Revista Romana de Interactiune Om-Calculator 9(3) 2016, 255-268

Recommendation Technique for the "Cold-Start" Problem

Mihaela Colhon[1], Adrian Iftene[2]

[1] University of Craiova, Department of Computer Science
Street Alexandru Ioan Cuza, Nr. 13, 200585, Craiova
E-mail: mcolhon@inf.ucv.ro

[2] Alexandru Ioan Cuza University of Iaşi, Faculty of Computer Science
Blvd. Carol I, no. 11, Iaşi
E-mail: adiftene@inf.uaic.ro

Abstract. Usually, users that benefit from a Recommender System outputs only get a list of items that the system assumes best match their needs, without having any clue regarding how the system managed to figure out what they like. In this paper, we propose a mechanism that generates content-based recommendations organized on levels of similarities with the selected product in order to let the user decide about what similarity degree (s)he wants to explore. We demonstrate empirically that our proposed mechanism can ensure good performance for a recommendation technique under the cold-start conditions.

Keywords: human-computer interaction, recommender system, content-based filtering, rough set theory.

1. Introduction

One of the basic concepts that govern our world is information, especially if we talk about the "world wide web". Lack of information can generate serious damage to any commercial, cultural, scientific or social organization with effects varying from unhappy clients or users, to smaller market shares, bankruptcy and so on. Too much information poses other problems like not knowing what piece of information can have more influence and which one is irrelevant to a certain situation, lack of structure that does not allow a proper access to data that is manipulated. In both cases, the outcome is far from what any organization wants. The answer to some of these problems lays in a better, more efficient use of information, and Recommender Systems seek to solve the issues raised by the tremendous amount of data available today.

Where do we see Recommender System? The answer is: "everywhere"; we daily meet websites where recommendation systems play an important role behind the scenes.

On each internet browsing session, we are inundated with lots of information which generate choices and options: what news to read? what movie to rent? what stock to buy? With the large amount of information into different web catalogues comes the difficulty for potential users of thos web repositories to use their standard search and browsing facilities. The sizes of these decision domains are frequently massive: Youtube is a good example for information overload as it is quoted with roughly 100 hours of videos uploaded per minute. Other classic examples are Netflix, a website of over 17,000 movies or videos in its selection or Amazon.com with over 410,000 titles in its Kindle store alone (Johnson and Divya, 2016). In this huge area of various data, supporting knowledge discovery is a significant challenge. Even simple decisions - what movie should I see this weekend? - can be difficult without prior knowledge of the candidates. And yet, how do we still get what we want or what we like? Simple, by means of Recommender Systems.

Recommender Systems represent a powerful method for guiding users in a personalized way in order to discover interesting objects in a big space of possible options. The main idea is to know what users want, even before they express their needs, and to offer them what they want, at the right time.

Another common situation for a modern recommendation system is a website with which a user can interact, usually through the so-called "office assistant" (see Figure 1). This kind of thinking applies to online stores, social networks, online content generators, or any other domain that involves supply and demand, having a representation in the virtual world. Obviously, this process depends on the particular human/computer interaction supported by the system when the items are presented, compared, and explained (Ricci et al., 2011).

Rough set theory was used in order to generate decision rules between an initial product attributes and other items attributes that might also be of interest (Su et al., 2010). This theory was initially developed in order to increase the likelihood of correct predictions by identifying and removing redundant variables (Li and Cercone, 2005). In this paper we propose a recommendation technique that makes use of rough set theory in order to generate attribute-based selections.

Figure 4. Example of an Office Assistant: the Apple consultant of www.icenter.ro

The article is organized as follows: in Section 2 we give an overview of the existing recommendation techniques by emphasizing their advantages upon the so-called “cold-start” problem. In Section 3 we give a possible solution for this issue by proposing a content-based recommendation approach which computes similarities measures between items attributes using rough set theory. The article ends with final conclusions and our proposal for future improvement.

2. Recommendation Techniques and the “Cold-Start” Problem

The different recommendation domain particularities and the corresponding users’ needs show that recommendation is not a one-size-fits-all problem. Specific tasks, information needs, and item domains represent unique problems for recommenders, and design and evaluation of recommenders needs to be done based on the user tasks to be supported (Ekstrand et al., 2011).

