A Case-Based Recommendation Approach for Market Basket Data

Anna Gatzioura, Miquel Sànchez-Marrè

Computer Science Department, Universitat Politècnica de Catalunya · BarcelonaTech

Abstract

In recent years, Recommender Systems have become an important part of various applications, supporting both customers and providers in their decision making processes. However, these systems still have to overcome limitations that reduce their performance, like recommendations' overspecialization, cold start and difficulties when items with unequal probability distribution appear or recommendations for sets of items are asked. A novel approach, addressing the above issues through a case-based recommendation methodology, is presented in this article. The scope of our approach is to generate meaningful recommendations based on items' co-occurring patterns and to provide more insight into customer buying habits. In contrast to current recommendation techniques that recommend items based on users' ratings or history, and to most case-based item recommenders that evaluate items' similarities, the implemented recommender uses a hierarchical model for the items and searches for similar sets of items, in order to recommend those that are most likely to satisfy a user.

Keywords

Recommender Systems, Case-Based Reasoning, User Preference Learning, Set of Items Recommendations, Items' Co-occurring Patterns, Market Basket Analysis

Introduction

Recommender Systems have been used in numerous domains in order to support both users in handling the information overload and finding adequate items to cover their needs and providers in identifying users' needs and increasing the amount and diversity of the items they sell.[1]

The widely used recommendation techniques tend to recommend *similar items to those liked by a user* in the past, or *items* that *similar users* liked, based on the hypothesis that users have a stable behavior over time, and that similar users share similar tastes. Most of these methodologies show limited performance when new users or less popular items appear (*cold start*), while others may lead to *overspecialization* of the recommended items. In addition, as they tend to ignore the underlying structure of users' preferences, and do not evaluate the hidden concepts under which items are selected, their accuracy decreases when trading items with unequal probability distributions. Previous research, mainly held in the fields of market research and customer behavior analysis, has revealed the existence of underlying patterns that determine the structure of users' market baskets.

The *market basket* is defined as the set of items bought by one customer in a single visit to a store, while *Market Basket Analysis, (MBA)*, refers to the search of meaningful associations in customer purchase data, aiming to discover the patterns that define the composition of these baskets.[2] The market basket domain is characterized by the existence of a large number of items and transactions that consist of co-occurring items. In such cases, more than simply predicting whether a single item will be or not liked by a user, there is the intention to capture the presence or absence of an item within a concrete buying concept, and become able to recommend *complementary items*. Most of the current recommendation methodologies do not take into account these co-occurrences while the *association rules* methodology that has been used as a basis for recommendations in such cases, comes up with several limitations, like computational and evaluation difficulties when applied to large datasets.[3]

Therefore, there is a necessity of developing *intelligent recommendation methodologies,* being able to generate valuable item recommendations based on users' habits and purchase patterns, while overcoming the drawbacks of the current recommendation techniques. Given that a user has already selected some items, the intention of these systems would be to propose the items that are most likely to complete a user's experience.

In the present article, we present an analysis and recommendation approach for situations where a user's experience *depends on the set of items selected together* more than on the standalone attributes of each item, like in the market basket analysis domain. We think of Case-Based Reasoning as being particularly appropriate, as through the adequate cases' representation the sets of items selected together may be modeled and compared. In addition the performance of a case-based reasoner benefits from the existence of a large amount of data like in the market basket analysis domain. The implemented methodology uses a hierarchical categorization of the items that the transactions consist of, in order to find out the structure of users' preferences and generate valuable recommendations. This system, by evaluating sets of selected items, is able to identify the items that are selected within concrete concepts and recommend them in similar situations to new or existing users. As it can be seen in the experimentation section, the developed recommender performs better than the state of the art techniques for similar problems.

Recommender Systems & Recommendation Techniques

Recommender Systems (RSs) are software tools and techniques for information retrieval and filtering, aiming to provide meaningful and effective item recommendations to the *active user.*[4] The term *item* refers to the type of entity (ex: product, service, information, etc.) being recommended and depends on the application area and on the objectives of the specific system. Recommender systems usually generate a set of (top-N) items expected to be liked by the user, or intend to predict whether a specific item will be or not of interest to her/him.

The widely used recommendation methodologies in commercial applications can be mainly divided into *Collaborative Filtering* and *Content-Based*. Their main differences can be seen in Figure 1.

