3{2}{1} = 0.17:0.03:0.47
% 3rd level items slave to item 1
in the 2nd level slave to root item 2
I_3{2}{2} = 0.21:0.03:0.54
% 3rd level items slave to item 2
in the 2nd level slave to root item 2
I_3{2}{3} = 0.24:0.04:0.60
% 3rd level items slave to item 3
in the 2nd level slave to root item 2
I_3{2}{4} = 0.29:0.05:0.64
% 3rd level items slave to item 4
in the 2nd level slave to root item 2
I_3{2}{5} = 0.32:0.06:0.74
% 3rd level items slave to item 5
in the 2nd level slave to root item 2

I_3{3}{1} = 0.36:0.07:0.71
% 3rd level items slave to item 1
in the 2nd level slave to root item 3
I_3{3}{2} = 0.38:0.08:0.78
% 3rd level items slave to item 2
in the 2nd level slave to root item 3
I_3{3}{3} = 0.43:0.09:0.97
% 3rd level items slave to item 3
in the 2nd level slave to root item 3

Q = [0.3 0.27 0.71] % Query from a user
k = 0 % Counter of available propositions
for i_1 = 1 with step of 1 to length(I_1) do
    for i_2 = 1 with step of 1 to length(I_2{i_1}) do
        for i_3 = 1 with step of 1 to length(I_3{i_1}{i_2}) do
            k = k+1
            proposition_ind(k,:) = [i_1 i_2 i_3]
            % The indices at each level for a proposition
            proposition(k,:) =
            [I_1(i_1) I_2{i_1}(i_2) I_3{i_1}{i_2}(i_3)]
            % A proposition
            d(k,:) = sqrt(sum((Q - proposition(k,:)).^2))
            Distance between the proposition and query
        end
    end
end
[min_val min_index] = min(d) % Find the minimal distance
proposition(min_index,:) % The best proposition
[d_sorted d_sorted_ind] = sort(d,'ascend')
proposition(d_sorted_ind(1:5),:) % Top 5 propositions
```

Fig. 3. Pseudocode exemplifying how the CBRS with three levels works

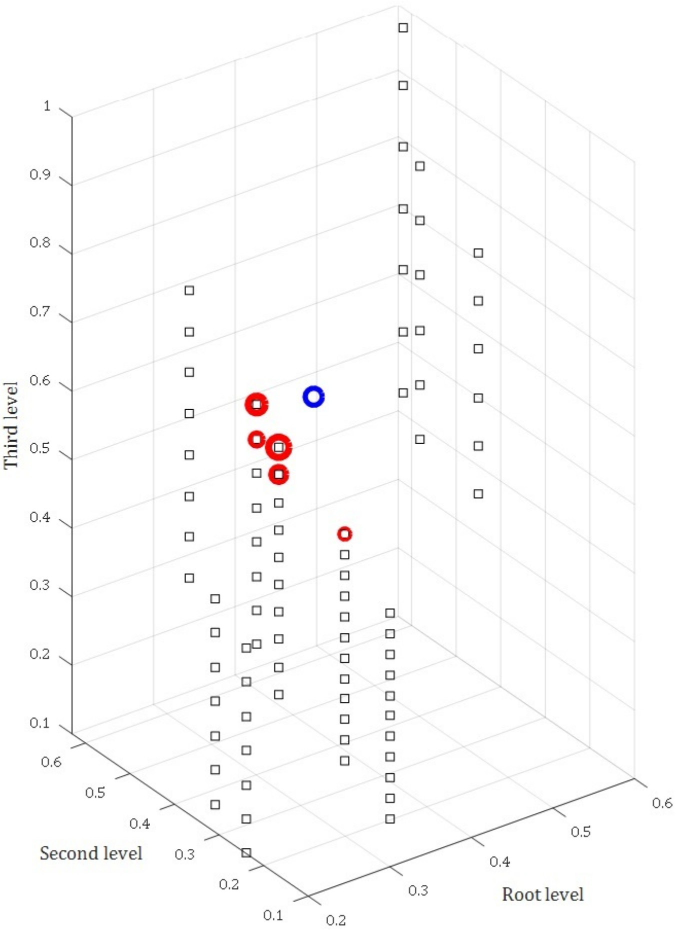


Fig. 4. A visualization of the toy CBRS example with 3 levels

VI. Discussion

Although the CBRS application side framework has a hierarchical structure, its fractal resemblance does not imply quickly growing item bunches and memory usage. There can be repeated items (at a level), so they are unified and thus the level becomes considerably narrower than in the "raw" hierarchy in Figure 1.