A good recommender system can overcome the so-called “cold-start” problem which appears when there is no data available about the user’s preferences. This means that the “cold-start” problem appears when somebody is using for the first time a software product. It has been observed that most of the times, a user renounces the use of a particular software product if his first-time usage does not meet his expectations. So the “cold-start” problem is a major one, and if it is not properly resolved it could cause loss of customers for software developers or e-commerce sites.

Any approach implemented by a recommender system must generate predictions about the user's interest without asking for more data. The implemented recommendation techniques are designed in order to improve the interaction with the user by giving what he/she needs in no time, with minimum effort from the user part. The recommendation techniques developed until now can be grouped into the following main paradigms (Bancu et al., 2012): collaborative filtering and content-based filtering.

Perhaps the most well-known recommendation technique is *collaborative filtering* (Schafer et al., 2007) used in the early stages of Recommender Systems. The collaborative filtering approach builds a model using knowledge:

- based on the user's past behaviour, known as *item-based collaborative filtering*. This knowledge could be items previously purchased or selected and/or numerical ratings given to those items.
- based on decisions made by similar users, known as *user-based collaborative filtering*. It is a well-known fact that if one item from a website catalogue was purchased by a female, then it can be suggested to a female user because it is possible that she could do the same choice.

When the model is completed, the recommender system uses it to predict items (or ratings for items) that the user may have an interest in. One classical example of this technique is the online news recommender GroupLens (Resnick et al., 1994). Ex: if Alice and Bob both rated a lot of news articles similarly in the past, then perhaps Alice will like the next article that Bob likes as well. But if Alice rates products that address women's needs, then perhaps, these will not be good recommendations for Bob.

Collaborative filtering does not rely on the content of items, but instead requires users to provide ratings on a common set of items, so that future predictions can be made from the user rating overlap. As a result, collaborative filtering recommenders commonly suffer from little user rating overlap at the early stage, thus this approach is not appropriate for the "cold-start" problem. Opposite to collaborative approaches, the content-based filtering can help to overcome the unknown issue by inferring similarities based on new data characteristics.

Content-based filtering approaches use content modelling of the product and user preferences. While the collaborative filtering makes recommendations using data from similar users, in content-based filtering the user preferences stand out (Alonso et al., 2010). This kind of filtering is often used in domains where extensive textual content is available, such as for recommending websites, books, and news articles. For example, LIBRA application is capable of making book recommendations from various websites using the Naive-Bayes text categorizing algorithm (Mooney and Roy, 2000).

By considering each user independent among the others and by putting accent on the profile of each item or product, content-based techniques are considered as a solution for the issue known as “cold-start” problem.

The collaborative and content-based filtering approaches are often combined, in the so-called Hybrid Recommender Systems (Burke, 2007).

Personalized recommender systems can be a promising solution for the information overload problem in social network sites. By modelling individual users’ interests and proactively making recommendations, personalized recommender systems have provided solutions to similar problems in a wide variety of domains (Adomavicius et al., 2005). Such recommenders are particularly suitable for social network sites because they are personalized and effortless for users. By personalized, we mean that the help provided by the recommenders is usually inherently personalized to individual users, and therefore match the ego-centric experiences of social network sites. By effortless, we mean that recommenders can help proactively, without requiring any knowledge, skill or effort from the users. In contrast, other solutions, such as browsing interfaces and search engines, usually require users’ participation and certain level of skills.

2.1 The proposed approach. Recommendations grouped on Similarity Classes

The proposed method is intended to be a solution to the “cold-start” problem encountered in the majority of the recommendation systems. For this reason, we sought to define a recommendation technique designed to be capable of making meaningful suggestions even if there is no previous activity (which means, previous selections) from the user part. The proposed technique makes use of a content-based recommendation approach which computes similarities measures between the items attributes using rough set theory. With this approach we intend to construct a Recommender System (shortly, RS) capable to work with new users, obtaining in this manner a RS with no "cold-start" problem.