In *Collaborative Filtering (CF)* recommendation techniques, items among those liked by similar users ("*neighbors*") are recommended to the active user. A user profile is built of the items that have been highly rated by the user, thus the similarity of users' tastes is deduced from their previous ratings. Although widely used in commercial applications, *collaborative RSs* still have to overcome *scalability* and *cold-start* problems that limit their performance.[6]

On the other hand, in *Content-Based (CB) RSs* user profiles are built from the characteristics of the items that a user has rated highly, and the items that she/he has not tried yet are compared to them. The items with the higher estimated possibility of being liked are then recommended.[4, 5] As CB techniques rely on more specific information about users and items they are able to recommend new items. However, they have to overcome the recommendations' limited diversity and possible *overspecialization*.
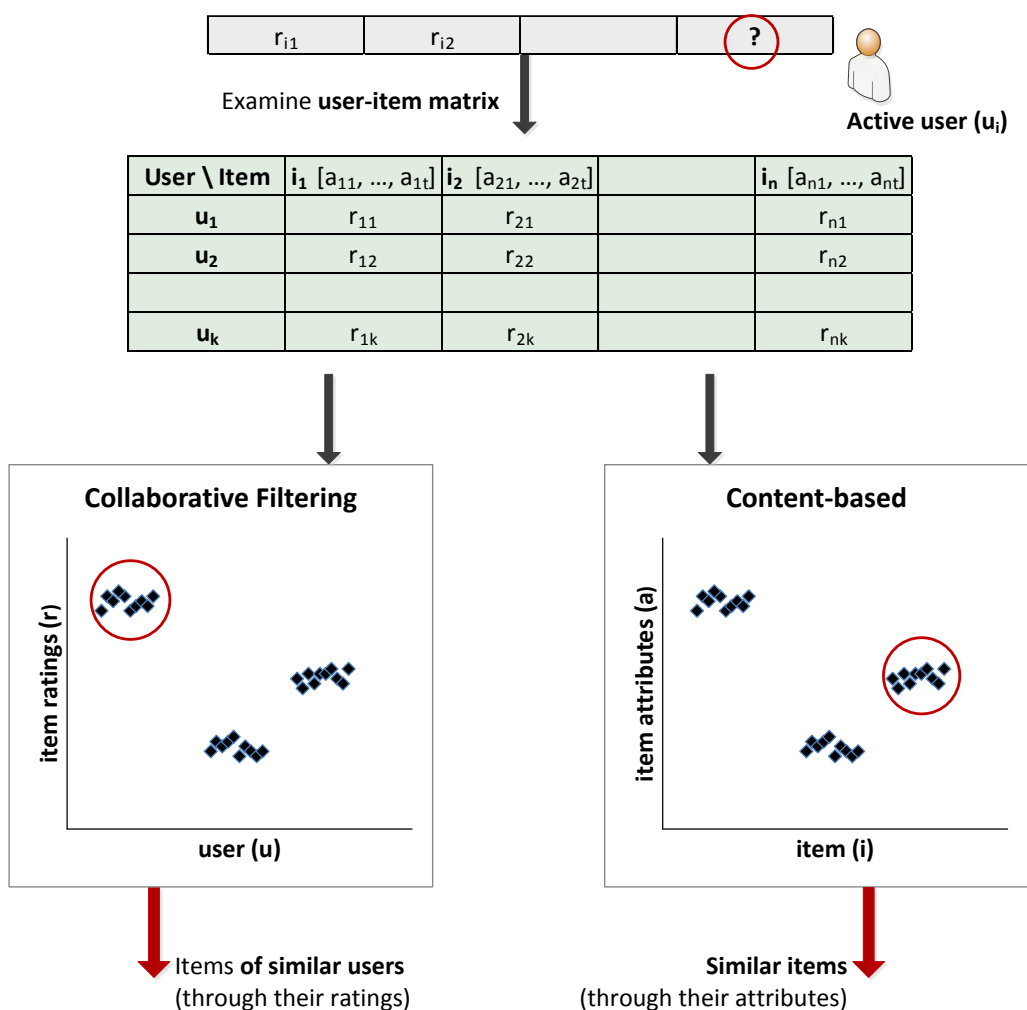

User \ Item	i_1 $[a_{11}, ..., a_{1t}]$	i_2 $[a_{21}, ..., a_{2t}]$		i_n $[a_{n1}, ..., a_{nt}]$
u_1	r_{11}	r_{21}		r_{n1}
u_2	r_{12}	r_{22}		r_{n2}
u_k	r_{1k}	r_{2k}		r_{nk}

Figure 1: *Collaborative Filtering and Content–based recommendations*

Various hybrid approaches have been proposed intending to leverage the strengths of both the above techniques, overcome their current limitations and improve the recommendation accuracy.[1, 5] The basic recommendation techniques identified in literature[1] are listed below,

- ➢ Collaborative Filtering (memory-based, model-based)
- ➢ Content-based Filtering (Neural Networks, probabilistic models, Naïve Bayes classifier)
- ➢ Knowledge-based (case-based, constraint-based)
- ➢ Utility-based
- ➢ Rule-based
- ➢ Demographic
- ➢ Other (context-aware, semantically enhanced, hybrid)

