A HYBRID COLLABORATIVE RECOMMENDER SYSTEM BASED ON USER PROFILES

Marco Degemmis, Pasquale Lops, Giovanni Semeraro, M. Francesca Costabile, Oriana Licchelli, Stefano P. Guida
Dipartimento di Informatica, University of Bari, Via E. Orabona, 4, 70125 BARI, Italy

Keywords: Hybrid recommender systems, Information Filtering, Machine learning, User profiling

Abstract: Nowadays, users are overwhelmed by the abundant amount of information delivered through the Internet. Especially in the e-commerce area, largest catalogues offer millions of products and are visited by users having a variety of interests. It is of particular interest to provide customers with personal advice: Web personalization has become an indispensable part of e-commerce. One type of personalization that many Web sites have started to embody is represented by *recommender systems*, which provide customers with personalized advices about products or services. Collaborative systems actually represent the state-of-the-art of recommendation engines used in most e-commerce sites. In this paper, we propose a hybrid method that aims at improving collaborative techniques by means of user profiles that store knowledge about user interests.

1 INTRODUCTION

Most of the largest e-commerce Web sites is using recommender systems to help their customers find products to purchase. A recommender system learns from customers and recommends products that they will find most valuable among the available products. Recommender systems have been revolutionizing the way shoppers and information seekers find what they want, because they effectively help users in selecting items that best meet their needs and tastes.

Such systems take input directly or indirectly from users and, based on user needs, preferences and usage patterns, they make personalized recommendations of products or services. Recommender systems are used to either predict whether a particular user will like a particular item (*prediction problem*), or to identify a set of N items that will be of interest to a certain user (top-N *recommendation problem*) (Sarwar, et al., 2002).

The literature on recommender systems distinguishes primarily between the collaborative and the content-based approaches. In the first approach, the content (e.g. text) plays an important role: the system suggests the items similar to those the user liked in the past, based on the content comparison. In contrast with the content-based approach, a collaborative approach assumes that there is a set of users using the system: user advice is based on the item ratings provided by other users.

Hybrid recommender systems combining both techniques have also been proposed to gain better performance with fewer of the drawbacks of any individual technique (Burke, 2002; Balabanovic and Shoham, 1997; Konstan, et al., 1998; Pazzani, 1999). Examples of this kind of hybrid systems are *Fab* (Balabanovic and Shoham, 1997) and *Ringo* (Shardanand and Maes, 1995). *Fab* maintains user profiles based on content analysis, and directly compares the profiles to determine similar users for collaborative recommendations. Items are recommended to a user both when they score highly against that user profile or when they are highly rated by a user with a similar profile.

Ringo is similar to *Fab* except that, during a similarity assessment among users, the system selects profiles of users with the highest correlation with an individual user. *Ringo* compares user profiles to determine which users have similar tastes. Once similar users have been identified, according to a classical collaborative approach, the system predicts how much the user may like an item that has not yet been rated by computing a weighted average of all the rates given to that item by the other users that have similar tastes.

In (Tuzhilin and Adomavicius, 1999), it is remarked that: “In order to provide more accurate

Degemmis M., Lops P., Semeraro G., Francesca Costabile M., Licchelli O. and P. Guida S. (2004).
A HYBRID COLLABORATIVE RECOMMENDER SYSTEM BASED ON USER PROFILES.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 162-169
DOI: 10.5220/0002638201620169

recommendations, it is necessary to base them on a thorough analysis of the *on-line behavior of the user* that is much broader than the behavior captured by current content-based filtering systems". Rules describing the on-line behavior of a user can be learned from the analysis of his/her transactional history using various data mining methods and can be included as a part of that user's profile. *Behavioral profiles* can describe much richer types of user behavior than user profiles from the content-based approach, but they do not provide any recommendations by themselves. Therefore, it is important to couple the behavioral profiling approach with other techniques.

We consider the integration of behavioral profiles and collaborative methods into one integral approach. This is in line with basic principles of marketing, according to which customer recommendations should be based on understanding behavior of that customer and on the preferences of similar customers. In our approach, rules describing the customer behavior are used in order to discover preferences of users, such as product categories. For example, in a book recommending context, rules could be used in order to determine whether a user is interested or not in a specific book category. A simple example of such rules is: "Customers that buy at least 3 books belonging to the *horror* category are interested in that book category".

