

Towards Collaborative Travel Recommender Systems

Cane Wingki Leung, Stephen Chifai Chan, Korris Fulai Chung

Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
{cswkleung, csschan, cskchung}@comp.polyu.edu.hk

ABSTRACT

Collaborative filtering (CF) based recommender systems have been proven to be a promising solution to the problem of information overload. Such systems provide personalized recommendations to users based on their previously expressed preferences and that of other similar users. In the past decade, they have been successfully applied in various domains, such as the recommendation of books and movies, where items are simple, independent and single units. When applied in the tourism domain, however, CF falls short due to the simplicity of existing techniques and complexity of tourism products. In view of this, a study was carried out to review the research problems and opportunities. This paper details the results of the study, which includes a review on the recent developments in CF as well as recommender systems in tourism, and suggests future research directions for personalized recommendation of tourist destinations and products.

Keywords: Recommender System, Collaborative Filtering, Personalization, Data Mining, Travel & Tourism

1. INTRODUCTION

In the past decades, the rapid development of Internet technologies has led to severe information overload. The amount of information available on the web, and the amount added daily, is too much for us to process. The use of recommender systems has therefore become increasingly popular. Such systems provide advices to users about items they might like to purchase or examine, avoid them from being drown in the vast amount of information [3].

There are three major types of recommender systems: content-based, knowledge-based and social- or collaborative-filtering (CF) based. Content-based recommender systems establish users' interest profiles by analyzing the features of their preferred items. In such systems, features of items are compared to those of the preferred items of the user. The more relevant items, measured by features similarity, are recommended. Knowledge-based recommender systems make use of knowledge about users and products to generate recommendations. They use a reasoning process to determine what products meet a user's requirements. Social-filtering or CF based recommender systems provide personalized recommendations according to user preferences. Such systems maintain data about users' purchasing habits or interests and from there identify groups of similar users. For a target user, known as the active user, items liked by other similar users are recommended.

In addition to providing personalized recommendations, CF offers a number of advantages over the other two techniques, making it one of the most promising solutions to information overload. As no content information is considered in the recommendation process, items being filtered need not be amenable to parsing by a computer. Besides, recommendations generated are based on tastes of users rather than more objective properties of the items themselves. The items

recommended to a user can therefore be very different (content-wise) from what the user has liked previously, overcoming a major limitation of content-based recommender systems [31]. When compared to knowledge-based recommender systems, CF techniques are much simpler and easier to implement. While the former requires a domain knowledge engineering process to build a knowledge base of items, the later can be fully automated. On the one hand, this allows CF to be easily applicable to any domain where a database of user preferences is available. On the other hand, this makes CF falls short when recommending complex items such as travel purchases [4]. Nevertheless, CF has become the most successful technology applied in recommender systems. To date, it has already been studied and applied to various domains such as the recommendation of books, Usenet articles and jokes.

In recent years, CF has gained research attention in the tourism domain for several reasons. By observing and predicting user preferences, CF prevents users from being overwhelmed by vast amount of travel information. This is particularly valuable in an information-intensive domain like tourism, which comprises a large variety of products such as accommodations, attractions, restaurants, and many others. Besides, one characteristic of the purchase of tourism products is that at the moment of decision-making, only information about the product, not the product itself, is available to users [26][33]. Current web-based tourist information portals can present archives of pre-defined destinations, some with rich multimedia content. While tourism product suppliers are always in favor of their own businesses, other visitors' experiences would probably provide more valuable and unbiased information about a destination. CF techniques would help users obtain such information according to their travel preferences [4].

Yet, CF is not readily applicable to the recommendation of tourism products [4][28], which are more diversified

and complex than products like books. With the objective to bridge the gap between CF and tourism products recommendation, a study was carried out to identify the problems in recommending tourism products, and to review the state-of-the-art in both areas. This paper details the results of the study, and suggests future research directions to work towards collaborative travel recommender systems.

1.1 Collaborative Filtering (CF)

The underlying philosophy of CF is that each individual belongs to a larger group of like-minded individuals. CD-based systems therefore make personalized recommendations to users based on the interests of similar users [7][20][27][31]. In a movie recommender system, for example, the movie Shrek may be suggested to a user Sam, after noticing that his taste is similar to that of another user Bob, who has liked Shrek.