Knowledge-based and especially *case-based recommenders* have mainly emerged as an alternative to CF recommenders intending to overcome their shortcomings while efficiently handling the existing information overload. Case-based recommenders implement a type of content-based recommendation that relies on a structured representation of *cases,* usually as sets of well-defined characteristics with their values.[8] These systems generally recommend items similar to those that the active user has described in his/her request. (See Related Work in Case-Based Reasoning)

Rule-based recommendation techniques generate item recommendations based on a set of rules extracted from a data corpus. *Association rules (ARs) mining* refers to the observation of transactions aiming to discover interesting hidden patterns and frequent associations among the existing items, usually expressed in the form of "if-then" statements.[3]

Recently, *semantic analysis, latent factors* and *probabilistic topic models*, arising from natural language processing, have been successfully applied to information retrieval and then to RSs, especially for tag recommendations. The basic idea behind them is that topics are sets of words from a given vocabulary and documents are formed as probability distributions over topics. These techniques show higher accuracy than rule-based techniques and are able to better handle sparcity problems. [13]

Furthermore, due to the evolution of mobile devices and the use of recommender systems in applications highly depended on the context (location, time, weather, movement state, emotional state etc.) within which are performed, context-aware recommenders have received increased attention. These systems more than grouping users or items based on their ratings or characteristics, group users or items associated with similar context information. [9]

Related Work in Case-Based Reasoning & Case-based Recommenders

Case-Based Reasoning (CBR) is a problem solving paradigm closely related to the human way of reasoning and acting in everyday situations when facing new problems. CBR uses old experiences in order to solve new problems, based on the following sentence, known as the CBR assumption, *"Similar problems have similar solutions"*.

A situation experienced in a way that it has been captured and learnt, is referred to as *past/previous case* and is stored in the *case base*. A new situation asking for solution forms the description of a *new/target case*. An important part of the CBR methodology is its learning ability that comes as a natural result of its problem solving process, as the case base is updated

each time a new experience is obtained. This knowledge may be reused when needed without implementing the whole process from scratch, or highlight a methodology that should be avoided in a similar situation. Therefore case-based reasoners are able to improve their problem solving performance over time.[10]

The CBR solving and learning process can be described as a cyclical process comprising four processes known as the CBR cycle (Figure 2) or "the four REs":[11]

Retrieve: the most relevant cases.

Reuse: the knowledge provided to the new problem.

Revise: the solution obtained.

Retain: the parts of the new solution that are likely to be used for future purposes.

Additionally to the knowledge obtained by the previous cases, there is also domain dependent knowledge supporting the CBR process.
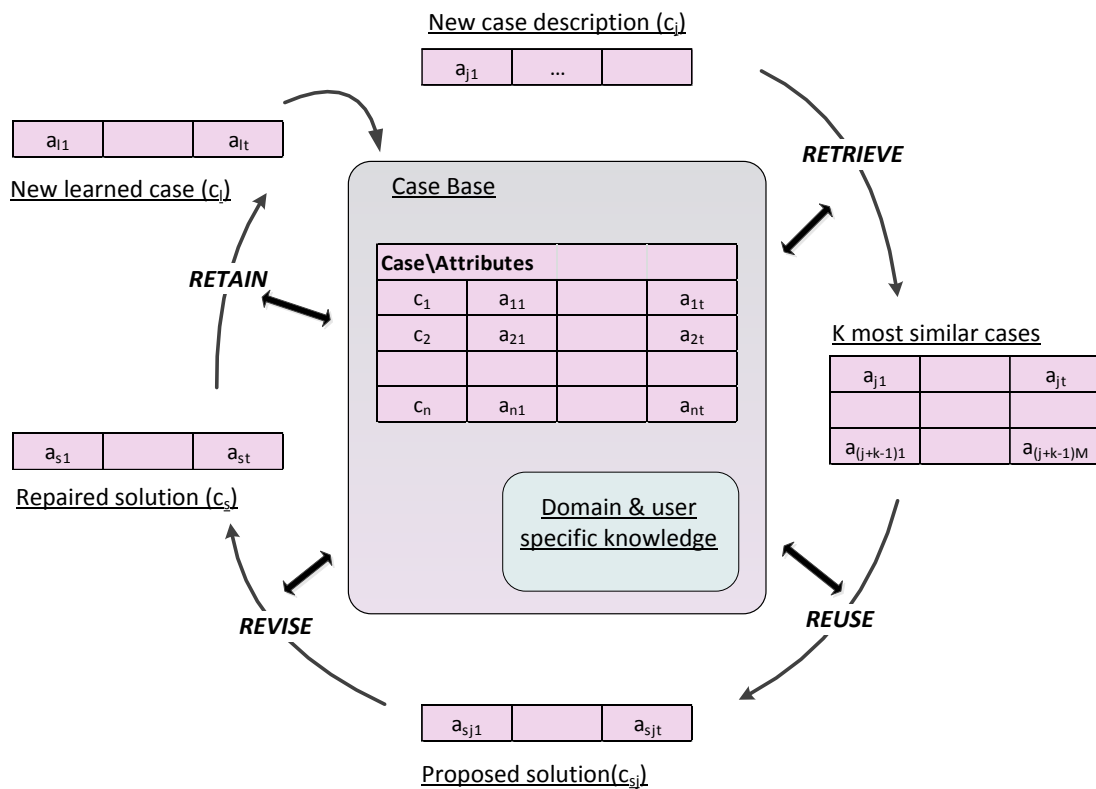