Preferences are stored in personal profiles exploited to group customers having the same interests. Our idea is that profiles could drive the collaborative method by reducing the set of users, on which the algorithm is applied, only to users interested in the same product categories. Profiles are inferred from the analysis of transactional data (browsing and purchasing history of users), without considering any content, and are exploited to discover for each user a set of "nearest neighbors" to compute collaborative recommendations. An intensive experimental session has been carried out to compare a pure collaborative approach to recommendation with respect to the one combined with user profiles of users.

The paper is organized as follows: Section 2 provides a description of the most frequently approaches used in recommender systems, i.e. collaborative and content-based ones. It also describes a possible way to combine the approaches to improve the entire recommendation process. Section 3 gives a description of the two systems, namely User Profile Engine (UPE) and Profile Extractor (PE), we integrated to build a hybrid recommender called U(PE)^2. Section 4 presents the experiments performed to evaluate the possible improvement of U(PE)^2, which exploits knowledge about the users' behavior, with respect to UPE, which implements a pure collaborative filtering algorithm. Conclusions are drawn in the last Section.

2 DIFFERENT APPROACHES TO RECOMMENDATIONS

There are many different techniques for implementing recommender systems (Resnick and Varian, 1997; Schafer, Konstan and Riedl, 1999; Terveen and Hill, 2001):

– *Collaborative filtering* is the most successful recommender system technology to date. The main idea is to recommend new items of interest for a particular user based on other users' ratings. These systems recommend products to a customer based on the correlation between that customer and other customers who showed interests in those products, e.g. who have purchased products from the e-commerce site.
– C*ontent-based* recommender systems suggest items based on their associated features. A pure content-based recommender system is one in which recommendations are made for a user based solely on a profile built by analyzing the content of items which that user has rated in the past.
– *Demographic* recommender systems aim at categorizing the user based on personal attributes and make recommendations based on demographic classes. The benefit of the approach is that it may not require a history of user ratings of the type needed by collaborative and content-based techniques.
– *Knowledge-based* recommenders attempt to suggest items based on inferences about a user's needs/preferences. In some sense, all recommendation techniques could be described as doing some kind of inference. Knowledge-based approaches have knowledge about how a particular item meets a particular user need, and can reason about the relationship between a need and a possible recommendation.

2.1 Collaborative Filtering Systems

Collaborative filtering is a type of recommendation technique that works by finding patterns of agreement among users of the system, leveraging the tastes and opinions about quality of all of the users to help each user individually.

Rather than recommending items because they are similar to items a user has liked in the past, a set of items that other similar users have liked is recommended. In other words, similarity of users rather than similarity of the items are computed. Typically, for each user a set of “nearest neighbor” users is found whose past rates have the strongest correlation. Rates for unseen items are predicted based on a combination of the rates known from the nearest neighbors. Pure collaborative recommendations give the possibility to deal with any kind of content. Since other users’ feedback influenced what is recommended, there is the potential to maintain effective performance given fewer rates from any individual user.

Collaborative filtering has a number of advantages over content-based methods:

- The knowledge engineering problem associated with content-based methods is relieved, since explicit content representations are not needed.
- The quality of collaborative filtering typically increases with the size of the user population, and collaborative recommendations benefit from improved diversity when compared to content-based recommendations.

However collaborative filtering does suffer from a number of significant downsides:

- It is not suitable for recommending new items because these techniques can only recommend items already rated by other users. If a new item is added to the content database, there can be a significant delay before this item will be considered for recommendation. Essentially, only when many users have seen and rated the item will it find its way into enough user profiles to become available for recommendation. This so-called “latency problem” is a serious limitation that often renders a pure collaborative recommendation strategy inappropriate for a given application domain.
- Collaborative recommendation can prove unsatisfactory in dealing with what might be termed an “unusual user”. There is no guarantee a set of recommendation partners will be available for a given target user, especially if there is insufficient overlap between the target profile and other profiles. If a target profile contains a small number of rates or ratings for a set of items that nobody else has reviewed, it may be difficult to make reliable recommendations using the collaborative technique.