The idea of this study appeared after we had experienced some e-commerce sites in a "cold-start" scenario: sites that had never been used by any of the authors, explored in non-register sessions. Analyzing the received recommendations3, we generally observed the same recommendations even when the selected products differ greatly (see Figures 2 and 3).

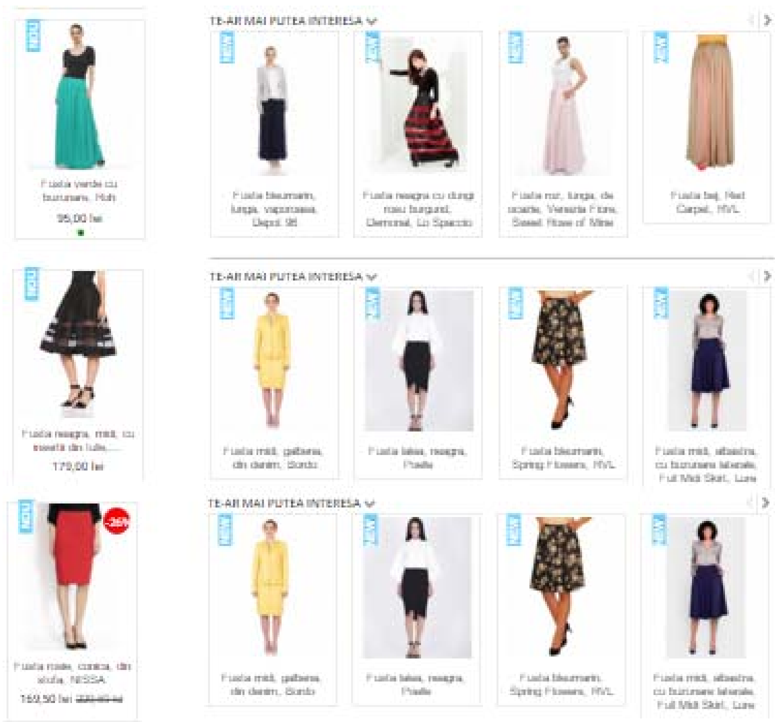

Figure 5. A set of chosen products (left) and their recommendations (right): same recommendations for different products

[3] For this study we did not consider recommendations of the form "other users also liked/bought/viewed" as we were interested in developing a content-based recommendation technique.

Figure 6. A set of chosen products (left) and their recommendations (right): recommendations not very related with the chosen item

Basic Rough Set Theory

Rough set theory as proposed by Pawlak and Slowinski in 1994 (Pawlak and Slowinski, 1994) is a powerful tool for constructing a multi-attribute decision analysis. This approach provides efficient algorithms for discovering hidden patterns in data, minimal sets of data, evaluating significance of data and generating sets of decision rules from data (Pawlak et al., 2005; Pawlak, 1999).

In rough set theory, the knowledge representation has the form of an information system defined as a finite two-dimensional table where the rows represent the objects and columns represent the objects’ attributes. It results that in information systems, data has the form of attribute-value entries. Usually, an information system is defined as a 4-tuple $S = (U, A, V, f)$, such that U is a finite set of objects, A is a finite set of attributes, V is a finite set of values for the attributes of the set A and f: $U \times A \rightarrow V$ is called the information function, where $f(u, a) = v$ maps each pair $(u, a) \in U \times A$ to the corresponding value $v \in V$.

For every $B \subseteq A$, $IND(B)$ denotes an equivalence relation on the set U of objects and represents the set of objects which are indiscernible by the attributes of B:

$IND(B) = \{(x, y) \in U \times U \mid \forall a \in B: f(x, a) = f(y, a)\}$

For every $B \subseteq A$ and $x \in U$, $[x]_B$ denotes the *equivalence class* of the object x with respect to the attributes of the set B:

$[x]_B = \{y \in U \mid \forall a \in B: f(x, a) = f(y, a)\}$

If we consider $B \subseteq A$ - a subset of attributes from A and $X \subseteq U$ - a subset of objects of the universe U then, by using the objects' equivalence classes, two approximation of the set X can be constructed:

- the lower approximation

$B_*X = \{x \in U \mid [x]_B \subseteq X\}$

We say that an element x belongs to the lower approximation if all the objects equivalent with x are included in the set X.