Figure 2: *CBR Cycle*

Cases can be viewed as composed of two parts, the *problem description* and the *problem solution*. Thus, cases with similar descriptions are retrieved and their solutions are adapted to the needs of the target problem.[10] The existence of a common and structured representation of the treated items enables case-based recommenders in calculating the similarities and generating meaningful recommendations of high quality, also for new items or to new users. Let be *P, Q* the subsets of problem descriptions and solutions respectively, then cases can be

denoted as ordered pairs $c = (p, q)$, where $p \in P$ and $q \in Q$, while the case base in a CBR system can be defined as the set of the known cases, $C = (P, Q)$.

The key idea behind CBR is that similar problems have similar solutions. Therefore a core concept of the CBR methodology, and a key factor of its successful application, is the *similarity measure* used to identify similar cases. These cases may have the same solution, or their solution may be easily adapted to match the characteristics of the current problem. Case-based recommenders follow the general CBR cycle and rely on the CBR core concepts of *similarity* and *retrieval*. For a case-based recommender, the *user query* serves as a new *problem specification* while the available items' and their descriptions form the cases in the case base. The item(s) to be recommended are retrieved, based on their similarity to the user's request.[7]

Let the description of a case with *t* attributes be $c[a_1, \ldots, a_t]$. In order to calculate the *global similarity* between two cases, first the *local similarities* of the different attributes have to be specified. Usually the global similarity can be calculated as the aggregation of the local similarities from the following equation: $Sim(c^I, c^R) = \frac{\sum_{i=1}^{n} w_i \times sim(a_i^I, a_i^R)}{\sum_{i=1}^{n} w_i}$,[12] where a_i^I, a_i^R are the values of the *i-th* attribute in the input c^I and the retrieved c^R cases respectively, $sim(a_i^I, a_i^R)$ is the local similarity function used for their comparison and $w_i \in [0,1]$ the corresponding weighting factor. Depending on the system's purpose and the type of recommended items, different local similarity functions are used, while the weighting factors may be specified by feature weighting algorithms or by the users as expression of their preferences.

CBR recommenders have been mainly applied in product recommendations especially in electronic stores in order support the intelligent product selection by identifying the products that best match a user's request.[7] These systems focus on the processes of mapping user requirements into a proper problem specification, selecting and retrieving items' based on their similarity to the user's request, and their presentation, as there are few stores that enable products' customization. CBR recommenders have been also applied in travel recommendations for both travel services (hotels, museums etc.) and complete travel plans.[15] In addition, due to their case modeling and the higher flexibility offered to the decision making processes, CBR recommender systems can be used also in applications where contextual and other user specific information related to the time of the selection has to be incorporated. Another interesting application of case-based recommendations is related to music playlists' generation. A playlist is a various and coherent collection of music items of some characteristics, presented in a meaningful sequence. Therefore modeling entire playlists as cases enables identifying the relevance of songs based on their co-occurrences. [16]

Approach Description

The majority of the used recommendation techniques ignore the fact that in many situations *the utility a user obtains depends on the set of items selected together more than on*

the isolated attributes of the selected items. Sometimes, these "relations" among items may seem obvious while in other situations they have to be inferred from the underlying patterns.

CF techniques evaluate only the ratings assigned by users to items, while CB techniques and the majority of case-based item recommenders focus on the characteristics of items that have been liked or requested by a user. The commonly used approach in market basket analysis, the ARs mining methodology, evaluates the presence or absence of items within a transaction to extract patterns and generate recommendations based on those. Recently, latent semantic analysis and probabilistic topic models have been applied to RSs for the recommendations of sets of items, by evaluating the similarities of latent topics in order to recommend items.