Users' preferences provide the basis for making collaborative recommendations. Such preferences can be expressed explicitly by means of ratings, or implicitly such as in purchase history. CF algorithms may compute a predicted score (rating) of a particular item or a list of recommended items, known as Top-N recommendation, for the active user.

1.2 Recommending Tourism Products

Recommending tourism products poses additional challenges to existing CF researches. They are mainly caused by the complexity of tourism products, and the special taxonomies they can have.

Firstly, tourism products are highly heterogeneous even at a single destination. They include restaurants, hotels, theme parks, and many others. Recommending these products with pure CF techniques can be easy, as long as each item is uniquely identified (by a product code, for example) and no content information is considered. However, these make less sense for tourism products as content information may also be essential in the filtering process. For example, if a user is looking for a place to ski, a travel recommender system must be able to filter out irrelevant activities such as hiking. Some existing techniques such as [14] and [30] already allow content information to be used in the recommendation process. This, however, can be more difficult in recommending tourism products due to **product heterogeneity** since products can have different representations, and the filtering of content information may be based on different features for different products.

Furthermore, tourism products may also contain essential context information such as availability and location. For example, if a user is looking for a place to ski in June, a travel advisory system may consider Australia as a suitable destination. If his/she wishes to ski in December, however, it will be more suitable to recommend some places in Europe due to the availability of the desired activity at different locations

and points in time. **Context awareness** is therefore another important factor in designing a travel recommender system.

Lastly, tourism products have strong inter-item relationships. They may also be arranged into different hierarchies or taxonomies. For example, the is-a hierarchy may exist if products are categorized according to their type (e.g. Surfer's Paradise is a beach). Besides, the part-of or component hierarchy may also exist if they are categorized according to their location (e.g. Surfer's paradise is part of Gold Coast). Another special example of component hierarchy in tourism products is tour package, which comprises multiple heterogeneous products such as flight tickets, accommodation, as well as tourist spots (e.g. a visit to Surfer's Paradise is included in a 2-day tour to Gold Coast). These products may exist in multiple packages, or may be purchased alone as a single unit of product. To recommend tourism products, different **product taxonomies** should be taken into consideration.

1.3 Organization

The rest of this paper is organized as follows. In Section 2, different types of CF techniques are described and their strengths and weaknesses are commented on. In Section 3, several CF applications are introduced. Related work in tourism is also described. In Section 4, some CF techniques and related work introduced are further discussed with respect to their ability to recommend tourism products. From there some future research directions are identified. Lastly, Section 5 concludes this paper.

2. CF TECHNIQUES

CF algorithms were traditionally classified into two major types, namely memory-based and model-based [2]. Memory-based algorithms operate over the entire user database, and recommend products using statistical methods. Model-based algorithms construct compact models from the user database, and recommend products using probabilistic methods. Since such models can be constructed offline, model-based algorithms usually improve online performance.

Such classification mainly reflects how products are recommended in real-time – either based on the entire database or models constructed and stored in memory. To more specifically describe various CF algorithms, in this paper they are classified by the underlying techniques they used to make recommendations.

2.1 K-Nearest Neighbor (k-nn)

K-nearest neighbor (k-nn), also known as pure CF, was commonly used in early CF-based systems. It works in three major steps, namely similarity weighting of users, neighbor selection and prediction computation.

At the similarity weighting stage, *each user* in the

database is assigned a similarity weight using some measures (such as those analyzed in [2] and [27]). Such similarity is reflected in the ratings they have given on items. For two users to be comparable, only items that both users have rated are counted. At the neighbor selection stage, a number of k nearest neighbors, who are the users having the highest similarity weights, of the active user are selected as predictors for items. Based on the interests of the selected neighbors and some partial information of the active user, prediction score of an item is computed.

K-nn makes predictions that are ratings-based and content-independent. It can therefore be applied to domains where textual descriptions of products are not available, not meaningful, or cannot be easily categorized by any attribute. Furthermore, it uses statistical approaches that are easy to implement, and can be applied as long as a database of users preferences on items is available. Despite its simplicity, empirical analyses [2][12] proved that it produces good predictions, especially if the active user has rated a significant number of items. Besides, predictions are generated by performing real-time computations over the database. As no pre-compilation of data is required, new data points can be easily added to the application.

