

A hybrid recommendation approach for a tourism system

Joel P. Lucas, Nuno Luz, María N. Moreno, Ricardo Anacleto, Ana Almeida Figueiredo, Constantino Martins

ABSTRACT

Many current e-commerce systems provide personalization when their content is shown to users. In this sense, recommender systems make personalized suggestions and provide information of items available in the system. Nowadays, there is a vast amount of methods, including data mining techniques that can be employed for personalization in recommender systems. However, these methods are still quite vulnerable to some limitations and shortcomings related to recommender environment. In order to deal with some of them, in this work we implement a recommendation methodology in a recommender system for tourism, where classification based on association is applied. Classification based on association methods, also named associative classification methods, consist of an alternative data mining technique, which combines concepts from classification and association in order to allow association rules to be employed in a prediction context. The proposed methodology was evaluated in some case studies, where we could verify that it is able to shorten limitations presented in recommender systems and to enhance recommendation quality.

Keywords: Recommender systems, Associative classification, Fuzzy logic

1. Introduction

The phenomenon of “information explosion” or “information overload” due to the exponential increasing of data available in the Web, results in a lot of useless data, in which it is very difficult to find valuable information. In e-commerce systems this fact is reflected by loads of products available for sale. Therefore, looking for the desired products in the entire catalog becomes a tedious task for users, and probably they cannot find the products they are interested in. The need of endowing e-commerce systems with mechanisms for selective and personalized presentation of products gave rise to the “recommender systems”, which help consumers in finding and purchasing products. Although, these systems are mainly used in the e-commerce environment, they are being extended to other domains, for instance, virtual libraries, news websites, scientific portals, e-learning systems, etc. In spite of the advances achieved in the recommender systems’ field, the recommendations provided by this type of systems have some important drawbacks, such as low reliability and high response time. Therefore, it is necessary to research in new recommender methods that join precision and performance as well as solving other usual problems of these systems (sparsity, grey sheep, first-rater.. .) that will be commented later. The aim of this work is to address these

important problems by means of the proposal of a hybrid method and its validation in a recent application area, the tourism.

Collaborative filtering methods are the most used in recommender systems. They make use of information related to evaluations (or ratings) provided by users. This can cause the sparsity problem when evaluations from users are insufficient. On the other hand, traditional collaborative filtering approaches based on nearest neighbor algorithms show serious performance and scalability problems. In the last years many recommendation techniques have been proposed aiming at improving the quality of the recommendations as well as dealing with other typical drawbacks of recommender systems. Data mining techniques have been successfully applied in recommender systems to predict user preferences. They do not present performance problems since predictive models are already built when the user logs in the system and they are less sensitive to sparsity problems. However, the results vary with the selected algorithm given that they present different behavior depending on the characteristics of the dataset. Therefore, it is necessary to apply suitable algorithms in order to obtain precise recommendations. In this work, a hybrid recommendation method is applied in a tourist recommender system, where we evaluate how it faces typical recommender systems drawbacks. This method combines a clustering technique and an associative classification algorithm to recommend touristic points to the users. It also employs fuzzy logics to enhance recommendations’ quality.

In Section 2, we describe the background and some drawbacks related to recommender systems. Section 3 includes the

recommendation method to be evaluated. In Section 4 we describe the system in which the method was implemented. The case studies used to evaluate such method and the results of its application are presented in section 5. Finally, the conclusions are reported in Section 6.

2. Recommender systems

Recommender systems are becoming indispensable in the web environment since they constitute a way of increasing customer satisfaction and taking positions in the competitive market of the electronic business activities. For many years traditional companies have improved their competitiveness by means of business intelligence strategies supported by techniques like data mining. Data mining algorithms find consumers' profiles and purchase patterns in the corporate databases that can be used for effective marketing and, in general, for business decision making. In the field of the e-commerce these procedures can also be applied in order to develop recommender systems but they have been extended to deal with specific problems inherent to this kind of applications.

2.1. Related work

Recommender systems can be classified depending on the type of method used for making recommendations (Cheung, Kwok, Law, & Tsui, 2003; Lee, Kim, & Rhee 2001). The two main categories are content-based methods and collaborative filtering algorithms. The main difference between the two approaches is the information used for recommending products to the active user. While the former one does not take into account the information acquired by other users, collaborative filtering methods use evaluations about products carried out for other users. In content-based methods web objects are recommended to a user taking into account those he has been interested in the past but the opinions of other users are not considered (Lee et al., 2001). Therefore, recommendations are based on data related to document features as well as on data related to the behavior of each user. Rule-based techniques or keyword-matching are commonly used (Lang, 1995) in this kind of methods.

Although content based methods were the first used in recommender system, currently they are not used in a solely way since they present important weaknesses, such as difficulty in distinguishing between high-quality and low-quality information that is on the same topic (Claypool et al., 1999) or to extract relevant features from certain objects as multimedia items (Balabanovič & Shoham, 1997).

In collaborative filtering methods, the recommendation process is based on ratings of other users who have similar preferences (Claypool et al., 1999). Breese et al. (1998) classified collaborative filtering methods into two groups: memory-based (or user-based) methods and model-based (or item-based) methods. Memory-based methods are also known as the nearest neighbor method because this one was the first technique proposed for these methods. Memory-based methods were the first approach to collaborative filtering, where the whole set of user opinions is needed, since the opinions of the active user are matched with the ones of all other users in order to find his neighbors (the most similar users to him). However, the scarce number of these opinions from some users or about some products causes the sparsity problem commented previously, which leads to poor recommendations. Therefore, these methods are not usually used in an isolated way in current recommendation systems.

On the other hand, model-based methods use data mining techniques in order to develop a model of user ratings, which is employed to predict user preferences. This type of methods has

been developed more recently in order to avoid the sparsity problem (Lucas, Laurent, Moreno, & Teisseire, 2012b; Sasikala & Vidhya, 2012) that affects mainly to memory based methods, but model based ones are also affected by it. Methods within this category usually apply data mining algorithms in order to build the recommendation models, which are used to predict user preferences. As commented in the previous section, these models are built off-line and they are just checked in recommendation time. Therefore, the building time does not affect to the response time and, consequently, scalability problems are avoided. There are many data mining methods to be applied, although machine learning techniques are the most common. Neural networks were the first one used in recommender systems (Bilgus & Pazzani, 1998) changing, in this way, the nearest neighbor approach of collaborative filtering methods by a classification approach. The same technique has been applied in several works such as (Roh et al., 2003) where it is combined with Case Based Reasoning, or (Chou, Li, Chen, & Wu, 2010) where it is used to predict consumer preferences taking into account his navigation behavior through navigation patterns extracted by means of an unsupervised web mining method.

Bayesian networks constitute another technique widely used in the induction of recommendation models (Condliff, Lewis, Madigan, & Posse, 1999) in a single way (Breese et al., 1998) or jointly with other methods (Campos, Fernández-Luna, Huete, & Rueda-Morales, 2010). The main shortcoming of these methods is the high computational cost of building the net, specially when the amount of data is great. Although this is not a critical drawback since classification models are built off-line, when these models need to be often updated it can become a serious inconvenience. It occurs in current recommender systems due to continuous changes in the database of products and users (Koren, 2010).

Support Vector Machines (SVM) are linear classifiers also used in recommender systems (Xu & Araki, 2006). In this context, every user is represented as a vector composed by ratings about products. A hyperplane separates the geometric space where the vectors are situated in classes representing groups of users of similar preferences. The performance depends on the separation margin of the data (Cheung et al., 2003). Since SVM technique was originally designed for binary classification and taking into account that it has yielded very good results in several domains, different strategies to reconstruct a multi-class classifier from binary SVM classifiers are studied (Wang, Yuan, Liu, Yu, & Li, 2009). In some works, SVM is used as a complementary technique to other methods. For instance, in [Diez, del Coz, Luaces, & Bahamonde, 2008] it is used to induce ranking functions from the preference judgments of each user as a previous step to the application of a clustering algorithm that builds groups of people with closely related tastes.

In spite of being an unsupervised method, clustering is often used as a model based technique since the induced groups of people with similar preferences constitute a way of classification. Thus, the predictions for the active user are based in the opinions of the members of the group he belongs to. By means of fuzzy logic a user can be assigned to more than one cluster with different belonging degree and receive recommendation from more than one group. In any case, the personalization achieved is lesser than the one provided by other methods. For that reason, clustering is usually used in combination with other algorithms, for example, as a previous step to the application of machine learning algorithms (Schafer, 2005) or other collaborative filtering methods. In (Kim & Ahn, 2008) a genetic clustering algorithm, GA-based K-means, is proposed for generating groups of similar users, where the desired collaborative filtering techniques are applied, in order to reduce the computational time required for applying those techniques to the entire dataset. Variant of this approach consists on obtaining optimal similarity functions from rating of users by means of a genetic algorithm (Bobadilla, Ortega, Hernando, &

Alcalá, 2011). Regardless of its simplicity, this proposal provides good accuracy and performance results without need of demographic data and other kind of information, apart from the ratings. However, the current trendy, specially in sparse contexts where ratings are insufficient, is to exploit hybrid methodologies combining different approaches in order to take advantage of the strengths of each of them. In recent works, even semantic information is added to the available data in order to improve recommendations (Blanco-Fernández et al., 2008; Kim, Alkhaldi, El Saddik, & Jo, 2011; Moreno, Lucas, López, & Muñoz, 2010).

Another group of works propose recommendation methods for different contexts, for instance, for collaborative work environments. In (Zhen, Huang, & Jiang, 2009) a collaborative filtering approach using team workflow information is applied in order to derive demands for knowledge of individual members. The shared knowledge by collaborative team members is also the focus of a distributed recommender proposal for a peer-to-peer (P2P) environment (Zhen, Jiang, & Song, 2010). In this case both the recommender and the knowledge is distributed between the peers of the P2P network of collaborative team members and the distributed recommender receives knowledge supplied from the source peers, and delivers knowledge to the destination peers. The recommendation method is based on computing the similarity between two peers from the similarity of knowledge resources in the two peers' profile information. Profile information is given by a set of attributes.

We propose in this work a hybrid method embracing aspects from collaborative filtering and content-based approaches, which is validated in a tourism recommender system. Tourism field is a promising application domain where many research works are being addressed to. Some of them are related in the next subsection.

2.2. Tourism recommender systems

Given its complexity, tourism is a privileged area for the application of artificial intelligence (stock & Zancanaro, 2002), and, in particular, Decision Support Systems (Felfernig, Gordea, Jannach, Teppan, & Zanker, 2007) which are a generalization of Recommender Systems. According to the Travel Industry Association of America (www.tia.org), in 2003, 30% of the United States adult population (64 million) used the Internet to look for information about destinations or to check prices and schedules. In addition, 66% of them booked travel via the Internet. Moreover, according to the World Travel & Tourism Council, travel and tourism represents approximately 11% of the worldwide GDP (gross domestic product). In this way, the tourism domain is a crucial and strategic application field in recommender systems.

Recommender systems designed for tourism applications are also named as Travel Recommender Systems or Destination Recommendation System. These systems can be classified as an intermediary between customer and travel agency (Loh, Lorenzi, Saldaa, & Lichnow, 2004). Moreover, they have the advantage of not being restricted by the human factor while travel agents are limited by their knowledge about destinations and places of interest and recommendations offered by them may be determined by their own opinions and tastes.

The tourism context is especially interesting because recommendations may refer to a variety of products, such as locations, attractions, accommodations, and flights, in order to provide a meaningful picture of the proposed travel (Werthner & Ricci, 2004). Current recommender systems for tourism are generally designed for accomplishing one of the two (or even both) following purposes: to aid the user in planning his trip (i.e. choosing one or more destinations) and to aid the user in planning what to do in a certain place (usually a city). Recommendation methods for tourism may explore the way items are presented to the user in order to enhance the system accessibility.

Nevertheless, both recommendation purposes are not trivial, because in the moment of planning a trip the user will be confronted with a vast amount of possible combinations of locations and activities. Moreover, there are specific constraints that should be taken into account at recommendation time, like local weather, traffic, seasonal events, etc. Therefore, an efficient recommender system has to help the user in filtering locations and/or activities in order to fulfill his needs and expectations. They can perform this task by matching the user's preferences and wishes against all available options and services, and then helping the customer to decide his/her travel plan (Loh et al., 2004). Hence, travel recommender systems attempt to emulate offline travel agents by providing users with knowledgeable travel suggestions to facilitate their decision-making processes (Berka & Plnig, 2004; Zanker, Fuchs, Hopken, Tuta, & Muller, 2008).

However, popular travel Web agencies, like Expedia (www.expedia.com), merely use the average ratings of other users as an informative parameter of a certain touristic place or destination. In this way, these agencies only use the potential of user communities by informing their overall preferences. According to Felfernig et al. (2007), this context exists mostly due to the difficulty on establishing reasonable user profiles, since individual travel planning activities are typically much less frequent than, for example, book purchases, and in addition the items themselves may have a far more complex structure.