The proposed recommender uses CBR to identify and recommend the items that seem more suitable for completing a user's buying experience provided that he/she has already selected some items. The system models complete transactions as cases and the recommended items come from the evaluation of those. As the cases are not restricted to the user that purchased them, the developed system can generate accurate item recommendations in cases of joint item selections, for new and existing users. Having analyzed the previous transactions and identified the concepts within which concrete items appear, the given part of a new transaction is matched over the existing ones in order to find the more adequate solution, the best way to fill this basket.

Following the formalization used in market basket analysis, let $I = \{i_1, i_2, \ldots, i_l\}$ be the set of l distinct elements called items that can be found in a database. Let D be a transactional database that contains a set of transactions $T = \{t_1, t_2, \ldots, t_m\}$, where each transaction $t_j = \{i_a, i_b, \ldots\}$ contains a subset of items from I that were purchased together at a specific period of time by a user from the set of users $U = \{u_1, u_2, \ldots, u_k\}$. In contrast to most CBR item recommenders[7, 8] that trade items as cases and their attributes form the case description $i[a_1, \ldots, a_t]$ we model every transaction as a case and its items as attributes of the case $c[i_1, \ldots, i_j]$, in order to capture the sets of items that are selected together. A case can be denoted as an ordered pair $c = (p, q)$ where $p = \{i_{i1}, \ldots, i_{ij}\} \in P$ is the problem description (the set of already selected items) and $q = \{i_{il}, \ldots, i_{i(l+n-1)}\} \in Q$ is the problem solution of size n (the set of items that will complete this transaction), both subsets of I with $p \cap q = \emptyset$. The case base $C = (P, Q)$ is the set of transactions that have been performed, where all the included items are known.

The treated items are not (directly) associated with quality attributes that would either enable a content-based comparison or affect their acceptance by the users. Therefore, we view all the items as attributes of the same importance for the user's buying experience. The global similarity of two cases is the aggregated similarity of the local similarities of the items included in them with equal weighting factors, therefore $Sim(c^I, c^R) = \sum_{i=1}^{t} sim(i_i^I, i_i^R)$.

Currently, a lot of alternative items/services able to cover the same, or very similar needs can be found, with their differences mainly encountered in marketing characteristics rather than in core specifications. In addition, although the number of registered users is usually high,

each one experiences only a small percentage of the available items. As a consequence, the resulting buying patterns are sparse and may represent situations that occur by chance. In order to identify user habits, it is important to identify the specification of an item that appears within a concrete *concept*. For instance, the set of items bought together before a football night, like coke, beer, nuts, etc., can be seen as the "snack for watching TV", concept.

A pre-processing phase that analyzes the items and transforms them to a convenient representation that enables their further comparison is important. More than classifying the items based on the exact values of their characteristics or the ratings assigned to them, our approach categorizes the items based on the types of their hierarchical attributes (categories that they belong to). As it can be seen in the tree structure of Figure 3, starting from the generic item concept, as root, at each level we specify one more category that an item belongs to, so we divide the existing sets of items into smaller, more coherent subsets. As we go lower in the tree these sets get smaller until reaching the leaves that are the unique items. At each level the items that have common ancestors and are not differentiated at the current level are grouped together and are treated as "same". More specific, as same items at the third level are regarded the items belonging to the same categories at the first two levels that have the same characteristics at the third level. For example (Figure 3), in a physical store at the second level all types of milk would be grouped together, while at the third level based on the type of milk, full and semi-skimmed milk are distinguished, at the forth level we have distinctions based on the distribution package, and finally the brand name specifies the unique item.

As from the initial unique ids cannot be extracted meaningful information about the items, to each of the new groups are assigned appropriate "ids/tags" in order to enable their comparison. These ids are generated based on the categories that the items belong to, following the path from the tree root to its position. Having specified the level of detail needed, the required ids are generated as seen in Figure 3 and the similarity of items in the new and the existing cases can be calculated based on those ids. Two items i_i^I, i_i^R from the input and a retrieved case are thought to be *similar* to the extent to which they share the same path from the tree root to their position. Thus their level of similarity can be calculated from the following equation: $sim\left(i_i^I, i_i^R\right) = \dfrac{length\_of\_common\_path\left(\left(i_i^I, i_i^R\right)\right)}{level\left(i_i^I\right)}$

Active user / New case (c_j)

i_{j1}		?