2.2 User Knowledge

A key issue in the personalization of a Web site is the automatic construction of accurate user profiles. A profile is a collection of information about an individual; it permits to recognize the user, know why he or she did something, and guess what he or she wants to do next. User profiling is typically either knowledge-based or behavior-based. Knowledge-based approaches engineer static models of users and dynamically match users to the closest model. The knowledge about users can be acquired in different ways. Generally speaking, it could be acquired through questionnaires, where users select different content types and services from a list of predefined choices. This implies that users must manually update their profiles when their interests change. These limitations clearly call for alternative methods that infer preference information implicitly and support automated content recommendation.

Behavior-based approaches use the user’ behavior itself as a model. Machine learning techniques are being used to recognize the regularities in the behavior of customers interacting with e-commerce Web sites and to infer a model of the interests of a user, referred to as *user profile* or *user model*. The user model is a collection of information about an individual and should be able to recognize the user, know why he or she did something, and guess what he or she wants to do next. The typical user profiling approach for recommender systems is behavioral-based, using a binary model (two classes) to represent what users find interesting and uninteresting. Machine-learning techniques are then used to assess potential items of interest in respect to the binary model.

2.3 Integrating Collaborative Recommender Systems with User Knowledge

User models have been used in *recommender systems* for content processing and information filtering. It could be useful to develop methods for integrating behavioral profiling with collaborative filtering into one integral approach. In particular, the approach we propose integrates collaborative techniques with user profiles inferred from the analysis of transactional data (browsing and purchasing history of users) without considering any content. There are two main alternatives to accomplish this task:

1. *Profiles Drive Collaborative Methods*. Profiles are used to reduce the set of items that should be used for computing recommendations. This means that standard collaborative methods will be applied, but they will work on a smaller consideration set of data. We expect this to increase the performances of the overall

technique in comparison to the stand-alone collaborative filtering method.

2. *Profiles Are Used After Collaborative Filtering*. Standard collaborative filtering techniques are used to generate a preliminary set of possible recommendations. Then, profiles are exploited to re-rank the set of the recommended items or to prune some of the items that were preliminarily recommended.

Our approach exploits the first alternative, but it reduces the set of users on which the algorithm is applied instead of reducing the set of items. In Section 3.3 we will give more details about the adopted approach.

3 PERSONALIZATION SYSTEMS

In previous work, we have developed two personalization systems, each exploiting a specific technique for providing recommendation: UPE, described in Section 3.1, is a recommender system that uses filtering techniques (collaborative and simple filtering), and PE, described in Section 3.2, is a knowledge-based recommender system.

3.1 User Profile Engine

UPE (User Profile Engine) is a recommender system that provides personalized suggestions (recommendations) about pages users might find interesting in a product catalogue on the Web (Buono, et al., 2002). The user profiles managed by UPE have a static component and a dynamic one.

The static component consists of a set of information that identifies each user and doesn't change (or change rarely). For example: name, nationality and type of user. The information sources come primarily from the registration forms that some users are required to fill. The dynamic component of user profile is the changing part of user data. The set of user preferences is part of the dynamic profile. UPE obtains this information by using different type of ratings: explicit ratings, i.e. the user explicitly indicates what he or she thinks about an item; implicit ratings, obtained by tracking user navigation (i.e. events as access to a Web page, print and/or save action, etc.). Even if explicit rating is fairly precise, it has disadvantages, such as: 1) stopping to enter explicit ratings can alter normal patterns of browsing and reading; 2) unless users perceive that there is a benefit providing the rates, they may stop providing them.

Implicit ratings are much more difficult to determine but they have the following advantages: 1) every interaction with the system (and every absence of interaction) can contribute to implicit rating; 2) can be gathered for free; 3) can be combined with several types of implicit ratings for a more accurate rating; 4) can be combined with explicit ratings for an enhanced rating.

Indeed, the method that is quite effective is a mixed technique that exploits implicit and explicit ratings and we implemented it in UPE. However, especially in the case of sites with many pages, we can be in a situation that some pages have not been evaluated by the current user (neither explicit nor implicit ratings are available). To overcome this situation, UPE uses an algorithm of collaborative filtering. It predicts user interests on an item not evaluated by taking into account the historical data set on rates of a users community stored into a database of existing rating provided by other users (Buono, et al., 2002).