Nevertheless, there are some ongoing works that effectively provide recommendations for their users. Most of the methods employed in these systems depend on just one of the filtering approaches (Ricci & Werthner, 2002). The two most successful technologies for travel and tourism using the content-based filtering approach, are the ones used in Triplehop's TripMatcher and VacationCoach's expert advice platform (Ricci, 2002). Their purpose as a recommender system is to aid the user in planning a trip (first recommendation purpose described above). In Triplehop's TripMatcher the user interacts with the system by proving input information such as his preferences, needs and characteristics. Afterwards, the input is matched with features of the set of destinations available. VacationCoach's requires repeated user interaction in order to provide recommendations. However, instead of asking the user specific questions about his preferences, it requests the user to classify himself in one general user profile, like "mountain lover" or "cinema addicted".

Currently, there are few works employing only the memory-based collaborative filtering approach. TripAdvisor, which is probably the most popular tourism recommender system for travel and tourism, is one of them. It is related to both recommendation purposes described before, i.e., to aid the user in planning a trip and also to provide assistance in what to do in a specific destination. Its recommendations are also based on ratings and comments collected from users. In this way, recommendations are subscribed to the active user by comparing his ratings with the ones of other users (users are grouped according to items' ratings). Despite TripAdvisor's popularity, its recommendation process does not consider any user modeling technique and, therefore, the system probably suffers from typical recommender systems drawbacks.

The use of model-based collaborative filtering techniques in recommender systems for tourism may avoid some limitations associated with memory-based methods. The use of case-based reasoning is quite common in tourism systems. Specifically for the tourism and travel domain, Ricci and Werthner (2002) developed a recommender system based on case-based reasoning, which aims at helping the user to choose a destination and also to plan his activities (both recommendation purposes). This system join data gathered from existent external tourism portals and makes use of a XML-based mediator architecture, data mapping methods, online analytical processing and similarity-based retrieval. Another

successful example of the use of case-based reasoning is the Dietorecs system, which is a recommender system that applies both recommendation approaches for collaborative filtering (memory-based and model-based). This system builds a recommender model through cases on an analysis of user's interactions over the five types of items that were defined on the system.

Nevertheless, case-based reasoning is often combined with interactive query refinement to provide reasonable recommendations of either complete trips or single products, and many approaches aim at eliciting the user preferences and requirements in a conversational dialog (Felfernig et al., 2007). The virtual spa advisor, VIBE (Jannach, Zanker, Jessenitschnig, & Seidler, 2007), is one of these systems. It provides the active user a single point of contact for multi-lingual guidance and preference recognition. Recommendations are provided when the dialog ends.

However, on-line users may be different with respect to their background knowledge, or their capabilities of expressing their needs and requirements (Felfernig et al., 2007). Since using one filtering technique can fall short when trying to make recommendations for complex products (Berka & Plnig, 2004), hybrid approaches that combine content and collaborative-based approaches are most likely to succeed (Ricci, 2002).

In that sense, the Tourism Information Provider (TIP) recommender system applies both content-based and collaborative approaches. The system provides information about touristic points based on the user characteristics. These characteristics are composed of demographic information, travel history and user interests on specific touristic points. These points are taken by the system as semantic groups and the active user is classified, according to the data in his profile, in one of these groups. TIP is a combination of an event-based system (EBS) and a location-based service (LBS) applied to a mobile environment (Hinze, Voisard, & Buchanan, 2009).

2.3. Weaknesses

The reliability of recommendations depends significantly on the method used in the recommender systems and the data from which the models are induced. Low reliability causes two types of errors: false negative and false positive. The first one refers to not recommended products, though the consumer would like them. The second one involves recommended products that the consumer does not like. False positives are the most critical errors since they will lead to angry consumers (Sarwar, Karypis, Konstan, & Riedl, 2000). In order to avoid all kind of errors but mainly false positives, methods employed in recommender systems must deal with some typical drawbacks usually occasioned by the fact that recommendations in collaborative filtering are based on user preferences rather than objective properties of domain items (Leung, Chan, & Chung, 2008). A description of the four most critical drawbacks is provided in the following paragraphs.

The cold-start problem has also been addressed in recent works. Most of them focus on finding new similarity metrics for the memory-based CF approach since traditional measures such as Pearson's correlation and cosine provide poor recommendations when the available number of ratings is scant, a situation that becomes critical in the cases of the cold-start and first-rater problems. In (Ahn, 2008), a heuristic similarity measure based on the minute meanings of co-ratings is proposed in order to improve recommendation performance. Another similarity measure can be found in (Bobadilla, Ortega, Hernando, & Bernal, 2012); this is a linear combination of simple similarity measures obtained by using optimization techniques based on neural networks.

Probably the most critical problem is related to data sparsity, caused by the fact that the number of ratings needed for making recommendations is greater than the number of the ratings

obtained from users. This problem becomes worse when the number of items and the number of users in the system is very large in comparison with the available ratings. Most recommender system techniques require user explicit expression of personal preferences for items, which are difficult to obtain. Thus, a way to deal with sparsity, is to develop methods for obtaining ratings implicitly. On the other hand, methods that also use other kind of attributes, apart from ratings, can be applied to reduce sparsity. This is the case of model based techniques as data mining; however, even using these methods sparsity still remains a critical drawback for recommender systems, since, although the number of required ratings is lesser than by memory based methods, the available ratings cannot be enough regarding the great number of products present in most of recommender systems.

Memory based methods present another problem also caused by the large number of items available in recommender systems. It is the scalability problem due to the high computational time required for making recommendation, which grows proportionally to the number of users and products in the system. Model based methods do not present this drawback since the recommender model is already built when the user accesses the system; therefore, time spent in building the model has no effects in the user response time is related to scalability.

Although the above mentioned drawbacks can be tackled by means of model-based collaborative filtering methods, there are some ones that these methods cannot solve. The "early (first) rater problem" (Claypool et al., 1999; Condliff et al., 1999) is an example of a drawback that may occur in all types of collaborative filtering methods. This problem refers to the impossibility of providing recommendations about an item that was recently added to the system. As a consequence, the item will have few (or none) ratings from users. In fact, the early rater problem is directly related to sparsity, because a system encompassing a high number of items will probably present many items that have not received any rating. In a similar way, this drawback also occurs with a new user accessing the system, because there is no available information about him, hence his behavior cannot be perceived in order to provide recommendations to him. This is also known as cold-start problem. An extreme case of the early rater problem occurs when a recommender system first begins, since every user suffers from the early rater problem for every item (Claypool et al., 1999). The first rater problem neither occurs in content based methods since they can provide recommendations based only on the properties of an item. According to Condliff et al. (1999), since a content-based system does not consider the social background of its users, the system is limited to recommend just items that are similar to those that a user has liked in the past. Therefore, there could be many false negatives, because these methods are not able to distinguish between high and low information quality within the same subject.

The "grey sheep problem" (Claypool et al., 1999) is another weakness associated with collaborative filtering methods. This problem refers to the users who have opinions that do not consistently agree or disagree with any group of users. As a consequence, these users do not receive recommendations. However, this problem does not occur in content-based methods, because these methods do not consider opinions acquired from other system users in order to make recommendations.

2.4. Associative classification and fuzzy logic

In previous works (Lucas et al. 2008, 2012a), we have found out that classification based on association (associative classification) yields better results than other classification techniques with data from recommender systems. Fuzzy logic has also been successfully applied in this field in combination with associative classification

(Lucas et al., 2012b). For that reason, the recommender method used in the tourism system presented in this paper is mainly based on fuzzy logic and associative classification. Thus, some works related to these technologies and their use in the recommendation field are described in this section.

In the same way that association rules are successfully applied in many domains, they can be used in recommender systems, but they need to be adapted to the special characteristics of this kind of systems (Cheung et al., 2003). Taking into account this consideration, association rules can form a very compact representation of preference data that may improve efficiency of storage as well as performance in recommender systems (Schafer, 2005).

Association rules were first introduced by Agrawal, Imielinski, and Swami (1993) aiming at discovering consuming patterns in retail databases. They demonstrated that an association rule expresses, in a dataset, the probability that the occurrence of a set of items implies the occurrence of another set of items. Thus, association is not considered as a prediction task, because it aims at describing data. On the contrary, classification is a prediction task, because it aims at predicting the value of an attribute (label) in a data set. The joining of concepts from classification and association (Liu, Hsu, & Ma, 1998) is an alternative approach for performing classification processes, where association rule models are employed for classification. Given that association models are commonly more effective than classification models, a crucial matter that encourages the use of association rules in classification is the high computational cost that current classification methods present. Several works (Liu et al., 1998; Li, Han, & Pei, 2001; Yin & Han, 2003; Thabtah, Cowling, & Peng, 2004) verified that classification based on association methods presents higher accuracy than traditional classification methods.

In recommender systems, association rules are usually used in the way proposed by Sarwar et al. (2000) in order to find associations among rated or co-purchased items by users, generating item recommendations based on the strength of the association between items. Thus, the use of association rules in recommender systems is basically a collaborative filtering approach, because rule mining is based on data gathered from user opinions. Sun, Kong, and Chen (2005) showed an improvement in accuracy when comparing the use of association rules to classical collaborative filtering methods (e.g. correlation matrix). Fu, Budzik, and Hammond (2000) have developed a recommender system of web pages using the Apriori algorithm to mine association rules on users' navigation history. Basically, in the recommender systems context, association rules are employed aiming at identifying items frequently found in "association" with items in which the active user has expressed interest. Current recommender systems can also combine other techniques with association rule mining. In (Zhang & Chang, 2005) association rules were applied with sequential rules in order to increase the efficiency of the recommendations. Another example of a hybrid algorithm is given in (Forsati & Meybodi, 2010) where a distributed learning automata and a weighted association rule mining algorithm are applied for web page recommendation.

Our approach consists on applying association rule mining in a classification context, where, instead of considering only the occurrence of items and users, we consider attributes describing the users and items of the system. In a similar way, the proposal of Lin, Alvarez, and Ruiz (2002) consists on mining rules for one target item in the consequent term using a variant of the rule generation module of the CBA algorithm (CBA-RG), based on the classical version of the Apriori algorithm. However, it does not use attributes of users or items, because it is based exclusively on their occurrence. Thus, only the most frequent items are recommended, however, some possible items of interest to the active user, but not frequent, are ignored. On the other hand, they consider relationships between users as well, where association rules for items and users

are mined separately and items are distinguished by means of a binary rating scheme (with "like" or "dislike" values). Users are associated according to their preferences (liking or disliking) over certain items on the system. However, just one type of association rule is used: if there are few ratings given by the active user, associations between items will be used, otherwise just associations between users will be considered. Rules are mined at runtime for each specific target user, where the specification of a minimum support is not required in advance. A target range is given for the number of rules, and the algorithm adjusts the minimum support for each user in order to obtain a rule set whose size is in the desired range (Lin, Alvarez, & Ruiz, 2002). However, this procedure may be very onerous when dealing with a sparse or high dimensional dataset, because the rules' mining is made through an iterative process so that the support is defined automatically. A maximum number of rules' threshold needs to be defined previously, then the algorithm executes the iterative process, increasing the minimum support count, until the number of rules is lower than the defined threshold. In the approach proposed in this work, the set of rules used for classification is constructed off-line, which enables the system to give quick answers to the user when he asks for recommendations.

The method proposed in (Forsati & Meybodi, 2010) also addresses the problem of finding the complete set of itemset satisfying a minimum support threshold by introducing the concept of weighted support to find frequent items. The traditional association rule mining algorithm is extended by allowing a weight to be associated with each item in a transaction to reflect the interest of each item within the transaction. The weight is based on the time spent by each user on each page and the visiting frequency. The weighted association rules of each URL are extracted from the web log data and similarity between active user sessions is calculated upon the weighted rules. The most similar rules to the active user session with the highest weighted confidence are used for recommendation. They are found by scoring each rule in terms of both its similarity to the active session and its weighted confidence. Since this method is used for page recommendation it exploits navigation information but not rating data, therefore it would be complicated to adapt it to a collaborative filtering approach.

IMSApriori (Kiran & Reddy, 2009) is a recent method that uses a particular metric to determine appropriate minimum support values per item in order to mine rare itemsets. Their authors have demonstrated that this method is better suited to mine rare itemsets than previous methods. In (Gedikli & Jannach, 2010) the predictive accuracy of the IMSApriori algorithm in a recommender context is evaluated. The work includes the proposal of a new procedure for association rule-based recommendation called NRR (Neighborhood-restricted Rule-based Recommender), which consists of learning a personalized set of rules for each user from a subset of the transaction database containing his nearest neighbors. In the study carried out the NRR method yielded better precision than IMSApriori and k-nearest neighbor (k-NN) for sparse datasets, while with medium and low sparsity the pure IMSApriori version showed a better behavior than the k-NN method. This proposal only considers associations between items in the rules but it does not consider characteristics of rules and items, thus, recommendations of not purchased or evaluated products cannot be provided. In the same way, new users without purchases or evaluations about products cannot receive recommendations. The treatment of the rare itemsets only solves the first rater and cold start problem in a partial way.