K most similar cases

i_{j1}		i_{jt}
$i_{(j+k-1)1}$		$i_{(j+k-1)t}$

Recommendations
Set of n items

Analyze cases into items

$Sim(c_l,c_j)=\Sigma sim(i_{lk},i_{jk})$

Items

Products (1...N)

Similarities of **cases**

Case Base

Case		Item	
c_1	i_{11}		i_{1t}
c_2	i_{21}		i_{2t}
c_n	i_{n1}		i_{nt}

L1.1		L1.N

Drinks (1...K)

$sim(i_{lk},i_{jk})$

L2.1		L2.K

Milk (1...L)

Similarities of **items**

L3.1		L3.L

Semi-skimmed (1...M) *Full-fat*

Transform items

Item	Level1	Level2	Level3	Level4	FinalID
X	L1.1	L2.1	L3.J	L4.M	1.1.J.M
Y	L1.1	L2.1	L3.J	L4.M	1.1.J.M
Z	L1.1	L2.1	L3.J	L4.1	1.1.J.1

Similarity (X, Y) > Similarity (X, Z)

L4.1		L4.M

Glass bottle *Paper brick*

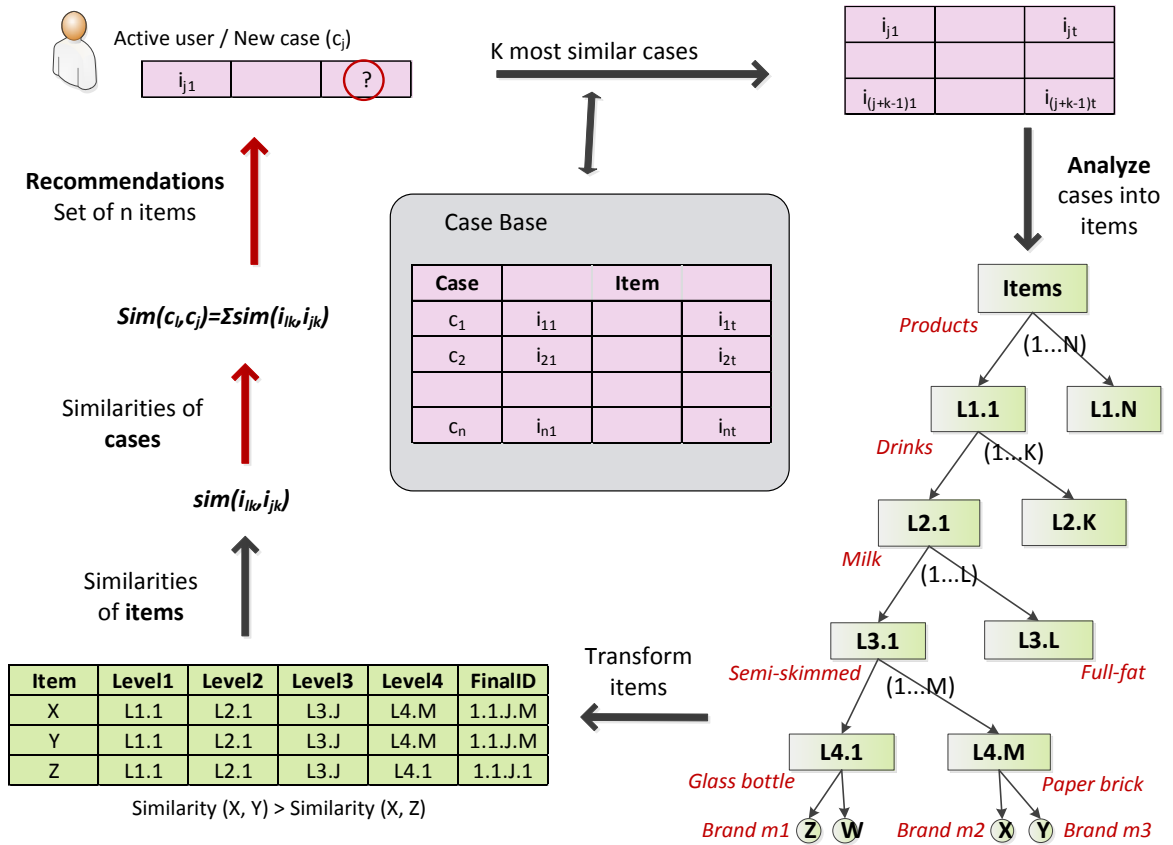*Brand m1* Z W   *Brand m2* X Y *Brand m3*

Figure 3: Recommendation process

Depending on the application domain, the hierarchical attributes and their classification may be extracted from a proper ontology or a simpler categorization. The described classification of items forms part of the recommendation pre-processing phase, is used to enable the calculation of similarities between cases and to highlight the type of items appearing in different cases. Nevertheless, the implemented recommendation methodology does not generate recommendations based on a clustering of items.

When a new user comes, the set of already selected items is compared to those in the existing cases, in order to identify past cases with similar descriptions and the way they were structured. The *k* most similar case(s) are retrieved, and the items that most frequently appear in their solution parts are recommended to the user. Therefore, even if the new case belongs to an unknown user, or it contains items that appear only under certain circumstances, the recommender is able to generate accurate recommendations.