The description-to-value mapping should not be confused with feature encoding [33]. Whereas the latter is used to operate with objects (items), the mapping is intended to compare them [34–37].

Price constraints do not imply the less-than-or-equal-to sign (with respect to the user). The same concerns other non-descriptive items like volume, quantity, etc. This is done for supporting neutrality of the results (propositions) returned by the RS. This neutrality means that the RS will not search for eventually beneficial commodities for the user (the less-than-or-equal-to price condition), nor will it search for profitability of the e-trader (the greater-than-or-equal-to price condition). The RS will find the propositions closest to the user query features, among which the price is just one item. However, the importance of the price level (as well as other particular levels) can be strengthened by increasing its weight. Then the price item on its $[0;\ 1]$-scale will have a greater value. Surely, the weighting does not violate the neutrality.

In terms of our CBRS, it is useful to distinguish the customer from the user. The difference is not that much, but it still nonetheless exists. The user is anyone who can make queries but not necessarily buy. The customer is the user which eventually buys via the CBRS, or bought previously and thus can be called the customer even if the one is not buying now.

VII. Conclusion

We have suggested a novel CBRS with a purpose to improve commodity realization for both the e-trader and customer. A commodity has hierarchical-structure properties which are mapped to the respective numerical scales at each level. The numerical scales are normalized to the unit interval so that a query from a user as a potential customer and any possible proposition from the e-trader is a multidimensional point of a nonnegative unit hypercube put on the coordinate origin. The user can weight levels. The distance between the query and propositions are measured by the respective metric in the Euclidean arithmetic space. The best proposition is defined by the shortest distance. Top N propositions are defined by N shortest distances. The suggested system is independent of user experience and supports neutrality, i. e., the RS does not have a bias to the e-trader profit-pursuit behavior nor to user benefits.

Acknowledgment

The authors would like to thank the regional software company KeepSolid, Inc. (Odessa, Ukraine) for their assistance in leading discussions with e-commerce agents.

References

[1] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. Kantor, Eds. Boston, MA, USA: Springer, 2011, pp. 1–29. DOI: 10.1007/978-0-387-85820-3

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, Apr. 2005. DOI: 10.1109/TKDE.2005.99

[3] H. Khatter, S. Arif, U. Singh, S. Mathur, and S. Jain, "Product recommendation system for E-commerce using collaborative filtering and textual clustering," in *Proc. 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2021, pp. 612–618. DOI: 10.1109/ICIRCA51532.2021.9544753

[4] P. Jariha and S. K. Jain, "A state-of-the-art recommender systems: An overview on concepts, methodology and challenges," in *Proc. 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, India, 2018, pp. 1769-1774. DOI: 10.1109/ICICCT.2018.8473275

[5] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-Commerce Recommendation Applications," *Data Mining and Knowledge Discovery*, vol. 5, no. 1, pp. 115–153, Jan. 2001. DOI: 10.1023/A:1009804230409

[6] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decision Support Systems*, vol. 74, pp. 12–32, Jul. 2015. DOI: 10.1016/J.DSS.2015.03.008

[7] E. Wong and Y. Wei, "Customer online shopping experience data analytics: Integrated customer segmentation and customised services prediction model," *Int. J. Retail Distrib. Manag.*, vol. 46, no. 4, pp. 406–420, Apr. 2018. DOI: 10.1108/IJRDM-06-2017-0130

[8] M. Li, B. M. Dias, I. Jarman, W. El-Deredy, and P. J. G. Lisboa, "Grocery shopping recommendations based on basket-sensitive random walk," in *Proc. 15th ACM SIGKDD international conference on Knowledge discovery and data mining – KDD '09*, New York, NY, USA, 2009, pp. 1215–1224. DOI: 10.1145/1557019.1557150

[9] H. Abdollahpouri, B. Robin, and M. Bamshad, "Popularity bias in ranking and recommendation," in *Proc. 2019 AAAI/ACM Conference on AI, Ethics, and Society – AIES '19*, New York, NY, USA, 2019, pp. 529–530. DOI: 10.1145/3306618.3314309

[10] M. Behl, F. Smarra, and R. Mangharam, "DR-Advisor: A data-driven demand response recommender system," *Applied Energy*, vol. 170, pp. 30–46, May 2016. DOI: 10.1016/j.apenergy.2016.02.090

[11] D. Mcsherry, "Explanation in recommender systems," *Artificial Intelligence Review*, vol. 24, no. 2, pp. 179–197, Oct. 2005. DOI: 10.1007/s10462-005-4612-x

[12] K. Reka and T. N. Ravi, "Enhancing User Profile by Combining User and Item Based Collaborative Filtering," *International Journal of Scientific & Technology Research*, vol. 8, no. 9, pp. 1602–1606, Sep. 2019.