The problem of managing numerical attributes is another important concern in the induction of association rules that can be addressed by means of fuzzy logic. This approach has been successfully used in many application areas; however, it has not been widely explored in recommender systems. In (Berka and Plnig, 2004) fuzzy association rules were used in the tourism recommendation field.

Fuzzy logic provides soft transitions between sets very suitable for tourism applications, where, a user is able, for example, to prefer a restaurant which is within a certain physical distance, but without having a fixed maximum distance. Apart from the advantages of this technique in particular domain, fuzzy logic can help minimizing, or even solving, typical drawbacks of these systems. Dubois, Hullermeier, and Prade (2006) affirm that fuzzy logic provides high-value properties to recover items stored in a database and, as a consequence, to provide recommendations for users. The reason is the capability of fuzzy sets to manage concepts such as similarity, preference and uncertainty in a unified way, while also performing approximate reasoning. Due to these advantages, especially for uncertainty, fuzzy logic can help to minimize the sparsity problem, which is the main drawback in current recommender systems.

Depending on the context and the type of method considered, fuzzy logic can be used both in content-based and collaborative filtering approaches. A general use of fuzzy logic with both types of methods is proposed in (Dubois, Hullermeier, & Prade, 2006), where a case-based decision support system is implemented as the basis to contemplate situations where users do not have absolute preferences, or where preferences are expressed relatively to the context in order to be stored.

There are more specific works, such as (Yager, 2003), where fuzzy set methods are used to describe information in a content-based recommender system, or (Cao, Li, & Liao, 2005) where fuzzy concepts are used in order to recommend to users products not usually consumed. This work also addresses other situations in which there may not be enough information about the customer's past purchases and the customer may give his specific requirements in each single purchase. In that context, fuzzy set operations are used in order to define relationships between user requirements and product features. On the other hand, fuzzy logic methods are applied in some works for developing recommendation approaches based on collaborative filtering. In (Nasraoui, Frigui, Joshi, & Krishnapuram, 1999) the notion of user session is defined as a compact temporary sequence of web accesses made by the user. These sessions are categorized using fuzzy partitions and, afterward, recommendations are made in accordance to the categorized sessions. In (Campos, Fernández-Luna, & Huete, 2008) a comprehensive approach is established, which combines fuzzy set theory with Bayesian Networks in order to represent the ambiguity or vagueness in the description of opinions provided by users.

More recent approaches make use of fuzzy logic in combination with other techniques. For instance, in (García-Crespo, López-Cuadrado, González-Carrasco, Colomo-Palacios, & Ruiz-Mezcua, 2012) semantic technologies and fuzzy logic are combined in a recommender system for investment portfolios. Recommendations are based on both psychological aspects of the investor and traditional financial parameters of the investments.

3. Recommendation framework

This section introduces the recommendation method to be validated in a tourist recommender system. It is composed by several data mining algorithms; however, the key aspect of this method is the join use of classification based on association and fuzzy sets. This combination aims at composing a hybrid method taking advantage of the strengths of both collaborative filtering and content-based approaches. In this way, the quality and effectiveness of the recommendations can be improved. We have developed a specific algorithm, CBA-Fuzzy, for mining the fuzzy association rules that constitute the associative classification model used for recommendation. The proposal take into account the main drawbacks of recommender system, described in subsection 2.3, and offers the appropriate mechanisms to minimize their effects. This framework is an adaptation for a tourism system of a previous generic

approach, which has been empirically validated with data from MovieLens and BookCrossing databases (Lucas et al., 2012b).

The framework consists of two parts, the first one encloses the process of building the recommendation models and the second corresponds to the recommendation process, which makes use of the models built previously. The first part process encloses two stages, generation of users' groups and rule set induction, which are carried out off-line, before the user accesses the system. The second part is in charge of classifying the active user at recommendation time in order to provide him personalized recommendations. In the next subsections these components are described.

3.1. Building the recommendation models

The recommendation framework includes aspects from collaborative filtering and content based methods since the recommendations to a specific user are made by comparing his preferences with the ones of other users but also taking into account features of users and products. These aspects are enclosed in the recommendation models which are built in two stages in the first part of the framework. The process is represented in Fig. 1 by means of two activity diagrams.

The first stage of the method consists on building groups of users with similar preferences and characteristics. In subsequent stages the active user is classified in one or more of these groups in order to make him recommendations according to his profile.

A clustering algorithm is applied to built the groups of users by using attributes containing demographic information about users (such as age, postal code and level of education) and also attributes concerning items to be recommended, which users have rated or purchased. Additionally, users' past interactions with the system by means of implicit actions (such as time spent seeing an item and number of mouse clicks) may be taken into account. In that sense, this process may be considered as a collaborative filtering approach to provide recommendations. The information about user preferences comes from the transactions they have carried out in the system. The examples provided as input to the clustering algorithm are formed by these transactions and the corresponding attributes from users and items.

After applying the clustering algorithm, a set $G = \{g_1, g_2, g_3, \dots, g_N\}$ of users' groups is obtained, where N is a predefined number of groups which may be set according to the number of users and items available in a particular recommender system. The set G is provided as input to the next step of the first stage of the method, which is responsible for assigning an ordered list of items (or products) $P = \{p_1, p_2, p_3, \dots, p_m\}$ to each group g_i , where $i \in \{1, 2, 3, \dots, N\}$. The top items in each list will be the ones who better represent each group; therefore, an ordination criterion must be established. The top items may be the ones who received better evaluation from the users of the group, or the most frequent ones (taking into account the number of purchases or given ratings) or any other criterion defined by an expert in the domain area involving the system. We consider the items' frequency by means of counting the number of accesses to the items in the group. The ordered list of items assigned to the user groups will be supplied as input to the recommendation process, which constitutes the second part of our method.

The second stage in the construction of the recommendation models is the induction of the associative classification rules by means of the CBA-Fuzzy algorithm (Lucas et al., 2012b). This algorithm is used to generate the rules composing the classification model employed for making recommendations. The main aspect of the algorithm is the combination of associative classification and fuzzy sets, which can provide important benefits. On the one hand, more reliable recommendations can be obtained since associative classification has a better behavior than other methods in

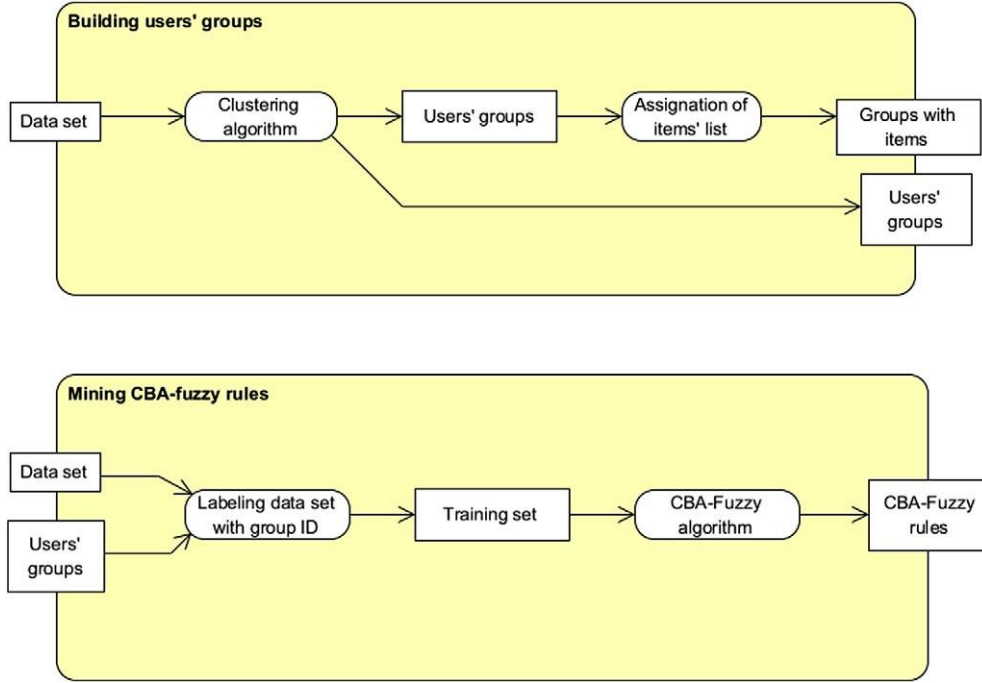


Fig. 1. Building the recommendation models.

sparse data contexts such as those from recommender systems where the number of rated products is insufficient to build the models (Lucas et al., 2012a). On the other hand, the fuzzy rules allow the classification of the user in more than one group, dealing in this way with other important drawbacks of recommender systems, such as the gray sheep problem. These rules will be responsible for classifying every new user at recommender time.

As showed in the second part of the Fig. 1, the rule generation process has two input sets: the groups of users provided as output by the clustering algorithm and the same input data set used for building the groups. The first activity for generating the list of classification rules is to combine the two inputs. At this point, we add, to the training set, the label attribute for classification. To do so, we take into account the examples composing users' groups and compare their values to the ones of the training set in order to fulfill the label attribute. Therefore, each sample of the training set will have an identification corresponding to a group of users. In this way we have a new training set, which will be the input for the CBA-Fuzzy algorithm. The output provided by the algorithm will be a set of classification rules $R(g_i) = \{r_1, r_2, r_3, \dots, r_p\}$, $g_i \in G$. Thus, the classification model will be composed of a set of class association rules available for each group of users.

Each classification rule encompasses a support and a confidence value. The confidence value expresses the degree of reliance of each rule. Therefore, before running the CBA-Fuzzy algorithm, a minimum threshold value for both measures (support and confidence) must be set up. It is recommended to set a high value for confidence and a low value for the support, especially in a scenario involving recommender systems, where we usually have sparse data and frequent itemsets might be less likely to occur. In addition, since all rules generated are considered in order to classify a new user, a rule ordering scheme is not taken into account.

3.1.1. CBA-Fuzzy algorithm

The CBA-Fuzzy algorithm (Lucas, Laurent, Moreno, and Teissiere, 2012b) is an extension of the approach of the CBA algorithm proposed by Liu et al. (1998), which is an associative classification

method consisting of two components: a rule generator (called CBA-RG) and a classifier builder (called CBA-CB). The rules' generator takes as basis the well known Apriori algorithm (Agrawal et al., 1993), hence the rules are generated from the so-called "frequent itemsets" that satisfy a minimum support threshold. Given that the mined rules are used for classification they must be "class association rules" satisfying the definition 1.

Definition 1. class association rule configuration.

Condset $\vdash y$

Where *condset* is a set of conditions evolving descriptive attributes in the dataset and *y* is a condition related to the values of the label attribute, that is, all the possible classes. An example of a rule following this definition is: $\{att1 = a\} \text{ AND } \{atr2 = b\} \vdash \{class = C_1\}$, where "a" and "b" are instances of attributes "att1" and "att2", respectively.

After obtaining the frequent itemsets, the algorithm will generate classification rules satisfying a minimum confidence threshold. Confidence is a measure that expresses the correspondence between items composing a rule. It is expressed by the occurrence frequency (percentage) of the rule among all the transactions containing the antecedent part. This measure can be obtained by means of the following definition.

Definition 2. confidence measure.

$$conf(A; B) = \frac{support(A; B)}{support(A)}$$

Where the support is a measure that assesses the frequency that the items of a rule occur in a dataset or, in other words, the number of transactions in which the items of the rule occur at the same time in the dataset.

On the other hand, the classifier builder component is responsible for producing a classifier out of the whole set of rules, which involves pruning and evaluating all possible rules. Pruning is also done in each subsequent pass of the rule generator. It uses the

pessimistic error rate based pruning method proposed by Quinlan (1993) for the C4.5 algorithm.

The Liu et al. CBA algorithm was extended in order to implement our proposal, the CBA-Fuzzy algorithm. The integration of fuzzy logic features in this algorithm consists of changing the data input format in order to deal with fuzzy values and the calculation of the support and confidence measures. Actually, the original CBA algorithm limits the input data to have only discrete numbers on attribute values and, in addition, they have to be ordered sequentially starting with the number 1. Hence, the algorithm requires a great pre-processing effort of input data, contrary to the CBA-Fuzzy algorithm that accepts any type of attribute value, even continuous or categorical attributes. To avoid the pre-processing step, the algorithm includes discretization and fuzzification processes for continuous attributes. The discretization process for numerical attributes can be done automatically by CBA-Fuzzy either using the equal-width approach, where samples are divided into a set $V = \{v_1, v_2, v_3, \dots, v_n\}$ of N intervals of the same length, or using the equal-depth approach, where the attribute range is divided into intervals containing approximately the same number of samples (same frequency).

The CBA-Fuzzy algorithm general workflow is shown in algorithm 1, where “D” is the dataset used as input for the algorithm (training set) and Df the dataset after the fuzzification process.