- the upper approximation

$B^*X = \{x \in U \mid [x]_B \cap X \neq \emptyset\}$

From the definition of the upper approximation results that set B^*X contains all the objects of the universe U that are equivalent (from the point of view of the attributes of B) with at least one object of the set X.

By definition we have:

$B_*X \subseteq X \subseteq B^*X$

These approximations can decrease or increase the number of the elements of the set X as the upper approximation can contain (besides the elements of X) other elements that are equivalent with some of X's elements by taking into account only the attributes of the set B, while the lower approximation filters the elements of X by means of the attributes considered in the set B.

If the difference between the upper approximation and the lower approximation of the set X is non-empty, then the set X is considered rough or vague with respect to the attributed of B, otherwise is taken as crisp set.

- As it will be presented in the following section, we can apply the rough set theory in order to automatically find, based on the attributes of a selected product or item, other products/items similar with the chosen one with the intended scope of creating recommendation lists. In a “cold-start” situation, this means that the recommended products must share the same characteristic for the common attributes with the chosen item.

- More precisely, if we consider the set *X* as the set consisting of a single element: the chosen item or product, and the set *A* as the set of all attributes that characterize the element of *X*, we can define a mechanism for constructing a decreasing similarity sequence of items with respect to the element of *X*. The items found in this manner can be given as recommendations to the user grouped on several similarity classes starting with the “very similar” class and ending with the “week similar” class in order to offer the user the possibility to decide which level of similarity wants to explore.

Pyramid of Similarity Classes

From a Recommender System point of view, only the upper approximation is useful, because it can bring into attention other items than the chosen ones (the set *X*). Our proposed recommendation technique creates levels of similarity by starting with the elements of set *X* (first level) and considering for the next levels the upper approximations of the set *X* with respect to a decreasing sequence of subsets of *A*. In this manner we obtain a pyramid of items grouped on classes of similarity with its top (see Figure 4).

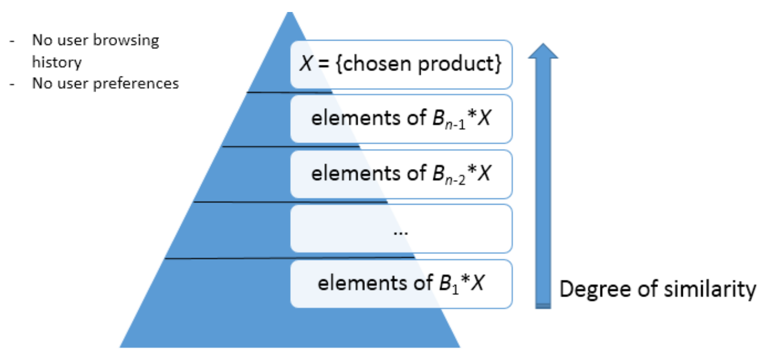

Figure 7. Pyramid of Similarity Classes

If we consider the set A as a set of n elements (attributes) arranged in a descending order upon their significance from a user point of view[4]:

$A = \{a_1, .., a_n\}$

then the similarity classes are determined as follows:
- the top of the pyramid consists of the elements of the set X
- the first recommendations level (the second level of the similarity pyramid) consists of all the elements of the upper approximations $B_{n-1}{*}X$ considered upon n-1 attributes as $B_{n-1}= \{a_1, \ldots, a_{n-1}\} \subseteq A$. We obtain that $X \subseteq B_{n-1}{*}X$.
- the last recommendations level, or the pyramid base, consists of all the elements of the upper approximations $B_1{*}X$, for $B_1 = \{a_1\}$. Obviously we have:

$X \subseteq B_{n-1}{*}X \subseteq \ldots \subseteq B_1{*}X.$

This property ensures that all the similarity classes $B_i{*}X$ are non-empty sets. For the worst case which corresponds to the situation where no recommendation can be generated we have:

$\{x\} = X = B_{n-1}{*}X = \ldots = B_1{*}X$

Remark 1. By considering the attributes of the set A in a descending order based on their significance, we ensure that the first attribute a_1 – which is the most significant one from the user's point of view – is taken into account for all the generated classes of similarity. For example, if we consider this first attribute as the representation of the chosen item's category, we ensure that all the resulted recommendations share the same category with the product selected by the user. We propose this approach taking into account a common sense opinion: if a user selects a TV product it is very likely that he/she is interested in a TV product and not in a, let's say washing machine, even if this last one shares the same brand as the

[4] Depending on the characteristic of the universe U, we can consider for the first attribute a_1, the items category and for the last attribute a_n the most unimportant attribute, for example, color or size.

chosen TV product.

Remark 2. We obtain that the objects of kth level are much similar with the objects of the set X (the first level) than the objects of (k+1)th level but fewer than the last ones.

Recommendation Technique

The method presented above can be implemented in a recommendation system if we consider the following representations:

1. we note by x the product chosen by the user;
2. we consider the set U as the set of all products of the current database or electronic catalogue;
3. because we apply this recommendation technique in a “cold-start” scenario, the set X has to include a single piece of information: the chosen product. We obtain $\{x\} = X \subseteq U$;
4. we consider the set A as the set of all the attributes that characterize the products of the universe U arranged in a decreasing order upon their significance degree value. We note by n the cardinality of the set A, that is $|A| = n$.
5. the first similarity class $B_{n-1}*X$, that is the second level of similarity pyramid, corresponds to the set $B_{n-1} \subseteq A$.
6. recommendations process stops with the last similarity class consisting of the elements B_1*X and which corresponds to the set of attributes $B_1 \subseteq A$.

The way the items are chosen for recommendation ensures close relating and gradually similarities with the selected item as it is illustrated in Figure 5. The recommendations explore all the items inside the category of x and, by creating a decreasing sequence of attributes’ sets, select the items that share the same attributes-values with the element of $x \in X$.

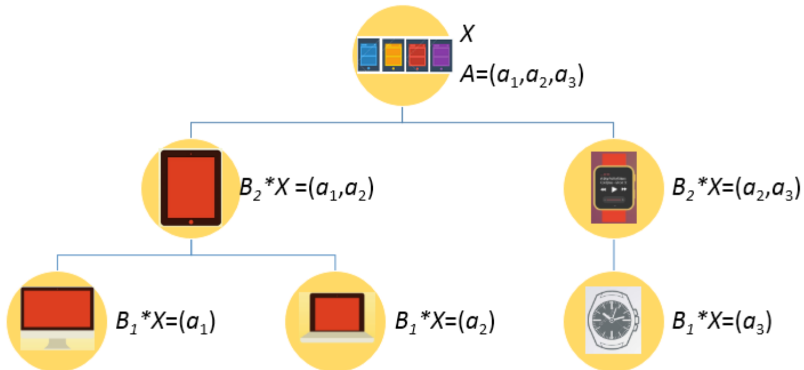

Figure 8. Example of a pyramid of recommendations

Using a similarity matrix (Table 1) we can illustrate the similarity degrees the classes B_i* X, $i = \overline{1, n-1}$ share with the target object x ∈ X.

Let us consider two products:

- the chosen product x ∈ X
- a recommended product y ∈ B_i* X, $i = \overline{1, n-1}$.

If we note by $sim_{n-1\times n}$ the similarity matrix, then the element sim(x,y) will represent the similarity degree between the products x and y.

A very common approach is to consider items as document vectors of an n-dimensional space. Therefore, their similarity is computed as the cosine of the angle that they form (Ricci et al., 2011). As a direct result, the value of the element sim(x,y) determined with the classical cosine similarity among binary vectors is:

$$sim(x,y) = \frac{\#_{attr}(x,y)}{\sqrt{\#_{attr}(x)}\sqrt{\#_{attr}(y)}}$$

where by $\#_{attr}(x,y)$ we note the number of common attributes that share same values across the products x and y and by $\#_{attr}(x)$ – the number of attributes that belong to the item x.