Experimentation Results & Evaluation

For the evaluation of the implemented recommender, a transactional dataset with real market data from a European supermarket was used. The number of the offered items was 102142, while 1057076 transactions were performed by 17672 customers. Each transaction is associated with the user that purchased it and the included items. Each item, apart from its name and unique id, is associated with the various categories that it belongs to (general category, item group and two item subgroups, that for example would be, drink, milk, semi-skimmed, glass bottle of 1litre, brand name). This information was used to transform the items into a proper depiction, as described above, before generating recommendations. In addition, demographic and subscription information about users, can be found in this dataset.

The recommender was tested for different values of the parameters that affect its performance, like the *number of similar cases retrieved (k),* the *number of recommended items* and the *level of detail* at which the items are represented. In order to specify the number of similar cases that would be used, recommendations were generated using 1, 5, 10, 20 and 30 similar cases. The experimental results shown that the best option is to use only the most similar case, thus k was set equal to 1.

The performance of the developed recommender system was compared to three of the recommendation techniques being widely used, namely association rules based, probabilistic topic model, and Collaborative Filtering user-based recommendations. However, only the first two techniques address exactly the same problem, recommendations of sets of items. CF recommendations focus on the ratings users assign to items and do not take into account joint item selections. However as the CF technique is being widely used in commercial applications its results are also presented. As the items' descriptions in the transactional database do not contain their quality attributes we did not test a CB or a CBR product recommendation approach. The *Apriori algorithm* was used to extract *association rules* from the given transactional database, based on which recommendations were generated. In addition, a *topic model recommender* (*Latent Dirichlet Allocation, LDA*) was implemented. In this approach, the offered items are seen as words of a vocabulary while transactions are treated as documents made as a combination of topics (item concepts) with some probability distribution.[14] In the CF approach the selection of an item is inferred as being a high rating of a user towards this item.

Various experiments were run, selecting each time randomly the 20% of the transactional database as new cases (test set) and the rest as the case base of the system (training set). Using only the most similar case (k=1), the average results for the recommendation of 5, 7, 9 and 11 items are presented in table 1, for different representation levels. As our intention was to evaluate the ability of the recommender to identify and recommend the missing items of the transactions, information retrieval metrics (precision, recall and f-measure) were used for the evaluation, with our focus being on the precision value.

	ARs			LDA			CF			CBR		
# Items	Pr	R	F-1	Pr	R	F-1	Pr	R	F-1	Pr	R	F-1
Level 2												
5	0,226	0,089	0,128	0,418	0,084	0,139	0,242	0,205	0,222	0,446	0,177	0,253
7	0,282	0,080	0,125	0,493	0,099	0,164	0,308	0,252	0,277	0,554	0,206	0,300
9	0,337	0,075	0,123	0,531	0,106	0,177	0,366	0,296	0,327	0,629	0,226	0,333
11	0,390	0,071	0,120	0,592	0,118	0,197	0,417	0,321	0,363	0,678	0,239	0,353
Level 3												
5	0,137	0,040	0,062	0,086	0,017	0,029	0,072	0,064	0,068	0,318	0,110	0,163
7	0,168	0,038	0,062	0,135	0,019	0,034	0,093	0,082	0,087	0,445	0,154	0,229
9	0,198	0,037	0,062	0,190	0,021	0,038	0,114	0,099	0,106	0,519	0,176	0,263
11	0,225	0,036	0,062	0,197	0,018	0,033	0,140	0,119	0,129	0,570	0,185	0,279

Table 1: *Recommendation results in terms of precision (Pr), recall (R) and f-measure (F-1) for the use of level 2 and level 3 items' representations*

As it can be seen, the accuracy of all the methodologies highly improves when using more abstract descriptions. However as the CBR recommender evaluates the degree of an item's similarity with the items in the target case and not only its presence or absence, it outperforms the other recommenders at both representation levels. In contrast, the LDA recommender evaluates similarities among item concepts (topics) while the ARs recommender evaluates only the presence or absence of items within transactions in order to extract the buying patterns and generate recommendations. Finally, CF recommenders take into account only the presence of items in the user profiles without evaluating the items' co-occurrences within transactions.

Conclusions and Future Work

Recommender Systems have become an important part of numerous commercial applications, enabling customers and providers in their decision making processes while pursuing their buying and selling strategies. To this direction, the identification of the items' selection patterns is of high importance.