The line 1 of the algorithm represents the formation of the dataset “D” from an input data file. The following lines correspond to the discretization process. The second input parameter of line 2 represents the type of discretization the analyst wants to perform. Hence, the analyst can set up the number of intervals and the type of discretization he finds more suitable. In lines 4 and 6, the appropriate membership function used to perform the fuzzification process is applied according to the type of discretization selected by means of the parameter “type”. In order to calculate the membership values of a sample of a discretized dataset using the equal-width approach, a triangular membership function (three parameters) is used. For the datasets discretized using the equal-depth approach, a trapezoidal membership function (four parameters) is used, because in this case some intervals are wider than others and, therefore, they encompass a region with a constant value defining an exclusive membership. During the fuzzification process, one or two membership values are assigned to each sample of the dataset, because each sample may belong to one or two intervals at the same time. The assignment of the membership value(s) depends on the proximity of the sample value to the interval range.

Algorithm 1. CBA-Fuzzy algorithm workflow.

```

1.  D = processInput(inputFile);
2.  V = discretize(D, type, N);
3.  if (type = "equal-width")
4.    then  $D_f = \text{applyFuzzyTriang}(D, V)$ ;
5.  else if (type = "equal-depth")
6.    then  $D_f = \text{applyFuzzyTrap}(D, V)$ ;
7.  end if
8.   $CR_s = \text{CBAFuzzy-RG}(D_f)$ ;
9.   $CRM = \text{CBA-CB}(CR_s)$ ;

```

Line 8 embodies the application of CBA-Fuzzy rule generator (CBAFuzzy-RG) to the fuzzified data and line 9 the building of the classifier by means of the classifier builder (CBA-CB).

The CBA-Fuzzy rule generation process differs from the “crisp version of CBA” in the calculation of the support and confidence measures. Instead of calculating the support of an item by counting the number of transactions in which it appear (summing 1 each

time the interval it belongs to appears), the CBA-Fuzzy considers partial memberships by summing continuous values between 0 and 1 each time an interval owning (totally or partially) the item appears. Algorithm 2 contains the pseudo code of the adapted version (CBAFuzzy-RG) of the CBA-RG module previously referred, where “k-itemsets” denotes an itemset having “k” items, “Freq_k” denotes the set of frequent “k-itemsets” and “C_k” is the set of candidate “k-itemsets”.

Algorithm 2. CBAFuzzy-RG’s pseudo code

```

1.  Freq1 = {large 1-itemsets};
2.  CR1 = genRules(Freq1);
3.  prCR1 = pruneRules(CR1);
4.  k = 2;
5.  while {Freqk-1 /  $\emptyset$ }
6.    Ck = candidateItemsetsGen(Freqk-1);
7.    for all (data case  $D_i$  such that  $D_i \in D_f$ )
8.      Cd = ruleSubset(Ck, d);
9.      for all (candidateItemset Ci such that  $C_i \in C_d$ )
10.        if ( $D_i.class = C_i.class$ )
11.          for all (attribute a such that  $a \in D_i$ )
12.            lineSupport = lineSupport + a.support;
13.          end for
14.          Ci.rulesupCount = Ci.rulesupCount + lineSupport;
15.        end if
16.      end for
17.    end for
18.    Freqk = {c  $\in$  Ck | c.rulesupCount > minsup};
19.    CRk = genRules(Freqk);
20.    prCRk = pruneRules(CRk);
21.    k++;
22.  end while
23.  CRs = [k CRk;
24.  prCRs = [k prCRk;

```

3.2. Recommendation process

The models built in the previous stage are used for recommending items to the active user when he is on-line. Firstly, the model of class association rules is required to classify the active user and predict in this way the group or groups he belongs to. Since preferences may change as time goes by, the most recent interaction data of the active user is taken to do the classification. To do that, the transaction “y” corresponding to the most accessed item in the last access records is checked against the rules’ set. This transaction has the same attributes of those used to build groups of users, thus, the provided recommendations will be well-suited to his current preferences. In case of the active user has not done any transaction, the recommender procedure considers just the user attributes by comparing the values of these attributes with the ones of the groups. On the other hand, if the models have been recently updated and the active user is already classified the classification rules are not necessary.

Fig. 2 shows an activity diagram of this stage that is carried out at runtime, when the user is interacting with the system.

The transaction “y” (represented as “user last transaction” in the diagram) and the model of class association rules obtained in the previous stage, are supplied as input to the recommender process. The process starts by selecting the set $R_c = \{r_1, r_2, r_3, \dots, r_N\}$ of N rules satisfying the condition that all antecedent terms’ values are matched by the user’s last transaction attribute values. In the case of continuous attributes, a partial membership to the interval is considered as a match. If there are no rules (or very few) respecting this condition we take into account the downward closure

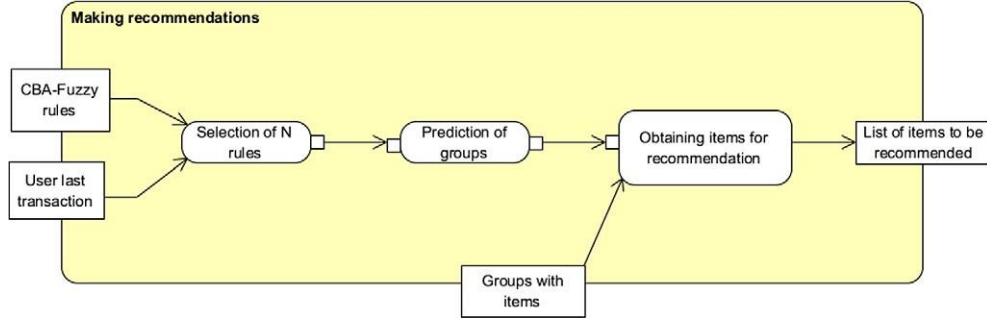


Fig. 2. Recommendation process.

property of association rules' support, which guarantees that for a frequent itemset, all its subsets are also frequent. In this way, we decrease the size of the itemset and successively verify if there are rules matching the condition stated before, in order to decrease the itemset's size until suitable rules are found. In fact, several authors like Toivonen, Klemettinen, Ronkainen, Hatonen, and Mannila (1995) and Liu, Hsu, and Ma (1999), argue that, usually, the more general rules are (the ones which encompass less terms), the more relevant and less ambiguous they are.

After obtaining R_c , the values of the label attribute (consequent term) of the rules in R_c are considered in order to obtain the possible groups (predicted groups) to which the active user owns to. At this point, we calculate his membership function to every class (group of users) found on R_c 's rules consequent terms. To do so, a discriminator function "g" is defined in order to calculate the degree of truth that the active user owns to every class found in R_c . Considering that the active user is represented by a transaction "y" and "h" represents a group of users, the discriminator function can be calculated by means of the following formula

$$g_h(y) = \frac{1}{\sum_{j=1}^k B(j,k)(y)} \sum_{j=1}^k B(j,k)(y) \cdot X(j,k)(y)$$

where k is the number of terms (attributes) in each rule, $X(j,k)(y)$ the value taken by the attribute $X(j,k)$ in the sample "y" and $B(j,k)[X(j,k)(y)]$ its degree of membership. Hence, this function calculates the product of attributes' degrees of membership for all the rules in R_c and then sum of the results obtained in each rule.

When the discriminator values for each class (or group of users) are obtained, the system compares them in order to find the greatest. In this sense, this method is different from most fuzzy associative classification approaches since they usually predict just one class label for a given instance. Nevertheless, our proposal can consider several classes, which are the ones satisfying a minimum discriminator threshold previously established. Therefore, we have "t" groups of users related to the active user. Once the active user is classified, the recommender process makes use of the sets of items assigned to the users' groups, which are generated in the first part of the framework and provided as input to the second part in charge of making recommendations. Given that each group g_i is associated to a list of items and each user is associated to one or several groups, the recommendation presented to the active user is a suggestion involving the " n_i " best ranked items from the lists of each group the user belongs to. The final list of recommended items is made by joining the previous lists. In order to have a constant number of recommended items, " n_i " will be inversely proportional to the numbers of groups "t", therefore, the more classes we have the less items are considered in each list.

As commented before, for a new user without transactions, only the user attributes are considered. In this way, the more suitable group to the user is found and then the most accessed item in this group is verified in order to compose the transaction "y" and continue the recommender process as usual.

On the other hand, if the active user data is in the training set (he belong to a group), the classification rules are not necessary since he is already classified.

4. The PSIS system

In this section we describe the system used to validate the method. Coelho et al. (2009) developed the Personalized Sightseeing Planning System (PSiS) aiming at aiding tourists to find a personalized tour plan in the city of Oporto, Portugal. The system helps the active user to spend his time efficiently and to promote local tourism and culture. The main focus of this work was at tour planning support, where a personalized visiting plan is provided to the active user. Each visiting plan tries to select the most adequate tourism items, which authors referred to as "points of interest", according to the user profile and also finds the available transportation modes among the select items. The PSiS system can be used by city councils in order to better explore their tourism resources and give tourists a personalized service, which can be very attractive and improves tourism service offer Coelho et al. (2009). The application of the method we want to validate in PSiS can enhance the system personalization and, therefore, explore even more tourism resources. This method was added in PSiS as an extra feature, which provides to the active user, a list of recommended points of interest. In the next subsection we describe the general architecture of PSiS.

Afterwards, in subsection 4.2, we show an overview of the taxonomy about the points of interest. Finally, in subsection 4.3,

we describe how the recommendation method was included in the system, as well as the user modeling process, which is intrinsically related to the recommendation mechanism.

4.1. General architecture

As described before, the PSiS aids the user to plan what to do in a specific place. To do so, the system is divided in three main components: Community Tags, Trips and Points of Interest. The first consists of a set of tags representing types of points of interest, which are graphically differentiated according to their level of relevance (i.e. frequency of visits by the whole community). The second (Trips) is composed of route recommendations, where each trip contains multiple tours. In this context, a tour refers to multiple activities in a certain destination and, therefore, a tour refers to multiple points of interest. Thus, the system allows the user to create (and to schedule) trips containing several points of interest to visit. In addition, after some user interactions, the system may suggest some trips to the user, where functioning and transportation schedules are considered. Therefore, route planning in PSiS can join points of interest with transportation alternatives and schedules, providing detailed planned itineraries for the tour plans generated to the user. In order to develop the algorithms responsible for the

tour planning tasks, the system employed a variation of the Traveling Salesman Problem (TSP) method, with additional constraints, as well as the Prize Collecting Traveling Salesman Problem (PCTSP). However, in this work we will not deepen in this question, since our validation is performed just on the third main component of PSiS (Points of Interest). In this subsection, we focus on the description of this component. Moreover, in order to schedule and suggest trips, the system needs to get some information related to interactions between points of interest and users.

In this context, Fig. 3 shows a PSiS's screen capture when the "Points of Interest" component is accessed. The graphical components mentioned above can also be visualized and accessed. The links to "Points of Interest" and "Trips" are on the top left and the link to "Community Tags" is on the bottom right corner of the screen.

As we can see on Fig. 3, the graphical interface of PSiS, particularly the Points of Interest component, allows user to visualize the available points of interest or to receive recommendations containing some of them. On the bottom left of the screen the points of interest are divided on categories. The "Igreja de Santo Ildefonso" point of interest belongs to the "Churches" category and to the "Religion" category. On the middle area of the screen the active user can click on any of the points of interest displayed and see its detailed information.). If the user gets interested in visiting the point of interest after seeing this information, he can add the point of interest to his "tour basket".

4.2. Points of interest's taxonomy

Currently, the PSiS database contains information provided by the municipality of Oporto (Câmara Municipal do Porto), from where it was gathered data related to the 241 points of interest currently available on the system. It is important to highlight that the current version of PSiS is a prototype, so that, in future more points of interest, as well as more detailed information, may be added.

Since there are more than two hundred heterogeneous points of interest, a taxonomy was defined in order to divide and organize them in categories. As we can see on Fig. 4, in the taxonomy, adapted from (Coelho, Martins, & Almeida, 2009), there is a root division between physical places and events. However, in this work we will not consider the points of interest belonging to the "Events" class, because the recommender method does not consider any sequential or time variable regarding events that happen only during a certain period of time.

According to the hierarchy presented on Fig. 4, the "Places" category encompasses eight subcategories, where "Religion" and "Cultural" own significantly more points of interest than the other six.

Despite the fact that a point of interest's category may be one over fourteen, every point of interest has the same data configuration. In order to apply the method to be validated, we used five points on interest's attributes, where two of them are continuous (average cost in Euro and average duration in minute) and the other ones are binary Yes or No attributes (indoor place, city's monument and visiting time at night). Hence, only "average cost" and "average duration" attributes need to be discretized and fuzzified.

In this context, the fuzzy attributes take an important role in the recommendation process, because the membership values defined on the fuzzification process will be responsible for defining the groups the user belongs to. Along with these two attributes, the user attributes are also considered in the recommendation process. In the next subsection, we describe how recommendations are performed on PSiS, as well as the user data employed in this process.