As it is well known, these algorithms are useful but also very time consuming. With the aim to further improve UPE performance, we have defined some heuristics that reduce the number of users involved in the computation of users' preferences. Such preferences are computed by using *weights* that reflect correlation (in this case the *Pearson* correlation) between pairs of users. The more objects two users have rated similarly, the closer the two users are. To reduce the number of computations, UPE re-calculates only the weights for users that at least one of the two users, during his or her interactions with the system, has produced a number of ratings (explicit or implicit) above a given threshold m. Furthermore, the system re-computes the predicted rating of a user for a certain item by taking into account only the users, that since the last rating updating, have generated a number of ratings (explicit or implicit) above a threshold n. More specifically the predicted rating is a weighted sum of ratings of the users selected for re-computation.

3.2 Profile Extractor

In order to provide personal recommendations based on a comprehensive knowledge of who customers are and how they behave, we have adopted an approach that uses information learned from transactional histories to construct individual

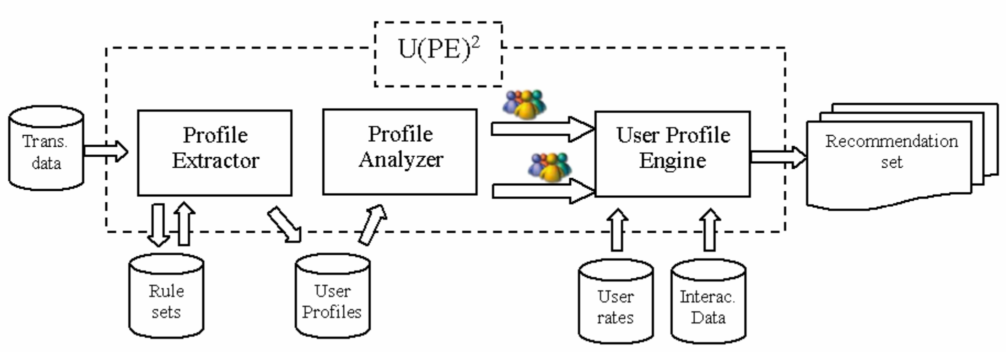

Figure 1: U(PE)^2 architecture

profiles. The advantage of using this technique is that profiles generated from a huge number of transactions tend to be statistically reliable.

The process of learning customer profiles is performed by the PE (*Profile Extractor)* personalization system (Semeraro, et al., 2003), which employs supervised learning techniques to automatically discover users' preferences from transactional data recorded during past visits to the e-commerce Web site. In Business to Consumer (B2C) e-commerce, items are grouped in a fixed number of categories. For example, at Amazon.com books in the catalogue are organized in many subject categories. PE is able to analyze data gathered from sources such as data warehouse or transactions, for instance, in order to infer rules describing the customer/user behavior. Rules are exploited to build profiles containing preferences such as the product categories the user is interested into.

From our point of view, the problem of learning user's preferences can be cast to the problem of inducing general concepts from examples labelled as members (or non-members) of the concepts. In this context, given a finite set of categories of interest $C = \{c_1, c_2, ...c_n\}$, the task consists in learning the target concept T_i "users interested in the category c_i". In the training phase, each user represents a positive example of users interested in the categories he or she likes and a negative example of users interested in the categories he or she dislikes. We chose an operational description of the target concept T_i, using a collection of rules that match against the features describing a user in order to decide if he or she is a member of T_i. Transactional data about customers are arranged into a set of unclassified instances (each instance represents a customer). The subset of the instances chosen to train the learning system has to be labeled by a domain expert, that classifies each instance as member or non-member of each category. The training instances are processed by the Profile Extractor, which induces a classification rule set for each category of interest. More precisely, the architecture of PE is made up of several sub-modules: (a) *XML I/O Wrapper*, which is the layer responsible for the extraction of data required for the learning process; (b) *Rules Manager*, which is implemented through one of the WEKA (Frank and Witten, 1998) classifiers. The learning algorithm adopted in the rule induction process is PART (Witten and Frank, 1999), which produces rules from pruned partial decision trees; (c) *Profile Manager*, which classifies each user on the ground of the users' transactions and the set of rules induced by the Rules Manager. The classifications, together with the interaction details of users, are gathered to form a *user profile*.