For the above reasons, k-nn was commonly used in recommender systems. This, however, has revealed a number of challenges for researchers. Among which the most essential ones are data sparseness, cold start problem and scalability.

Data sparseness arises as users can only rate the items, or part of the items, that they have access to (that is, they cannot rate any unobserved items). In k-nn, it is possible that the set of common items rated by the active user and his neighbors is very small, even if the selected neighbors are already the "nearest" ones. This resulted in poor prediction quality [29].

Furthermore, data sparseness implies that given a very large database of items, there may not be a complete set of ratings across all items. The cold-start problem, or the first-rater problem, arises when no prediction about a certain item can be made for the active user as no other users have rated it before [16].

Scalability is an important issue especially in real-time applications. K-nn has severe performance bottleneck in the similarity weighting stage, as it requires real-time computations to be performed over the entire database for each prediction. Performance of the algorithms degrades when the numbers of users and items grow in the system.

2.2 Classification

Some researches regarded CF as a classification problem [1][21]. Instead of predicting the rating of an item, classification techniques predict whether the active user would like or dislike it. To do so, user ratings are

first discretized, usually into two classes - *Like* and *Dislike*. Classification techniques, such as Bayesian network [2], are then used to make predictions for the active user.

A Bayesian network makes use of decision trees to predict the likeliness of items, conditioned on whether some other items have been liked previously. Since decision tree induction is beyond the scope of this paper, it is not described here. Please refer to [11] for details.

In this technique, decision trees induction is done offline. Only the decision tree built for the target item has to be inspected in real-time, resulted in improved online performance over k-nn. Prediction quality, however, degraded due to its probabilistic nature.

2.3 Clustering

2.3.1 User-based Clustering

The Bayesian cluster model divides the user base into many groups (clusters) according to their similarity [2]. The recommendation process first assigns the active user to the cluster containing the most similar users. Then, it uses the preferences of users in that cluster to generate recommendations. Since the active user is only compared to a controlled number of clusters, scalability is improved. Similar to the classification technique, it produces less personal recommendations.

Eigentaste [8], another user-based clustering technique, clusters users by their similarities reflected in the ratings they have given on a gauge set of items. Then, the mean for each non-gauge item in each cluster is computed based on the number of users who have rated that item. Lookup tables of recommendations for each cluster are built by sorting the non-gauge items in descending order of their mean ratings. In real-time, recommendations are obtained by querying the pre-computed lookup tables. This simple online process ensures scalability. Since users are required to rate a gauge set of items, the ratings matrixes representing their preferences are dense. Users can therefore be well correlated to preserve prediction accuracy.

2.3.2 Item-based Clustering

Sarwar et al. [29] explored the item-based clustering approach, in which recommendations are produced by finding items that are similar to those the active user has liked previously. As the number of items and their relationships are relatively static, as compared to the number of users and ratings, recommendations can be computed offline. This avoids the performance bottleneck of k-nn and improves scalability. However, prediction quality of this approach only shows slight improvement over k-nn. This might be due to the fact that data sparseness was not addressed in the research.

2.4 Association Rule Mining

Association rule mining (ARM) is a data mining technique for discovering interesting relationships between items by finding items frequently appeared together in transactional databases. An association rule is denoted as $A \Rightarrow B$, and measured by two measures known as support and confidence. For example, the rule $A \Rightarrow B$ [20%, 90%] means that 90% of users who purchased item A also purchased item B, and 20% of all users purchased both of them [11].

ARM is usually used for market basket analysis. It can also be used to generate recommendations after interesting rules are mined. For example if $A \Rightarrow B$ is interesting in a certain system, and the active user has liked item A previously, he/she will probably like item B also. Note that a rule is considered as interesting only if its support and confidence values are higher than the user-specified minimum.

2.4.1 Adaptive Association Rule Mining

Lin et al. [18], however, pointed out that traditional ARM algorithms are inefficient for collaborative recommendation for two reasons. Firstly, such algorithms mine rules for all items in the database. Therefore, many rules mined will not be relevant for a given user. Secondly, the minimum support and confidence have to be specified in advance, which might lead to either too many or too few rules due to variations in user tastes and items' popularities.