4.3. User modeling and recommendation

In the models proposed in the literature two groups of factors influencing destination choice can be identified: personal features and travel features (Ricci, 2002). In the context of this work, personal features refer to user data and travel features refer to data

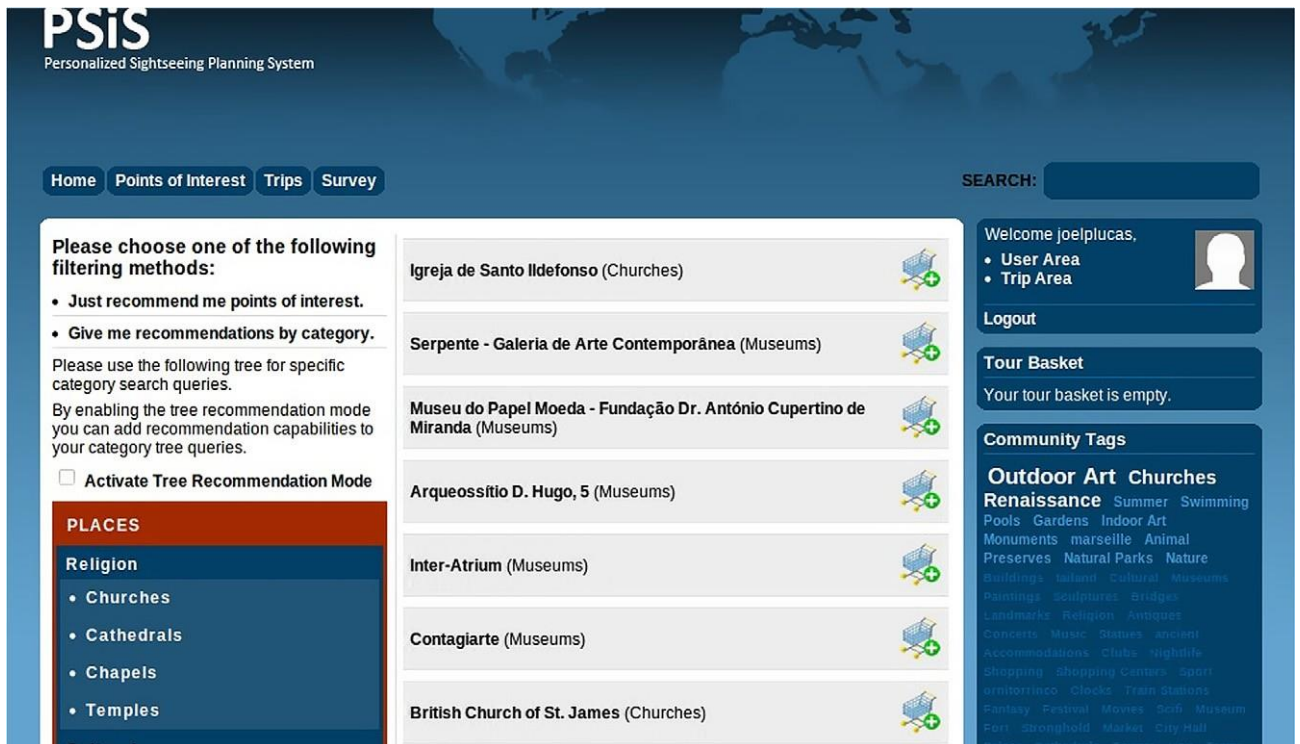


Fig. 3. The PSiS graphical interface.

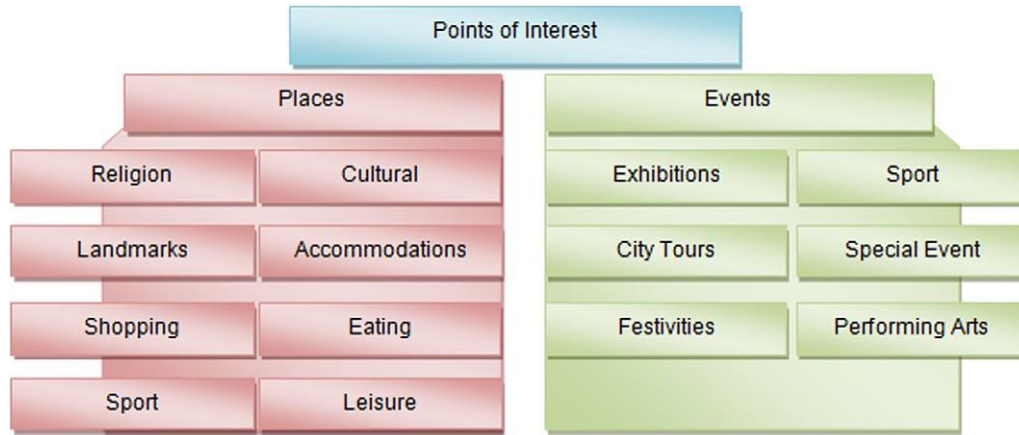


Fig. 4. Points of Interest Taxonomy on PSiS.

about points of interest. Hence, in order to provide recommendations, PSiS needs to evaluate both points of interest information and user information. To do so, a model architecture for structuring user information was defined, which is based on the “Student User Modeling Architecture” proposed by Benyon and Murray, (1993). This architecture separates user information elements in an information hierarchy. Although in that example the architecture was used with the purpose of modeling Educational Adaptive Hypermedia (EAH) users, it was found to be perfectly usable in a variety of other situations, such as tourism (Coelho et al., 2009). The architecture organizes user data into two main modules: Domain Dependent Data (DDD) and Domain Independent Data (DID).

The first module involves user information which is particularly related to the system domain. This information is usually gathered by means of an information retrieval technique. Fink and Kobsa (2002) described some types of information that user models in tourism-based systems have been dealing with recently. Among these types of information, PSiS employs one type which refers to past user interactions with the system, such as past trips, past item visits and other kinds of past events. In the use of the proposed method, user accesses to points of interest were taken into account in order to define a recommendation model.

The second module from the architecture mentioned above (DID) involves static user information, which is not related to user interactions on the system. This module is divided into two types (Generic Profile and Psychological Profile), however, in this work we only consider the Generic Profile type, since it is the kind of information usually available in recommender systems. This information from DID is related to user’s personal data (user’s name, e-mail, system username), demographic data (gender, marital status, age, etc.), academic background, work experience, etc. The user’s demographic data has a direct relationship with user modeling and especially with the recommender mechanisms employed. Therefore, it is used for applying the machine learning techniques implemented in PSiS as a part of the recommendation method.

Despite having a comprehensive user information structure, recommendation’s quality may be affected by the diverse nature of items available. The active user may visualize more than two hundred points of interest available in PSiS, therefore he won’t probably access to items of his interest and, consequently, will waste his time accessing to points of interest he would never visit in a tour. In order to aid user in the exploration of the available points of interest and, therefore, to make him focus his attention on points of interest he would probably visit, the recommendation mechanism of PSiS offers two types of recommendation: categories of points of interest or single points of interest. In this work, we will focus on the second type of recommendation provided in PSiS,

since it encloses a greater level of personalization and the proposed method is designed for this kind of recommendations. Fig. 5 shows a screen capture of the results related to recommendation of specific points of interest. It is referred by the system as “Just recommend me points of interest” while the other type is referred as “Give me recommendations by category”.

Fig. 5 is analogous to Fig. 3, however the scenario presented is different, as the points of interest shown are different. The recommendation is based on user profiles and item characteristics (using the proposed method) instead of merely showing them randomly. In this way, the user is more likely to add diverse points of interest to the tour basket. The points of interest are shown in an ordered list based on the results provided by the recommendation method.

5. Evaluating recommendations in PSiS

This section includes the validation of the recommender method by testing its recommendation mechanism on the PSiS’ functionalities. We apply and analyze the results of this method on the option of recommending single items (the “Just recommend me points of interest” option shown in Fig. 5).

5.1. The dataset

The dataset employed for evaluating the method was obtained from PSiS and its users. The method firstly uses information about the 241 points of interests available in the system. This type of information was already discussed in subsection 4.2 and was included in the system on the moment it was first launched. Therefore, we used the same six attributes about points of interest described in subsection 4.2.

On the other hand, the data obtained through PSiS’ users is constantly changing, as new users may enroll in the system. On the moment we built the recommender model (applying the algorithms and techniques described in Section 3), PSiS encompassed 20,924 users and 25,545 accesses.

According to the approach considered in (Rittman, 2005), the density correlation of this dataset was 0.116, obtained through the division of the product of distinct values of its attributes by the number of records. Therefore, this dataset may be considered sparse, because non-sparse datasets, like MovieLens, present a correlation value close to 1.

5.2. Recommender modeling

After setting up the input dataset, PSiS runs the components of the method in analysis. In this subsection, we describe the genera-

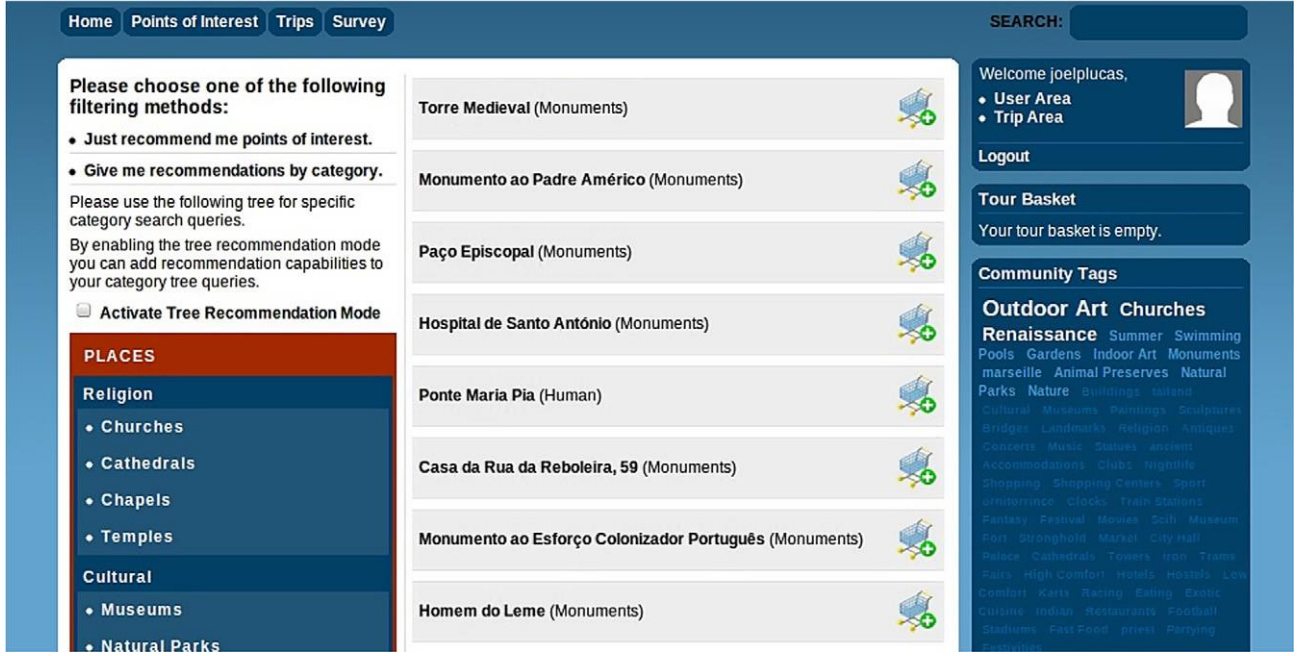


Fig. 5. Recommended Points of Interest.

tion of the recommender model. Since user data, and eventually points of interest data as well, change as time goes by, the recommender model used for providing recommendations at runtime needs to be updated frequently. This update is programmed to be performed when a predefined threshold of new records are available in the system. This threshold is defined by the system's administrator according to the server's capacity, where initially we set the threshold to the same value of the number of rows of the input dataset (25,545). The recommender model generation is performed automatically by PSiS, without the need of human interaction.

According to the structure of the recommender framework described in Section 3, the first component to be ran in order to generate the recommender model is the adapted version of K-Means also implemented in Java. This algorithm employs an unsupervised learning approach in order to obtain a set of groups of records as output. These groups reflect common patterns of user accesses. Thus, the output provided by this component is a set $G = \{g_1, g_2, g_3, \dots, g_N\}$, where N is the number of groups. The value of N can be obtained by the own clustering algorithm, for instance by using the iterative k-means algorithm, or can be established by the system's administrator taking into account his domain knowledge and the number of records in the database. In order to obtain clusters with a representative number of examples, in this study N was initially set to 25, because the number of examples was approximately 25 thousand. In this way, the number of groups followed a correlation of 1/1000, and we can assume that relevant user access patterns can be extracted from at least every thousand examples. In this context, a group g_i is represented by a record with the same attributes of the input dataset. Table 1 contains the first four groups obtained after running the method's adapted version of K-Means.