[13] M. Quadrana, P. Cremonesi, and D. Jannach, "Sequence-aware recommender systems," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–36, Jul. 2018. DOI: 10.1145/3190616

[14] M. D. Buhmann et al., "Recommender Systems," in *Encyclopedia of Machine Learning*, C. Sammut and G. Webb, Eds. Boston, MA, USA: Springer US, 2011, pp. 829–838. DOI: 10.1007/978-0-387-30164-8_705

[15] L. Martínez, L. G. Pérez, M. J. Barranco, and M. Espinilla, "A knowledge based recommender system based on consistent preference relations," in *Intelligent Decision and Policy Making Support Systems*, D. Ruan, F. Hardeman, and K. van der Meer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 93–111. DOI: 10.1007/978-3-540-78308-4_6

[16] P. Lops, D. Jannach, C. Musto, T. Bogers, and M. Koolen, "Trends in content-based recommendation: Preface to the special issue on Recommender systems based on rich item descriptions," *User Model. User-adapt Interact.*, vol. 29, no. 2, pp. 239–249, Mar. 2019. DOI: 10.1007/s11257-019-09231-w

[17] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proc. 2008 Eighth IEEE International Conference on Data Mining*, Pisa, Italy, 2008, pp. 263–272. DOI: 10.1109/ICDM.2008.22

[18] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, pp. 1–19, Oct. 2009. DOI: 10.1155/2009/421425

[19] M.-H. Park, J.-H. Hong, and S.-B. Cho, "Location-based recommendation system using Bayesian user's preference model in mobile devices," in *Ubiquitous Intelligence and Computing. UIC 2007. Lecture Notes in Computer Science, vol. 4611*, J. Indulska, J. Ma, L. T. Yang, T. Ungerer,

and J. Cao, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1130–1139. DOI: 10.1007/978-3-540-73549-6_110

[20] W. X. Zhao, S. Li, Y. He, L. Wang, J.-R. Wen, and X. Li, "Exploring demographic information in social media for product recommendation," *Knowledge and Information Systems*, vol. 49, no. 1, pp. 61–89, Oct. 2016. DOI: 10.1007/s10115-015-0897-5

[21] R. Burke, "Knowledge-based recommender systems," *Encyclopedia of library and information science*, vol. 69, no. 32, pp. 180–200, Dec. 2000. DOI: 10.1007/s10115-015-0897-5

[22] P. Bonhard, C. Harries, J. McCarthy, and M. A. Sasse, "Accounting for taste: Using profile similarity to improve recommender systems," in *Proc. SIGCHI conference on Human Factors in computing systems – CHI '06*, New York, NY, USA, 2006, pp. 1057–1066. DOI: 10.1145/1124772.1124930

[23] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Model. User-adapt Interact.*, vol. 12, no. 4, pp. 331–370, Nov. 2002. DOI: 10.1023/A:1021240730564

[24] F. S. Gohari and M. J. Tarokh, "Classification and Comparison of the Hybrid Collaborative Filtering Systems," *International Journal of Research*, vol. 6, no. 2, pp. 129–148, Mar. 2017. DOI: 10.22105/RIEJ.2017.49158

[25] T. Di Noia, R. Mirizzi, V. C. Ostuni, D. Romito, and M. Zanker, "Linked open data to support content-based recommender systems," in *Proc. 8th International Conference on Semantic Systems – I-SEMANTICS '12*, New York, NY, USA, 2012, pp. 1–8. DOI: 10.1145/2362499.2362501

[26] A. Felfernig and R. Burke, "Constraint-based recommender systems: Technologies and research issues," in *Proc. 10th international conference on Electronic commerce – ICEC '08*, New York, NY, USA, 2008, pp. 1–10. DOI: 10.1145/1409540.1409544

[27] B. Amini, R. Ibrahim, and M. S. Othman, "Discovering the impact of knowledge in recommender systems: A comparative study," *Int. J. Comput. Sci. Eng. Surv.*, vol. 2, no. 3, pp. 1–14, Aug. 2011. DOI: 10.5121/ijcses.2011.2301

[28] A. Felfernig, K. Isak, and T. Kruggel, "Testing Knowledge-based Recommender Systems," *Oesterreichische Gesellschaft fuer Artificial Intelligence*, vol. 24, no. 4, pp. 12–18, Jan. 2005.