To address these issues, they suggested an adaptive-support rule mining algorithm, known as AR-CRS, tailored for CF. AR-CRS mines rules for one target item at a time, and automatically adjusts the minimum support value to mine a user-specified number of rules. It can compute associations among users, items, or both.

2.4.2 Multi-level Association Rule Mining

Kim & Kim [15] applied multi-level association rules mining (MAR) to address data sparseness and the cold start problem. MAR can be applied on items that are organized in a hierarchical category structure, such as the classification of goods in a department store (is-a hierarchy). Its key feature is to mine rules at different level of abstraction [10]. A major concern of MAR is that higher support is likely to exist at higher level of abstraction. Different minimum supports should therefore be used at different levels.

When applied in CF, MAR can compute preferences for items that are not covered by the traditional, single-level technique. To do so, the category C_i to which a preferred item i belongs is found. If any association rule among C_i and another category C_j exists, some preferences will be given to items that belong to C_j . The consideration of such product taxonomy allows recommending cold-start or less popular products.

2.5 Hybrid Techniques

Personality Diagnosis (PD) [25] combined k-nn and clustering to improve prediction quality. In addition to the traditional similarity weighting measures, it employs a probabilistic interpretation of results to provide better predictions. PD retains the advantages of k-nn such as simplicity and adding new data points with ease. In terms of accuracy, it outperforms several approaches including k-nn and the Bayesian network model although it does not address data sparseness. In terms of time and space performance, it does not provide any advantage over k-nn.

O'Connor & Herlocker [23] combined k-nn and clustering to improve scalability. Their technique uses existing data partitioning and clustering techniques to cluster items based on user ratings. Predictions are then computed independently in each cluster. Experimental results show that their technique is more scalable than k-nn, as its real-time prediction process only considers items in a single cluster. Consistent improvement, however, in its prediction accuracy could not be observed. A possible reason for this is that their algorithm restricts items to being exclusively in one single cluster. Items that might have significant predictive value for multiple clusters therefore could not reflect real user preferences.

Sarwar et al. [30] combined information filtering (IF) and CF to provide better recommendations. They integrated filtering agents, known as filterbots, that evaluate and rate items using syntactic features to provide a dense ratings set. This improves prediction accuracy, and its effectiveness was proved in [9].

The Item-based Clustering Hybrid Method (ICHM) also integrated content information into CF to improve prediction quality and solve the cold start problem [17]. In ICHM, items are clustered based on content information. In a movie recommender, for example, movies are clustered by their genre, director, actor, etc. For a new item with no ratings, its prediction score is computed as a weighted sum of the ratings given to other items in the same cluster. Experimental results show that this approach improves prediction accuracy. It is also able to recommend cold-start items with reduced accuracy, as compared to using a dataset with no cold-start items.

3. CF APPLICATIONS AND RELATED WORK

CF has been proven successful in both research and practice. A few examples selected from different domains are introduced in this section. Amazon.com [19] is one of the most successful and well-known online retailers. It makes use of purchase histories of customers to produce recommendations using the item-based technique. GroupLens [27][16] is a pioneering and well-known project in automated CF. It had used k-nn to recommend Usenet articles. Filtering agents were later integrated into the system to improve prediction quality [30]. MovieLens is another recommender system developed by the GroupLens team. It recommends

movies to users using GroupLens' technologies. Jester [8] is an online joke recommender. When it was first developed, it had used k-nn to generate recommendations. When the number of registered users increased, however, it was found that the processing time of the system increased linearly until it crashed. The constant time algorithm Eigentaste was therefore developed and adopted.

Some related applications in the tourism domain were also studied. They are mainly based on knowledge-based technologies. CF is only used to rank the recommended products according to user preferences.

Dietorecs [4] and NutKing [22] are two similar applications developed by the eCommerce and Tourism Research Laboratory (eCTRL) [6]. They recommend travel products and activities based on the Case-Based Reasoning (CBR) technology, which is a problem solving technique that solves the current problem by adapting solutions for previously solved, similar problems. The CBR cycle consists of 4 stages, namely Retrieve, Reuse, Revise and Retain. Detailed descriptions of each stage can be found in [34].