We can see that the values of the groups' attributes are the mean value of all records in the group, for the continuous attributes (avgDuration, avgCost and dateOfBirth), and the most frequent value for the categorical attributes.

After building the groups and before running the CBA-Fuzzy algorithm, an ordered list of points of interest is associated to each group (as shown in Fig. 1). This list, $P = \{p_1, p_2, p_3, \dots, p_m\}$ is assigned to each group generated, where "m" is the total number of items (241). At this moment, the system counts the number of

times each user has accessed every point of interest. Since the results of K-Means let us know what group each user of the input dataset belongs to, it is possible to know the sum of accesses of all users of each group to every point of interest. Table 2 includes the first four items of a cluster, where the counter attribute represents the sum of accesses of all users of a certain group to a certain point of interest.

Table 2 shows the four most accessed points of interest by the group number 24. The counter attribute represents the sum of accesses of all users of the group to a certain point of interest identified by pointID. The top points of interest on every group will be used at recommendation time for providing recommendations to a new user (i.e. a user who is not in the input dataset). However, to do so the new user needs to be previously associated to one or more groups. Therefore, the rule generation component of the method is executed.

The first task of the rule generation process (as shown in Fig. 1) is to prepare a training set for the classification algorithm. The training set is the input dataset used in the previous module but extended with one more attribute, the group, which is the label attribute used for classification. The association of a group to every record of the initial dataset is made according to the results obtained by the application of the K-Means algorithm. Table 3 shows four examples of the training set provided to CBA-Fuzzy.

As we can see in the Table 3, the input dataset was extended by means of the addition of the class attribute. The values of the continuous attributes are neither discretized nor fuzzyfied, therefore, the first task of CBA-Fuzzy algorithm is to discretize and fuzzyfy the training set (as shown in algorithm 1).

With regard to the discretization process of the continuous attributes (dateOfBirth, averageDuration and averageCost), these attributes were discretized in four distinct intervals with equal frequency. The discretization process yielded the following intervals: date Of Birth: [1933–1956], [1956–1972], [1972–1983] and [1983–2001]; average Duration: [0–20], [2–30], [30–60] and [60–240]; average Cost: [0–1], [–7], [–20] and [20–150].

Subsequently, by means of the results supplied by the discretization and fuzzyfication processes, the training set remains with an appropriate format for being provided to the rule generation com-

Table 1
Obtained groups.

clusterid	AvgDur.	avgcost	isindoor	IsMon.	IsNotur.	dateofbirth	Gender	MaritalSt.	Country	Religion
1	46.57	6.27	0	1	0	1957	Male	Married	S.Africa	Islam
2	18.72	1.91	0	1	0	1984	Male	Single	UK	Islam
3	17.28	0.97	0	1	0	1976	Male	Married	Angola	Judaism
4	49.16	0.69	1	0	1	1930	Male	Divorced	USA	Budism

Table 2
Frequent points of interest in groups.

clusterid	pointid	Counter
24	1890	12
24	1629	10
24	1884	8
24	1633	4

ponent of CBA-Fuzzy. The first four examples of this training set are shown in the Table 4, where some values of the continuous attributes encompass two intervals with their respective membership values correlated to each record.

If we focus on the attribute dateOfBirth of the third row, we can observe that attribute values of this group belong in a 61% to [1972–1983] and a 50% to the [1983–2001] interval. That means this sample is close to the left border of the first interval. Then, this training set is provided as input to the CBAFuzzy-RG component (whose code is presented in subsection 3.2). For running this component, we set up a low support (1%) and a high confidence (75%) since the data has a sparse nature (its density correlation value is only 0.116).

After running CBA-Fuzzy, the output of the rule generation process on this dataset consist of 189 classification rules, where the maximum number of terms is six and the minimum is four. The ones with higher confidence have priority for classifying the active user. Below, two classification rules obtained through this process are shown:

R28: $\text{fav_duration} \frac{1}{4} \geq 30 \quad 60\% \text{ AND } \text{fdm_monument} \frac{1}{4} 0 \text{ AND } \text{fdm_indoors} \frac{1}{4} 1 \text{ AND } \text{fgender} \frac{1}{4} \text{Female} \text{ AND } \text{fmarital_status} \frac{1}{4} \text{Divorced} \text{ AND } \text{freligion} \frac{1}{4} \text{Atheism} \rightarrow G8; \text{Conf } \frac{1}{4} 95.03\%$

R56: $\text{fdm_monument} \frac{1}{4} 0 \text{ AND } \text{fdm_visiting_time} \frac{1}{4} 1 \text{ AND } \text{fgender} \frac{1}{4} \text{Male} \text{ AND } \text{fmarital_status} \frac{1}{4} \text{Divorced} \text{ AND } \text{fcountry} \frac{1}{4} \text{UnitedStates} \rightarrow G4; \text{Conf } \frac{1}{4} 89.75\%$

Table 3
Training set provided to CBA-Fuzzy.

AvgDur.	avgcost	isindoor	IsMon.	IsNotur.	Dateofbirth	Gender	MaritalSt.	Country	Religion	Class
10	0	0	1	0	1972	Male	Divorced	Brazil	Islam	G17
60	0	1	0	0	1972	Male	Divorced	USA	Islam	G4
30	0	0	0	1	1980	Male	Single	China	Hiduiism	G7
10	0	0	1	0	1987	Female	Single	S.Africa	Hiduiism	G15

Table 4
Input dataset view.

AvgDur.	Avgcost	IsInd.	IsMon.	IsNotur.	Dateofbirth	Gender	MaritalSt.	Country	Religion	Class
[0–20]:1.00	[0–1]:1.00	0	1	0	[1956–1972]: 0.50; [1972–1983]:0.50	Male	Divorced	Brazil	Islam	G17
[30–60]:0.50:[60–240]:0.50	[0–1]:1.00	1	0	0	[1956–1972]: 0.50; [1972–1983]:0.50	Male	Divorced	USA	Islam	G4
[20–30]:0.50:[30–60]:0.50	[0–1]:1.00	0	0	1	[1972–1983]: 0.61; [1983–2001]: 0.39	Male	Single	China	Hiduiism	G7
[0–20]:1.00	[0–1]:1.00	0	1	0	[1972–1983]: 0.80; [1983–2001]: 0.20	Female	Single	S.Africa	Hiduiism	G15

Rules 28 and 56 own six and five attributes respectively, plus the label attribute. Information related to classification rules, including the confidence measure value, is stored in the database of PSiS.

5.3. Testing the recommender model

In this subsection we evaluate the recommendation phase of the method we want to validate. To do so, we took into account an active user U1 and simulated his behaviour in the system in order to deploy the recommendation process performed online.

Therefore, firstly we need to obtain a transaction y_1 to represent U1's interaction in the system. This step is necessary if the user has not been classified yet or the models have not been updated recently. In this case, we consider the points of interest that U1 most accessed on the 30 last access records in the PSiS database. This number of transactions can be lesser if the user is relatively new and he still does not have enough accesses. In this case all user transactions are taken into account. If several points of interest have the maximum number of accesses, then y_1 will point to the last one. In this case study, the system gathers a point of interest, which is a monument and appears 7 times on the last 30 U1's transactions. This information is detailed in Table 5.

The data of y_1 has the same configuration (same attributes) of the datasets detailed previously. Moreover, it refers to current behaviour of U1 in PSiS, which is a crucial requirement for obtaining quality recommendations if the models are not frequently updated, because user interests may change significantly along the time and a tourist may visit other places and change his preferences and tastes or even other life experiences may make him change his mind.

After obtaining the y_1 transaction, the system tries to find a set $R_c = \{r_1, r_2, r_3, \dots, r_N\}$ of “N” rules in which all antecedent terms match the data of y_1 . For instance, the rule number 28 (which was shown in the previous subsection) is matched in four terms by y_1 : avg_duration (partial membership of 0.5), dm_indoors, gender and religion. However, among all discovered rules, only three of them were matched by at least four terms. In this way, the downward-closure property was applied and the system found

Table 5
Data of transaction $y1$.

AvgDur.	Avgcost	IsIndoor	IsMon.	IsNotur.	Dateofbirth	Gender	MaritalSt.	Country	Religion
30	5	1	1	0	1983	Female	Married	Brazil	Atheism

Table 6
Top three groups.

ID	AvgDur.	Avgcost	IsInd.	IsMon.	IsNotur.	Dateofbirth	Gender	MaritalSt.	Country	Religion	Memb
9	18.49	1.39	0	1	0	1981.4	Male	Married	Portugal	Judaism	8.5
21	121.89	15.86	0	0	1	1958.1	Male	Divorced	USA	Judaism	6
25	50.77	0.67	1	0	0	1975.9	Female	Divorced	Venezuela	Budism	5.79

47 rules, with at least three terms, all of them matching attributes' values of $y1$.

Afterward, the values of the label attribute (consequent term) of rules in the set R_c are considered in order to calculate the membership degree to every group of transactions in G . On the continuous attributes, the membership degree is calculated on every continuous attribute. For instance, the avg_duration value for $U1$ is 30, therefore, it matches with rules encompassing the avg_duration term with both $[0-30[$ and $[30-60[$. In this case, the participation of $U1$ is of 0.5 on rules encompassing $[0-30[$ on the avg_duration term and also of 0.5 on rules encompassing $[30-60[$ on the same

term. Hence, after calculating, by means of the discovered classification rules, the membership degrees of $y1$ to the groups, we obtained eight groups with membership degree greater than zero. In order to make the list of recommended points of interest only the top n groups with a membership degree greater than a given threshold are considered. In this case, we obtain three groups with a membership degree greater than the threshold 5. They are showed in Table 6 where the last column contains the membership degree.

After obtaining the three most correlated groups to the active user's last transaction, the system consults the most accessed points of interest in each of the three groups shown in Table 6 with the aim of obtaining the items' list associate to each group. Given that the final list presented to the user contains a constant number of the best ranked items in each group, according to the procedure described in the subsection 3.2, the number of items in the individual list of each group is inversely proportional to the number of groups. In this case, the number of items in the final list was settled as 8, thus we selected the three most accessed points of interest for

the first and the second group (clusterID = 9 and clusterID = 21), and two for the group with lowest membership degree (cluster_id = 25). Table 7 shows the result of this selection.

Fig. 6 shows the result of the recommendation described in this subsection. It is important to highlight that the top three points of

Table 7
Frequent points of interest in groups.

Clusterid	Pointid	Counter
9	1521	31
9	1503	25
9	1491	19
21	1574	28
21	1537	20
21	1573	18
25	1510	37
25	1566	31

Please choose one of the following filtering methods:

- Just recommend me points of interest.
- Give me recommendations by category.

Please use the following tree for specific category search queries.
By enabling the tree recommendation mode you can add recommendation capabilities to your category tree queries.

☐ Activate Tree Recommendation Mode

PLACES

Religion

- Churches
- Cathedrals
- Chapels
- Temples

Cultural

- Museums
- Natural Parks

Torre Medieval (Monuments)



Monumento ao Padre Américo (Monuments)



Paço Episcopal (Monuments)



Hospital de Santo António (Monuments)



Ponte Maria Pia (Human)



Casa da Rua da Reboleira, 59 (Monuments)



Monumento ao Esforço Colonizador Português (Monuments)



Homem do Leme (Monuments)



Fig. 6. Recommendation provided to $U1$.

interest recommended are related to the three groups respectively (i.e. “Torre Medieval” is the top item on group number 9, “Monumento ao Padre Américo” is the top on group 21 and “Paço Episcopal” is the top on group 25). In this way, the top points of interest displayed on the screen will be the top items among all groups. The subsequent points of interest displayed on the screen are also intercalated according to the second top item on each group.

5.4. Empirical analysis

In this subsection we evaluate how typical recommender systems drawbacks (sparsity, scalability, first-rater and gray sheep problem) are managed with the use of the method. To do so, we simulate typical scenarios in which these drawbacks are likely to occur. At this moment, we give special attention to the first-rater and gray sheep problems, since sparsity is already intrinsically related to the data of PSiS and scalability is not taken into account in this context because the recommender model is built off-line.

In order to simulate the grey-sheep problem, we created a user U2 whose data is composed of the less frequent values in the database. Along with U2’s data we selected the less accessed point of interest (“Barcadouro”, with only 7 accesses) among all groups in order to compose the transaction y_2 . In this way, y_2 encompasses non-frequent information, and it will certainly be difficult to find similar records to be related with. In Table 8, we detail the data of transaction y_2 . As the dateOfBirth attribute was discretized using equal frequencies approach and then we took the lowest value (1933) of the less frequent interval ([1933-1950]).

Subsequently, we proceed in the same way we did in subsection 5.3, so that we need to find the set $R_c = \{r_1, r_2, r_3, \dots, r_N\}$ where all antecedent terms of the “N” rules match the data of y_2 . After applying the downward-closure property, the system found 23 rules encompassing at least three terms and matching all attribute values of y_2 . Not surprisingly, in this scenario the system found significantly less rules (half) than in the scenario described in subsection 5.3.