[29] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. Kantor, Eds. Boston, MA, USA: Springer, 2011, pp. 187–215. DOI: 10.1007/978-0-387-85820-3_6

[30] L. Guo, K. Luan, X. Zheng, and J. Qian, "A service recommendation method based on requirements for the cloud environment," *J. Control Sci. Eng.*, vol. 2021, pp. 1–11, 2021. DOI: 10.1155/2021/6669798

[31] R. Palit and R. Chatterjee, "Recommender system using K-nearest neighbors and singular value decomposition algorithms: A hybrid approach," in *Advances in Intelligent Systems and Computing*, H. Das, P. Pattnaik, S. Rautaray, and K. C. Li, Eds. Singapore: Springer Singapore, 2020, pp. 497–504. DOI: 10.1007/978-981-15-2414-1_50

[32] F. Moodi and H. Saadatfar, "An improved K-means algorithm for big data," *IET Software*, pp. 1–12, Jun. 2021. DOI: 10.1049/sfw2.12032

[33] U. I. Haq, I. Gondal, P. Vamplew, and S. Brown, "Categorical features transformation with compact one-hot encoder for fraud detection in distributed environment," in *Communications in Computer and Information Science*, R. Islam et al., Ed. Singapore: Springer Singapore, 2019, pp. 69–80. DOI: 10.1007/978-981-13-6661-1_6

[34] P. Cerda and G. Varoquaux, "Encoding high-cardinality string categorical variables," *IEEE Trans. Knowl. Data Eng.*, May 2020. DOI: 10.1109/TKDE.2020.2992529

[35] J. T. Hancock and T. M. Khoshgoftaar, "Survey on categorical data for neural networks," *J. Big Data*, vol. 28, no. 7, pp. 1–41, Apr. 2020. DOI: 10.1186/s40537-020-00305-w

[36] D. Mezzogori and F. Zammori, "An entity embeddings deep learning approach for demand forecast of highly differentiated products," *Procedia Manuf.*, vol. 39, pp. 1793–1800, 2019. DOI: 10.1016/j.promfg.2020.01.260

[37] B. Schifferer, C. Deotte, and E. Oldridge, "Tutorial: Feature Engineering for Recommender Systems," in *Proc. Fourteenth ACM Conference on Recommender Systems*, New York, NY, USA, 2020, pp. 754–755. DOI: 10.1145/3383313.3411543

Hanna Yehoshyna is an Associate Professor at the Department of Information Technologies of the Odessa Polytechnic State University. She received the Degree of Candidate of Technical Science in Artificial Intelligence Systems and Tools in 2009. The status of Associate Professor was received in 2011. She is a participant of the Data Science Program between the Faculty of Computer Science and Languages of Hochschule Anhalt University of Applied Sciences and the State University of Intelligent Technologies and Telecommunications for the realization of the joint Double Degree Master Program. Her research interests include recommender systems, machine learning and big data, natural language processing.

Vadim Romanuke is a Professor at the Department of Technical Cybernetics and Information Technologies of the Odessa National Maritime University. He received the Degree of Candidate of Technical Sciences in Mathematical Modeling and Computational Methods in 2006. The degree of Doctor of Technical Sciences in Mathematical Modeling and Computational Methods was received in 2014. In 2016, he received the status of a Full Professor. His current research interests concern decision making, statistical approximation, game theory, semantic image segmentation, software optimization.