A case base refers to a repository of representative cases, which are previously adopted solutions and/or possible solutions to new problems. In Dietorecs and NutKing, it is actually a pool of travel products and activities recommended by domain experts with respect to different travel settings (e.g. duration of travel). To start generating recommendations, a new case containing the active user's profile and travel settings is created. Pre-defined recommendations are then retrieved from the case base (the Retrieve stage), and ranked according to the preferences of the active user. The retrieved cases can be accepted as-is, or modified by the active user by adding or removing elements in the recommended cases (the Reuse stage).

Ski Europe [32] recommends places for skiing in Europe based on the TripMatcher technology [4]. Similar to Dietorecs and NutKing, the system maintains a knowledge base of destinations, developed by professional researchers worldwide. For each destination, travel experts have ranked different activities during different times of the year. TripMatcher applies information filtering and text mining algorithms to such context information, products' content information and users' ratings to generate recommendations.

In CBR, management of the case base is an important issue as it directly affects the correctness and quality of the solutions produced by a system. The major concern here is the selection of representative cases, including the number of cases and their content. In the CBR cycle, the Revise and Retain stages allow domain experts to, based on the solutions modified by users in the Reuse stage, revise the cases originally stored in the case base, retain the cases by updating the original ones, or store them as new cases. Thus they are very important steps

to help maintain the quality of the case base. However, how these important steps have been addressed in Dietorecs and NutKing, is unclear. Although the TripMatcher technology does not rely on CBR, it faces a similar problem as it produce recommendations from a knowledge based of pre-defined trips.

Due to the complexity of tourism products, it is important to develop a data model for their representation. As mentioned before, tourism products are heterogeneous, complex, inter-related and may contain essential context and content information. However, existing CF algorithms are all based on very simple data models. In fact, only ratings/purchase data and the unique identifiers of products are used in most algorithms. Integrating data models in CF would facilitate the development of more sophisticated, flexible and interoperable recommendation engines.

The Open Tourism Consortium (OTC) [24] is a consortium of companies, government agencies, individuals, and universities participating in the development of publicly available standards and open source software to support tourism. Recently, they developed an XML-based data exchange language known as TourML to represent objects and events, such as hotels and restaurants, of interest to tourists. Although OTC is not doing any CF-related research, their work can provide reference for modeling different tourism products.

4. FUTURE RESEARCH DIRECTIONS

As aforementioned, the widespread use of pure CF methods has revealed some fundamental challenges concerning the scalability and prediction quality of CF systems. Since then, CF researches have been focusing on developing more scalable and accurate algorithms. Although numerous approaches have been developed, the dominant paradigm for performing CF has been based on techniques regarding items in the domain as some simple, single and independent units. This paradigm, although has been proven successful in various domains, falls short when applied in the tourism domain which poses additional challenges namely product heterogeneity, context awareness and the existence of product taxonomies to CF.

Another problem in this paradigm is non-transitive association [14]. In user-based methods, if two users have both experienced or rated similar items but not the same ones, their correlation is lost. This is known as the user-based non-transitive association. In contrast, using item-similarity instead of user similarity avoids this problem, but results in the item-based non-transitive association problem if two similar items have never been experienced or rated by a user.

Due to the presence of content and context information, purely CF algorithms may not be sufficient for travel recommender systems. Existing solutions, such as TripMatcher [4], therefore combined different

techniques with CF to form a recommendation engine. Such technology has already addressed product heterogeneity and context awareness using information retrieval and text mining algorithms. However, it does not take into consideration product taxonomies in the recommendation process.

In fact in most CF algorithms, recommendations are generated based on the relationship between some attributes of items and users' profiles. The item-based technique [29] behaves differently in the technical aspect, but still it is based on the relationship between items and users identified earlier to generate recommendations. They do not take into consideration inter-item relationships and product taxonomies, which are likely to be found in tourism products. Obviously, treating items as single units in a recommender system is inappropriate in the tourism domain.

MAR has already shown that it is possible to integrate the is-a hierarchy to improve recommendations [15]. Such hierarchy, and other possible hierarchies, have not yet been considered in travel recommender systems. This reveals a potential research problem, which is to extend the existing techniques to incorporate different product taxonomies in the recommendation process. This may lessen the problems of non-transitive association in CF, because item similarities are already implied in the hierarchy itself.