The same trend was repeated when the label attribute of rules was considered in order to calculate the membership degree to groups in G. Despite finding almost the same number of groups (seven) with membership degrees greater than zero, the obtained degrees of membership were significantly lower than the ones in the previous scenario. Table 9 shows the top three groups matching y_2 .

After comparing Tables 6–9, we can observe that the highest membership degrees found for y_2 were lower than the ones found for y_1 , however, we can conclude that, despite y_2 has presented weaker membership to groups, it is still related to them with a reasonable degree of membership (at least of 1.37) and U2 will be able to receive recommendations. Hence, we may conclude that the effects of the gray-sheep problem did not significantly affect the

Table 10
U3 data.

dateofbirth	Gender	MaritalStatus	Country	Religion
1983	Female	Married	Brazil	Atheism

recommendation quality, as the model is based on strong classification rules holding high confidence.

In order to simulate the first-rater problem, we created an active user U3 with the same data of U1. However, U3 will not access any point of interest and will ask for recommendations. Thus, there is no information about any point of interest for composing a transaction y_3 . In order to solve this problem, the system considers just the user attributes. Table 10 shows U3’s data.

Using the data in Table 10, the system finds (by comparing the attributes values directly) the most suitable group for U3, which is shown in Table 11.

After knowing what group U3 is more similar to, the system finds the most accessed point of interest within this group. Afterwards, the data of this point of interest is joined to U3 data in order to compose the y_3 transaction, which will have the same configuration of y_1 and y_2 . From this moment, the system continues the process in the same way as for the two previous scenarios.

In this way, U3 receives a recommendation in the usual way, and therefore, the first-rater problem does not prevent U3 from receiving recommendations, even if he had just recently enrolled the system. Thus, the quality of the provided recommendation is not significantly affected.

5.5. Satisfaction survey

In order to evaluate the method implemented in PSiS, we made a satisfaction survey with the PSiS’ users regarding the recommendation they received. This survey consists of four multiple choice questions, which have five possible answers: totally disagree, mostly disagree, neutral, mostly agree and totally agree. Fig. 7 shows a screen capture of part of the survey described in this subsection.

We specify the four questions of this survey, which, actually, are assumptions that users will appraise:

- Question 1: The recommendation provided was useful/interesting.
- Question 2: I am interested in visiting the recommended points of interest.
- Question 3: I became interested in more points of interest after receiving the recommendation.
- Question 4: The recommendation process was boring or inconvenient.

Table 8
Data of transaction y_2 .

AvgDur.	Avgcost	Isindoor	IsMon.	IsNotur.	Dateofbirth	Gender	MaritalSt.	Country	Religion
240	100	0	0	1	1933	Female	Divorced	Australia	Atheism

Table 9
Top three groups related to y_2 .

ID	AvgDur.	Avgcost	IsInd.	IsMon.	IsNotur.	Dateofbirth	Gender	MaritalSt	Country	Religion	Memb
5	51.26	0.73	1	0	0	1984.1	Male	Divorced	Venezuela	Budism	5.7
20	50.52	0.67	1	0	0	1946.2	Female	Married	Antartica	Christianism	3
25	50.77	0.67	1	0	0	1975.9	Female	Divorced	Venezuela	Budism	1.37

Table 11
Most suitable group for U3.

Clusterid	AvgDur.	Avgcost	Isindoor	IsMon.	IsNotur.	Dateofbirth	Gender	MaritalSt.	Country	Religion
18	51.49	0.67	1	0	0	1963.8	Male	Married	Brazil	Atheism

Survey about the recommendation process on the PSiS system

This survey consists of an evaluation of the recommendation process provided on the PSiS system at section "Just recommend me points of interest"

The recommendation provided was useful/interesting

- ☐ Totally disagree
- ☐ Mostly disagree
- ☐ Neutral
- ☐ Mostly agree
- ☐ Totally agree

I am interested in visiting the recommended points of interest

- ☐ Totally disagree

Fig. 7. Screen capture of the survey.

The survey was shared in social networks and was answered by 105 users with diverse profiles. Fig. 8 shows the distribution of the categorical attributes of the 105 users who answer the survey. In this figure users are classified in men (represented in blue color) and women (represented in red).

We may observe in figure that around 60% of the system users are men. The distribution of the MaritalStatus attribute reveals that 62 users are single, 37 are married and 7 are divorced. The Religion attribute shows that the majority of the users are catholic (62), but there is also a great number of atheists (27) and Jewish (12).

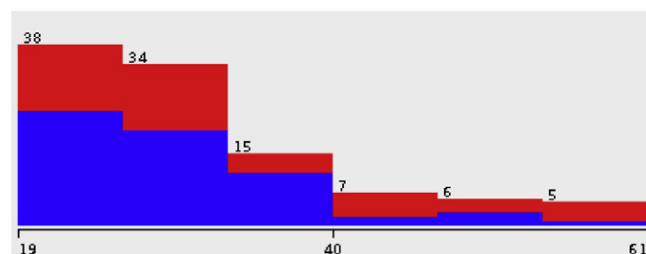


Fig. 9. Distribution of the Age Attribute.

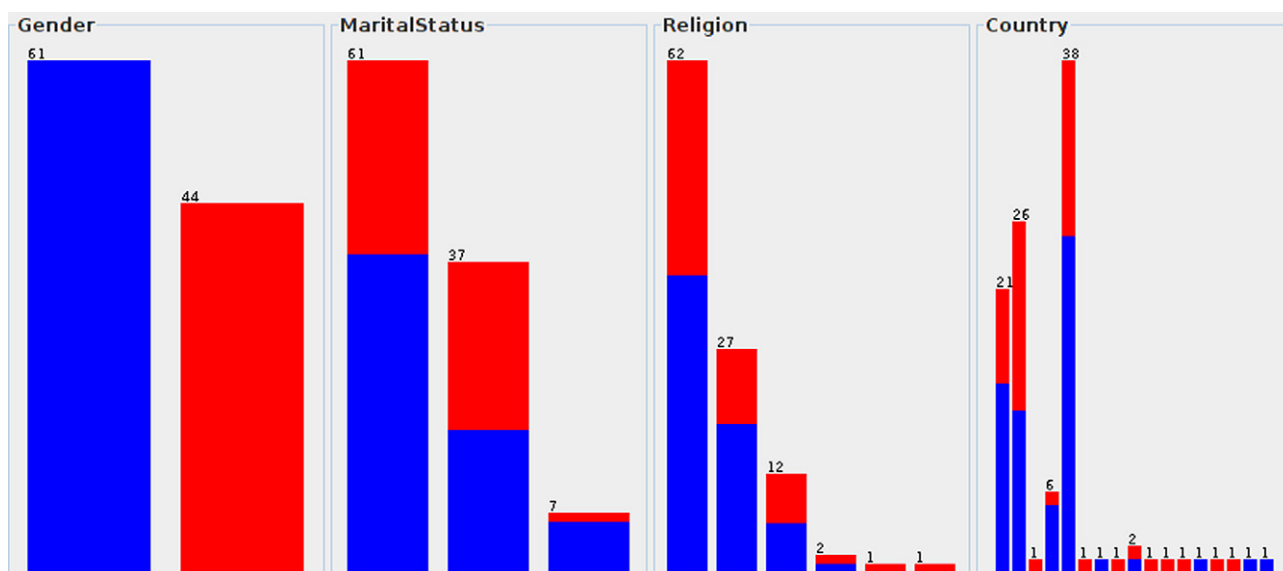


Fig. 8. Categorical attributes' distribution of users' data.

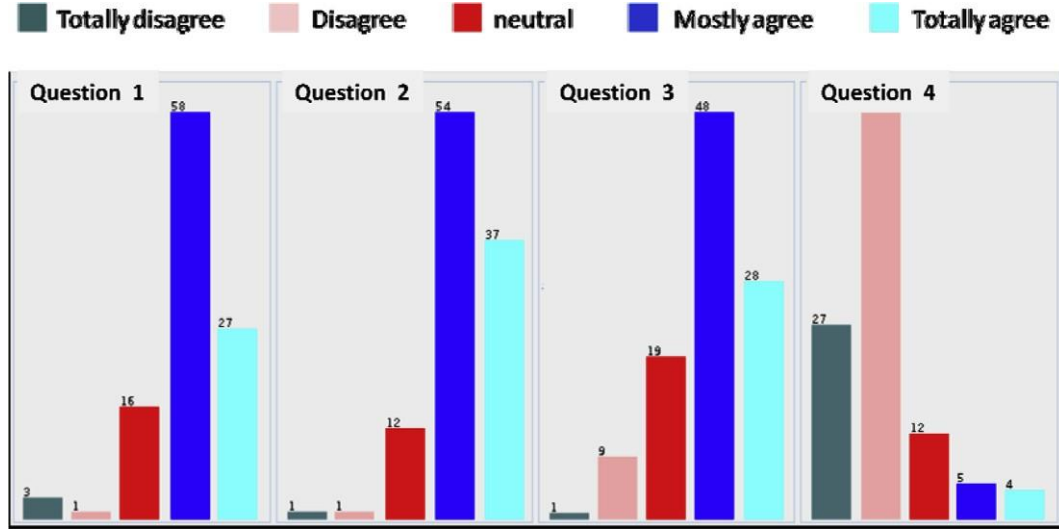


Fig. 10. Distribution of the attribute related to users appraisals.

Finally, the Country attribute encompasses users of, basically, four countries: Spain (38), Portugal (26), Brazil (21) and Colombia (6).

Fig. 9 details the distribution of the attribute related to users' age, which is specified on the horizontal line depicted on the bottom part of the figure. Frequencies are depicted on the top of each interval.

This figure shows that the majority of the users belong to the interval between 19 and 40 years old. Besides, we may observe that oldest user is 61 years old and that the average age of users is approximately 32.

Taking into account that these users answered the four questions reported previously (respecting the same attribute order), in Fig. 10 we detail the number of users who answered each of the five possible answers: totally disagree, disagree, neutral, mostly agree and totally agree.

Through this figure we may notice that question 4 does not follow the same distribution tendency as the other questions, because it involves a negative assumption concerning the recommendation process (the recommendation process was boring or inconvenient), to which 57 users (more than 50% of all users) expressed that they do not agree the recommendation is inconvenient. Besides, 27 users (approximately 30% of all users) expressed that they totally disagree the process is inconvenient. On the other hand, in questions 1 and 2, 85 and 91 users, respectively, evaluated positively (totally agree or mostly agree) the recommendation and the points of interest it suggested to them. Moreover, 76 users considered (totally agree or mostly agree) that the recommendation was even a determinant factor for calling their attention to some points of interest available on the system. Therefore, we may conclude that most users have evaluated positively the whole recommendation process, which they considered useful and interesting.

6. Conclusions

The method proposed in this work aims at avoiding several shortcomings of current recommender systems as scalability, sparsity, first-rater or gray sheep problems. Its implementation in a tourism system and the case study carried out show that the techniques enclosed in the method, fuzzy logic and classification based on association, can be effectively applied to recommender systems.

On the one hand, scalability drawback is avoided since the proposal is a model based approach where recommendation models

are induced off-line, unlike the memory based methods. Therefore, time spent in building the models does not influence the user response time.

Given that the model induction time is not a problem in model-based collaborative filtering and the superior behavior of the CBA associative classification algorithm with sparse data, we argue that CBA is very appropriate to be employed in recommender systems as well as its extension for encompassing fuzzy sets' features. The proposed CBA-Fuzzy algorithm includes this extension keeping the foundations of CBA original algorithm. In that way, the sparsity problem affecting seriously to recommender systems is addressed since associative classification methods, specially CBA, are less sensitive to sparsity than traditional classification algorithms. In addition, taking into account that accuracy of classification based on association methods has a straight correlation to the characteristics of data attributes, we may say that the CBA-Fuzzy algorithm provides some advances for a recommendation scenario. It includes the discretization process and the definition of the degrees of membership to the generated intervals and, hence, it brings more significance and value to data.

Since the method employs collaborative filtering and content-based approaches, it may be seen as a hybrid method and it can benefit from advantages of both categories of methods in order to minimize common drawbacks of recommender systems. Firstly, as it employs historical data from other users, the characterization of the groups and the classification model may be seen as a collaborative filtering approach. On the other hand, as our method considers active user's past behavior to determine which group he belongs to, it can be viewed as a content-based method too. By means of the analysis made by means of the simulations of real critical situations in Section 5, we were able to confirm that it was possible to minimize significantly some effects of recommender systems limitations as first-rater and gray sheep problems.

In spite of the efforts addressed to improve recommender systems and the advances reached in order to deal with their main drawbacks, there are still numerous challenges correlated to these systems. The number of users and items are constantly increasing in web systems due to their popularization and, therefore, future adaptation and extensions would be needed in the proposed technique. Moreover, certain application domains and/or particularities related to a specific system may also lead future developments, especially in the model based technique employed. There are numerous perspectives associated to this work that would allow

to reduce recommender systems' limitations even more. The method tested in this work could be extended to deal with frequent drawbacks associated to other Web mining areas as well as to new limitations that might emerge in future.