Table 2. The similarity matrix

Similarity	The chosen product, x

<table>
<tr><td>Classes</td><td colspan="6">$X = \{x\}$</td></tr>
<tr><td>1st recommendations level, $B_{n-1}* X$</td><td colspan="5">$(\forall)\ y \in B_{n-1}*X$, $\#(x,y)= n-1$</td><td></td></tr>
<tr><td>2nd recommendations level, $B_{n-2}* X$</td><td colspan="4">$(\forall)\ y \in B_{n-2}*X$, $\#(x,y)= n-2$</td><td></td><td></td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr>
<tr><td>last recommendations level, $B_1* X$</td><td>$(\forall)\ y \in B_1*X$, $\#(x,y)=1$</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

3. Conclusion

In this paper, we proposed a recommendation technique that makes use of rough set theory in order to create a pyramid of similarities with a chosen product. In this manner, we try to offer a solution to the “cold-start” problem as no information about the users’ previous experiences are considered. Also, the proposed technique organizes the recommended items in a gradual manner in order to let the user to decide which level of similarity wants to explore.

As future work we consider creating an evaluation framework in order to validate our proposed technique.

References

Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A. (2005) Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. ACM Transactions on Information Systems, 23, 1, 103–145.

Alonso, R., P. Bramsen, H. Li, (2010) Incremental user modeling with heterogeneous user behaviors. International conference on knowledge management and information sharing, KMIS 2010, 129-134.

Bancu, C., Dagadita, M., Dascalu, M., Dobre, C., Trausan-Matu, S. Florea, A. M. (2012) ARSYS -- Article Recommender System, In 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 349-355

Burke, R. (2007) Hybrid web recommender systems. In: The Adaptive Web, Springer Berlin / Heidelberg, 377–408.

Ekstrand, M D., Riedl, J. T., Konstan, J. A. (2011) Collaborative Filtering Recommender Systems, In Foundations and Trends in Human–Computer Interaction 4, 2, 81-173.

Johnson, A., Divya, R. A (2016) Survey On Efficient Service Recommendation On Large

Data Clusters. In *IJCSET* 6, 2, 78-81.

Li, J., Cercone, N. (2005) A rough set based model to rank the importance of association rules. In International Workshop on Rough Sets, Fuzzy Sets, Data Mining and Granular-Soft Computing 2005, Springer Berlin Heidelberg, 109-118.

Mihăescu, M. C., Tacu, M. G., Burdescu, D. D. (2014) Use Case of Cognitive and HCI Analysis for an E-Learning Tool. In Informatica (Slovenia), 38, 3, ISSN 0350-5596, 273-280.

Mooney, R.J., Roy, L. (2000) Content-Based Book Recommending Using Learning for Text Categorization. In: Proceedings of the 5th ACM Conference on Digital Libraries, ACM Press, New York, US, San Antonio, US, 195–204.

Pawlak, Z., Polkowski, L., Skowron, A. (2005) Rough Sets: An approach to vagueness. In Encyclopedia of Database Technologies and Applications, Hershey, PA, Idea Group Inc, 575-580.

Pawlak, Z., Slowinski, R. (1994) Rough set approach to multi-attribute decision analysis, In *European J. of OR.* 72, 3, 443-459.

Pawlak, Z. (1999) Data Mining - a Rough Set Perspective. In Proceedings of Methodologies for Knowledge Discovery and Data Mining: Third Pacific-Asia Conference, PAKDD-99, Springer Berlin Heidelberg, 3-12.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J. (1994) GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proc. of CSCW, Chapel Hill, NC, 175-186.

Ricci, F., Rokach, L., Shapira, B. (2011) Recommender Systems Handbook, Springer Science+Business Media, LLC

Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S. (2007) Collaborative filtering recommender systems. In: The Adaptive Web, Springer Berlin / Heidelberg, 291–324.

Su, J.H., Wang, B.W., Hsiao, C.Y., Tseng, V.S. (2010) Personalized rough-set-based recommendation by integrating multiple contents and collaborative information. Information Sciences, vol. 180(1), 113-131