Yet, MAR applied to the is-a hierarchy may not be readily extendable to other hierarchies such as part-of. The is-a hierarchy exercises the upward closure property which affects the generation and pruning of associations rules. In such hierarchy, the support count of a higher-level item must be equal to the sum of that of its child items. In other words, if a higher-level item is infrequent (with respect to the user-specified minimum support), all of its child items must also be infrequent. In contrast, if a lower-level item is frequent, all of its parent items must also be frequent. Due to the presence of this property, the algorithm in [15] recommends products of a certain category, if some associations between that category and the category of a previously liked item exist. Considering other taxonomies in the recommendation process, however, have not yet been studied in the literature of CF. Further research on this is required.

To apply MAR in travel recommender systems, its challenges and shortcomings should also be addressed. The major concern in MAR has been the determination of minimum supports at different level of hierarchies (is-a hierarchy). Whether this applies to other product hierarchies is unknown, but should be aware of in future researches. The adaptive minimum support method proposed in [18] may be considered, as it can adjust the minimum support value according to the desired number of resulting rules.

A shortcoming of applying ARM/MAR in CF was observed. Traditional ARM/MAR generate "interesting

rules" based on items' frequencies of appearance in transactions in a database. In other words, the appearance of an item in a user's transaction implies that it is liked by that user. In classical ratings-based CF, however, ratings can be positive or negative. Using ARM/MAR requires transforming ratings data into class labels such as like and dislike. In [15], for example, a certain rating given by a user is converted to 1 if it is greater than the average rating given by that user, or 0 otherwise. Applying this technique to the Jester dataset [13], ratings should first be transformed from a continuous rating scale (-10 to 10) to like (1) and dislike (0). Consider a case where all ratings given by a user are negative with an average -5. According to [15], a joke with rating -1 will also be considered as being liked by the user. This actually implies a loss of information in the user's likeliness of items. Fuzzy logic may be a possible solution to this problem.

5. CONCLUSION

CF has been a promising solution to the problem of information overload in the past decade. In recent years, it began to gain research attention in the tourism domain for its ability to produce personalized recommendations. However, existing CF techniques fall short when recommending tourism products which are heterogeneous and complex. With the objective to bridge the gap between CF and tourism products recommendation, a study was carried out to identify the problems in recommending tourism products, and to review the state-of-the-art in both areas. This paper details the results of the study, and suggests future research directions to work towards collaborative travel recommender systems.

REFERENCES

- [1] Basu, C., Hirsh, H., Cohen, W., "Recommendation as classification: using social and content-based information in recommendation". *Proceeding of the 15th National Conference on Artificial Intelligence*, pp. 714-720, 1998
- [2] Breese, J. S., Heckerman, D., Kadie, C., "Empirical analysis of predictive algorithms for collaborative filtering". *Proceeding of the 14th Conference on Uncertainty in Artificial Intelligence*, pp. 43-52, 1998
- [3] Burke, R., "Knowledge-based recommender systems". In A. Kent (ed.), *Encyclopedia of Library and Information Systems*. Vol. 69, Supplement 32. New York: Marcel Dekker, 2000
- [4] Delgado, J., Davidson, R., "Knowledge bases and user profiling in travel and hospitality recommender systems". *Proceeding of ENTER 2002*, pp. 1-16, 2002
- [5] Fesenmaier, D. R., Ricci, F., Schaumlechner, E., Wöber, K., Zanellai, C., "Dietores: travel advisory for multiple decision styles". *Proceeding of ENTER 2003*, pp. 232-241, 2003
- [6] eCommerce and Tourism Research Laboratory: <http://ectrl.itc.it>
- [7] Goldberg, D., Nichols, D., Oki, B., Terry, D., "Using collaborative filtering to weave an information