References

- Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In P. Buneman & S. Jajodia (Eds.), *ACMSIGMOD international conference on management of data* (pp. 207–216). DC, USA: Washington.
- Ahn, H. J. (2008). A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences*, 178(1), 37–51.
- Balabanovi, M., & Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3), 66–72.
- Benyon, D., & Murray, D. (1993). Developing adaptive systems to fit individual aptitudes. UI '93. In *Proceedings of the 1st international conference on intelligent user interfaces IUI* (pp. 115–121). Orlando, FL, USA.
- Berka, T., & Plnig, M. (2004). Designing recommender systems for tourism. In *The 11th international conference on information technology in travel and tourism*, Cairo, Egypt.
- Bilgus, D., & Pazzani, M. J. (1998). Learning collaborative information filters. In *15th international conference in machine learning* (pp. 46–54). Bari, Italy: Morgan Kaufmann.
- Blanco-Fernández, Y., Pazos-Arias, Gil-Solla, J. J., Ramos-Cabrera, A., López-Nores, M., García-Duque, J., Fernández-Vilas, A., et al. (2008). A flexible semantic inference methodology to reason about user preferences in knowledge-based recommender systems. *Knowledge-Based Systems*, 21, 305–320.
- Bobadilla, J., Ortega, F., Hernando, A., & Bernal, J. (2012). A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems*, 26(2012), 225–238.
- Bobadilla, J., Ortega, F., Hernando, A., & Alcalá, J. (2011). Improving collaborative filtering recommender system results and performance using genetic algorithms. *Knowledge-Based Systems*, 24(2011), 1310–1316.
- Breese, J., Heckerman, D., & Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of uncertainty in artificial intelligence UAI* (pp. 43–52). Helsinki, Finland.
- Campos, L. M., Fernández-Luna, J. M., & Huete, J. F. (2008). A collaborative recommender system based on probabilistic inference from fuzzy observations. *Fuzzy Sets and Systems*, 159(12), 1554–1576.
- Campos, L. M., Fernández-Luna, J. M., Huete, J. F., & Rueda-Morales, M. A. (2010). Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian Networks. *International Journal of Approximate Reasoning*, 51(2010), 785–799.
- Cao, Y., Li, Y., & Liao, X. (2005). Applying fuzzy logic to recommend consumer electronics. In *ICDCIT, Bhubaneswar, India, Lectures Notes in Computer Science* (Vol. 3816, pp. 278–289). Springer.
- Cheung, K.-W., Kwok, J. T., Law, M. H., & Tsui, K.-C. (2003). Mining customer product ratings for personalized marketing. *Decision Support Systems*, 35(2), 231–243.
- Chou, P. H., Li, P. H., Chen, K. K., & Wu, M. J. (2010). Integrating web mining and neural network for personalized e-commerce automatic service. *Expert Systems with Applications*, 37(4), 2898–2910.
- Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartini, M. (1999). Combining content-based and collaborative filters in an online newspaper. *ACM SIGIR workshop on recommender systems*. Berkeley, CA: ACM Press.
- Coelho, B., Martins, C., & Almeida, A. 2009. Adaptive tourism modeling and socialization system. In *2009 international conference on computational science and engineering* (Vol. 4, pp. 645–652). Vancouver, Canada.
- Condliff, M. K., Lewis, D. D., Madigan, D., Posse, C. (1999). Bayesian mixed-effects models for recommender systems. In *Proceedings of SIGIR-99 Workshop on recommender systems algorithms and evaluation*, Berkeley, CA. <<http://www.cs.umbc.edu/~ian/sigir99-rec>>.
- Diez, J., del Coz, J. J., Luaces, O., & Bahamonde, A. (2008). Clustering people according to their preference criteria. *Expert Systems with Applications*, 34(2008), 1274–1284.
- Dubois, D., Hullermeier, E., & Prade, H. (2006). Fuzzy methods for case-based recommendation and decision support. *Journal of Intelligent Information Systems*, 27(2), 95–115.
- Felfernig, A., Gordea, S., Jannach, D., Teppan, E., & Zanker, M. (2007). A short survey of recommendation technologies in travel and tourism. *OEGAI Journal*, 25(7), 17–22.
- Fink, J., & Kobsa, A. (2002). User modeling for personalized city tours. *Artificial Intelligence Review*, 18(1), 33–74.
- Forsati, R., & Meybodi, M. R. (2010). Effective page recommendation algorithms based on distributed learning automata and weighted association rules. *Expert Systems with Applications*, 37(2010), 1316–1330.
- Fu, X., Budzik, J., & Hammond, K. J. (2000). Mining navigation history for recommendation. In *IUI '00: Proceedings of the 5th international conference on intelligent user interfaces* (pp. 106–112). New York, NY, USA: ACM Press.
- García-Crespo, A., López-Cuadrado, J. L., González-Carrasco, I., Colomo-Palacios, R., & Ruiz-Mezcua, B. (2012). SINVLIO: Using semantics and fuzzy logic to provide individual investment portfolio recommendations. *Knowledge-Based Systems*, 27, 103–118.
- Gedikli, F., & Jannach, D. (2010). Neighborhood-restricted mining and weighted application of association rules for recommenders. In *8th Workshop on intelligent techniques for web personalization & recommender systems (ITWP'10)*. Big Island of Hawaii, June 20.
- Hinze, A., Voisard, A., & Buchanan, G. (2009). Tip: Personalizing information delivery in a tourist information system. *Journal of IT & Tourism*, 11(3), 247–264.
- Jannach, D., Zanker, M., Jessenitschnig, M., & Seidler, O. (2007). Developing a conversational travel advisor with advisor suite. In *Information and Communication Technologies in Tourism*, 43–52.
- Kim, H. N., Alkhalidi, A., El Saddik, A., & Jo, G. S. (2011). Collaborative user modeling with user-generated tags for social recommender systems. *Expert Systems with Applications*, 38(7), 8488–8496.
- Kim, K., & Ahn, H. (2008). A recommender system using GA K-means clustering in an online shopping market. *Expert Systems with Applications*, 34(2), 1200–1209.
- Kiran R. U. & Reddy, P. K. (2009). An improved multiple minimum support based approach to mine rare association rules. In *proceedings of the ieee symposium on computational intelligence and data mining, CIDM* (pp. 340–347). Nashville, TN, USA.
- Koren, Y. (2010). Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4), 89–97.
- Lang, K. (1995). Newsweeder: learning to Filter netnews. In *Proceedings of the 12th international conference on machine learning, morgan kaufmann publishers Inc.* (pp. 331–339). CA, USA: San Mateo.
- Lee, C.-H., Kim, Y.-H., & Rhee, P.-K. (2001). Web personalization expert with combining collaborative filtering and association rule mining technique. *Expert Systems and Applications*, 21(3), 131–137.
- Leung, C. W., Chan, S. C., & Chung, F. (2008). An empirical study of a cross-level association rule mining approach to cold-start recommendations. *Knowledge-Based Systems*, 21(7), 515–529.
- Li, W., Han, J., & Pei, J. (2001). CMAR: Accurate and efficient classification based on multiple class-association rules. In *Proceeding of IEEE international conference on data mining, ICDM*, (pp. 369–376). California, USA: San Jose.
- Lin, W., Alvarez, S. A., & Ruiz, C. (2002). Efficient adaptive-support association rule mining for recommender systems. *Data Mining and Knowledge Discovery*, 6(1), 83–105.
- Liu, B., Hsu, W., & Ma, Y. (1998). Integrating classification and association rule mining. In *Knowledge Discovery and Data Mining*, 80–86.
- Liu, B., Hsu, W., & Ma, Y. 1999. Pruning and summarizing the discovered associations. In *KDD 99: Proceedings of the 5th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 125–134). New York, NY, USA: ACM.
- Loh, S., Lorenzi, F., Saldaa, R., & Lichnow, D. (2004). A tourism recommender system based on collaboration and text analysis. *Information Technology and Tourism*, 6(3), 157–165.
- Lucas, J. P., Segreña, S., & Moreno, M. N. (2008). Comparing the use of traditional and associative classifiers towards personalized recommendations. In *20th International conference on software engineering and knowledge, engineering* pp. 607–612.
- Lucas, J. P., Laurent, A., Moreno, M. N., & Teisseire, M. (2012b). A fuzzy associative classification approach for recommender systems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 20(4), 579–617.
- Lucas, J. P., Segreña, S., & Moreno, M. N. (2012a). Making use of associative classifiers in order to alleviate typical drawbacks in recommender systems. *Expert Systems with Applications*, 39(1), 1273–1283.
- Moreno, M. N., Lucas, J. P., López, V., & Muñoz, M. (2010). Semantic based web mining for recommender systems. *Advances in Intelligent and Soft Computing*, 79, 17–25.
- Nasraoui, O., Frigui, H., Joshi, A., & Krishnapuram, R. (1999). Mining web access logs using relational competitive fuzzy clustering. In *Proceedings of the eight international fuzzy systems association world congress*. Taipei, Taiwan.
- Quinlan, R. (1993). C4.5: Programs for machine learning (morgan kaufmann series in machine learning). Morgan Kaufmann.
- Ricci, F. 2002. Travel recommender systems. Tech. rep., eCommerce and Tourism Research Laboratory, ITC-Irst, Trento, Italy.
- Ricci, F., & Werthner, H. (2002). Case-based querying for travel planning recommendation. *Information Technology and Tourism*, 4, 215–226.
- Rittman, M. (2005). What is sparsity, and why should I be concerned with it?. Oracle news, June 27, <http://www.dba-oracle.com/oracle_news/2005_6_27_What_Is_Sparsity.htm>.
- Roh, T. H., Oh, K. J., & Han, I. (2003). The collaborative filtering recommendation based on SOM cluster-indexing CBR. *Expert Systems with Applications*, 25(3), 413–423.
- Sarwar, B. M., Karypis, G., Konstan, J. A., Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. In *2nd ACM conference on electronic commerce* (pp. 158–167). MN, USA: Minneapolis.
- Sasikala, P., & Vidhya, V. (2012). An efficient concept-based mining model for deriving user profiles. *International Journal of Applied Information Systems*, 1(6), 26–34.
- Schafer, J. B. (2005). The application of data-mining to recommender systems. In J. Wang (Ed.), *Encyclopedia of data warehousing and mining*. Information Science Publishing.
- Stock, O., & Zancanaro, M. (2002). Intelligent interactive information presentation for cultural tourism. In *International CLASS workshop on natural intelligent and effective interaction in multimodal dialogue systems*. Denmark: Copenhagen.

- Sun, X., Kong, F., & Chen, H. (2005). Using quantitative association rules in collaborative filtering. In Fan, W., Wu, Z., & Yang, J. (Eds.), *WAIM*, Vol. 3739 of *Lecture Notes in Computer Science*, 822–827, Springer.
- Thabtah, F., Cowling, P., & Peng, Y. (2004). MMAC: A new multi-class, multi-label associative classification approach. In *Fourth IEEE international conference on data mining* (pp. 217–224) Brighton, UK.
- Toivonen, H., Klemettinen, M., Ronkainen, P., Hatonen, K., & Mannila, H. (1995). Pruning and grouping discovered association rules. In *MLnet workshop on statistics, machine learning, and discovery in databases* (pp. 47–52). Heraklion, Crete, Greece.
- Wang, A., Yuan, W., Liu, J., Yu, Z., & Li, H. (2009). A novel pattern recognition algorithm: Combining ART network with SVM to reconstruct a multi-class classifier. *Computers and Mathematics with Applications*, 57, 1908–1914.
- Werthner, H., & Ricci, F. (2004). E-commerce and tourism. *Communications of the ACM*, 47(12), 101–105.
- Xu, J. A., & Araki, K. (2006). ASVM-based personal recommendation system for TV programs, in *Proc. Int. Conf. on multi-media modeling conference* (pp. 401–404) Beijing, China.
- Yager, R. R. (2003). Fuzzy logic methods in recommender systems. *Fuzzy Sets and Systems*, 136(2), 133–149.
- Yin, X., & Han, J. (2003). Cpar: Classification based on predictive association rules. In *SIAM international conference on data mining (SDM03)* (pp. 331–335). CA, USA: San Francisco.
- Zanker, M., Fuchs, M., Hopken, W., Tuta, M., & Muller, N. 2008. Evaluating recommender systems in tourism – A case study from austria. In *ENTER*, 24–34.
- Zhang, F., Chang, H. Y. (2005). On a hybrid rule based recommender system. In *The fifth international conference on computer and information technology* (pp. 194–198). Washington, DC, USA. IEEE Computer Society.
- Zhen, L., Huang, G. Q., & Jiang, Z. (2009). Recommender system based on workflow. *Decision Support Systems*, 48(1), 237–245.
- Zhen, L., Jiang, Z., & Song, H. (2010). Distributed recommender for peer-to-peer knowledge sharing. *Information Sciences*, 180(18), 3546–3561.