In the present article, a different recommendation approach for the market basket analysis and similar set of items recommendations, using Case-Based Reasoning was presented. Given that a user has already selected some items, the developed recommender can generate accurate recommendations for items that might be of interest to him/her while being able to provide more insight into the patterns that define transactions' structure. According to the experimentation done, the proposed approach performs better that the tested state of the art recommendation techniques (*ARs, LDA* and *CF*) for this kind of problems. One of its main advantages is its ability to recommend *complementary items* to the already selected ones.

Additionally it can recommend *less popular items* and generate recommendations to *new users*, reducing the *cold start* and the *overspecialization* problems of CF and CB techniques.

This work could be further extended by incorporating a second processing level into the recommendation methodology. At this level, additional quality characteristics of the items and constraints related to them may be incorporated (for example to recommend only new items, etc.).

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. B. Kantor, Eds. Boston, MA: Springer US, 2011, pp. 1–35.
2. L. Cavique, "A scalable algorithm for the market basket analysis," *Journal of Retailing and Consumer Services*, vol. 14, no. 6, pp. 400–407, Nov. 2007.
3. R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *ACM SIGMOD Record*, 1993, vol. 22, pp. 207–216.
4. P. Melville and V. Sindhwani, "Recommender Systems," in: *Encyclopedia of Machine Learning*, 2010.
5. G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 6, pp. 734–749, 2005.
6. Z. Huang, D. Zeng, H. Chen, "A Comparison of Collaborative-Filtering Recommendation Algorithms for E-commerce," *Intelligent Systems, IEEE,* vol. 22, no. 5, pp. 68-78, Sep.-Oct. 2007.
7. D. Bridge, M. H. GöKer, L. McGINTY, and B. Smyth, "Case-based recommender systems," *The Knowledge Engineering Review*, vol. 20, no. 03, p. 315, May 2006.
8. F. Lorenzi and F. Ricci, "Case-based recommender systems: A unifying view," in *Intelligent Techniques for Web Personalization*, Springer, 2005, pp. 89–113.
9. J. H. Su, H. H. Yeh, P. S. Yu, V. S. Tseng, "Music Recommendation Using Content and Context Information Mining," *Intelligent Systems, IEEE* , vol.25, no.1, pp.16-26, Jan.-Feb. 2010.
10. R. Bergmann, "Introduction to case-based reasoning," *University of Kaiserlautern*, 1998.
11. R. L. de Mántaras, "Case-Based Reasoning," *Machine Learning and its Application*, vol. 2049, pp.127-145, 2001.
12. G. Finnie and Z. Sun, "Similarity and metrics in case-based reasoning," *International journal of intelligent systems*, vol. 17, no. 3, pp. 273–287, 2002.
13. M. Steyvers and T. Griffiths, "Probabilistic topic models," *Handbook of latent semantic analysis,* vol. 427, pp. 424-440, 2007.

14. K. Christidis, D. Apostolou, G. Mentzas, "Exploring Customer Preferences with Probabilistic Topic Models," *European Conference on Machine Learning, Workshop on Preference Learning*, 2010.

15. F. Ricci, B. Arslan, N. Mirzadeh, A. Venturini, "ITR: A Case-Based Travel Advisory System," *Case-Based Reasoning, Springer,* vol. 2416, pp. 613-627, 2002.

16. C. Baccigalupo, E. Plaza, "Case-Based Sequential Ordering of Songs for Playlist Recommendation," in T. Roth-Berghofer, M. H. Göker, H. A. Güvenir (eds.) ECCBR 2006, *Lecture Notes in Computer Science,* vol. 4106, pp.286-300, 2006.

The Authors

Anna Gatzioura is a PhD student at the Universitat Politècnica de Catalunya · BarcelonaTech. Her research interests include Intelligent Decision Support Systems, Recommender Systems, Machine Learning and Artificial Intelligence applications. She received her Engineering and Master degrees from the National Technical University of Athens. Contact her at Campus Nord, Building Omega, Universitat Politècnica de Catalunya, Jordi Girona 1-3, 08034 Barcelona, Spain; gatzioura@cs.upc.edu.

Miquel Sànchez-Marrè is an Associate Professor at the Computer Science Department at Universitat Politècnica de Catalunya · BarcelonaTech. His research interests include Case-Based Reasoning, Machine Learning, Data Mining, Intelligent Decision Support Systems, Integrated architectures for AI Artificial Intelligence, and Applications of IDSS. He is the coordinator of the Spanish Network for Advancement and Transfer of the Applied Computational Intelligence (ATICA) and a founding member of the Catalan Association for Artificial Intelligence (ACIA). He received his PhD in Computer Science from the Universitat Politècnica de Catalunya · BarcelonaTech. Contact him at Campus Nord, Building Omega, Universitat Politècnica de Catalunya, Jordi Girona 1-3, 08034 Barcelona, Spain; miquel@cs.upc.edu.