- tapestry". *Communications of the ACM*, 35(12): 61-70, 1992
- [8] Goldberg, K., Roeder, T., Gupta D., Perkins, C., "Eigentaste: a constant time collaborative filtering algorithm". *Information Retrieval*, 4(2): 133-151, 2001
- [9] Good, N., Schafer, J. B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., Riedl, J., "Combining collaborative filtering with personal agents for better recommendations". *Proceeding of the 16th National Conference on Artificial Intelligence*, pp. 439-446, 1999
- [10] Han, J., Fu, Y., "Mining multiple-level association rules in large databases". *Knowledge and Data Engineering*, 11(5): 798-804, 1999
- [11] Han, J., Kamber, M., *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, 2000
- [12] Herlocker, J., Konstan, J., Riedl, J., "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms". *Information Retrieval*, 5, 287-310, 2002
- [13] Jester Data Download Page: <http://www.ieor.berkeley.edu/~goldberg/jester-data/>
- [14] Kim, B. M., Li, Q., Kim, J. W., Kim, J., "A new collaborative recommender system addressing three problems". *Proceeding of the 8th Pacific Rim Intl. Conference on Artificial Intelligence*, 2004
- [15] Kim, C., Kim, J., "A recommendation algorithm using multi-level association rules". *Proceeding of the IEEE/WIC Intl. Conference on Web Intelligence*, pp. 524-527, 2003
- [16] Konstan J., Miller B., Maltz, D. Herlocker J., Gordon, L., Riedl, J., "Applying collaborative filtering to Usenet news". *Communications of the ACM*. 40(3): 77-87, 1997
- [17] Li, Q., Kim, B. M., "Clustering app. roach for hybrid recommender system". *Proceeding of the IEEE/WIC Intl. Conference on Web Intelligence (WI'03)*, pp. 33-38, 2003
- [18] Lin, W., Alvarez, S. A., Ruiz, C., "Efficient adaptive-supp. ort association rule mining for recommender systems". *Data Mining and Knowledge Discovery*, 6(1): 83-105, 2002
- [19] Linden, G., Smith, B., York, J., "Amazon.com recommendations: item-to-item collaborative filtering". *Industry Report, IEEE*, pp. 76-80, 2003
- [20] Maltz, D., Ehrlich, K., "Pointing the way: active collaborative filtering". *Proceeding of ACM CHI '95 Conference on Human Factors in Computing Systems*, ACM Press, pp. 202-209, 1995
- [21] Miyahara, K., Pazzani, M. J., "Collaborative filtering with the simple Bayesian classifier". *Proceeding of the 6th Pacific Rim Intl. Conference on Artificial Intelligence*, pp. 679-689, 2000
- [22] NutKing: <http://itr.itc.it:8080/dev/jsp/index.jsp>
- [23] O'Connor, M., Herlocker, J., "Clustering items for collaborative filtering". *Recommender Systems Workshop*, 1999
- [24] Open Tourism Consortium Knowledge Repository: <http://www.opentourism.org/wiki/wiki.phtml>
- [25] Pennock, D.M., Horvitz, E., "Collaborative filtering by personality diagnosis: a hybrid memory- and model-based app. roach". *Proceeding of the 16th Conference on Uncertainty in Artificial Intelligence*, pp. 473-480, 2000
- [26] Poon, A., *Tourism, Technology and Competitive Strategies*. CAB Intl., Wallingford, U.K., 1993
- [27] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J., "GroupLens: an open architecture for collaborative filtering of netnews". *Proceeding of CSCW '94*, pp. 175-186., 1994
- [28] Ricci, F., "Travel recommender systems". *IEEE Intelligent Systems*, 17 (6): 55-57, 2002
- [29] Sarwar, B.M., Karypis, G., Konstan, J., Riedl, J., "Item-based collaborative filtering recommendation algorithms". *Proceeding of WWW10*, pp. 285-295, 2001
- [30] Sarwar, B.M., Konstan, J. Borchers, A., Herlocker, J., Miller, B., Riedl, J., "Using filtering agents to improve prediction quality in the GroupLens Research collaborative filtering system". *Proceeding of CSCW '98*, pp. 345-354, 1998
- [31] Shardanand, U., Maes, P., "Social information filtering: algorithms for automating 'word of mouth'". *Proceeding of CHI'95*, pp. 210-217, 1995
- [32] Ski Europe: <http://www.ski-europe.com/>
- [33] Stabb, S., Werther, H., "Intelligent systems for tourism". *IEEE Intelligent Systems*, 17(6): 53-55, 2002
- [34] Watson, I., *App. lying Case-Based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann Publishers, Inc., 1997