3.3 Integrating UPE and PE: U(PE)^2

Our idea is to produce a hybrid method by integrating behavioral profiles inferred by PE and the collaborative method implemented by UPE into one integral approach in an attempt to demonstrate that it outperforms the pure collaborative filtering method. The resulting system U(PE)^2 (Fig. 1) implements a cascade hybrid method: profiles inferred by PE are exploited by the Profile Analyzer to group customers having similar preferences. In our case, preferences are the product categories the customer is interested in. Our idea is that profiles could drive the collaborative method by reducing the set of users, on which the algorithm is applied, only to users interested in the same product categories. PE is applied to induce rules (describing "classes" of users) that are exploited to build the profiles. Then, the collaborative filtering algorithm is applied to each group of users selected by the Profile Analyzer. In this way, it is possible to improve computational performance by carrying out parallel computation for each group of users. We actually use PE to classify registered users and assign them to the content categories of their interest; we then apply collaborative filtering algorithm to the users of each class, in order to generate recommendations that fit their interests.

4 EXPERIMENTAL WORK

We have performed two experiments in order to compare the performance of the proposed hybrid recommender system $U(PE)^2$ with UPE. The former measures the evaluation of UPE implementing the classical collaborative filtering technique (see Section 3.1). The latter measures the evaluation of the hybrid system $U(PE)^2$ obtained integrating the behavioral profiles inferred by PE with the UPE collaborative method. The performance of $U(PE)^2$ has been compared with the UPE personalization system. For both experiments we used historical browsing data from an Italian e-commerce company. This dataset contains information about 380 users on 154 catalogue products; in particular, it contains explicit rates given by users and implicit rates computed by the system on the basis of the user behavior. Each action performed by a user on a Web page, for example zooming on the picture of a product, corresponds to a rate. We divided the dataset into a training set and a test set by using 90%/10% training/test ratio. From each user in the test set, ratings for 25% of items were randomly withheld. Predictions were computed for the withheld items using each of the different algorithms. In the first experiment, the dataset was converted into a user-product matrix that had 380 rows (i.e., 380 users) and 154 columns (i.e., products that were rated by at least one of the users). Predictions were computed for the withheld items using the pure collaborative filtering technique implemented by UPE. In the second experiment, the dataset was converted into 11 user-product matrices, each corresponding to a specific product category C_i in which PE classified the users. Each matrix had n_i rows (i.e., the number of users that PE has classified as interested in the category C_i) and 154 columns (i.e., products that were rated by at least one of the users). In this case, the UPE collaborative filtering was applied separately to each matrix. Both experiments were repeated 5 times selecting a different test set (the intersection of the five test sets was empty). This procedure allows running 5 experiments that are completely different. Finally, the results of experiment 1 were averaged over the 5 runs and ones of experiment 2 were averaged over all categories.

The quality of the predictions was measured by comparing the predicted values for the withheld ratings to the actual ratings, using several metrics.

In general, recommender systems research has used several types of measures for evaluating the success of a recommender system. We consider only two types of metrics for evaluating predictions and recommendations respectively.

To evaluate an *individual item prediction* we used the Mean Absolute Error (MAE) between ratings and predictions. MAE is a measure of the deviation of recommendations from their true user-specified values. For the prediction of N items ($p_1,\ldots,p_N$) and a real evaluation of a user ($r_1,\ldots,r_N$), $E = (|p_1\text{-}r_1|,\ldots,|p_n\text{-}r_N|)$ is calculated. We can compute MAE by first summing the squared absolute errors of the N corresponding ratings-prediction pairs and then computing the average. Since the task was to identify or retrieve items preferred by users from a repository, traditional information retrieval measures were adopted, namely *Precision (Pr)*, *Recall (Re)* (Sebastiani, 2002). We have adapted the definition of recall and precision to our case as our experiment is different from standard IR in the sense that we have a fixed number of recommended items. In the evaluation phase, the concept of *relevant item* is central. An item is considered as relevant by a user if the score he or she has given is greater than 2.5. An item is considered as relevant by a system if the computed numerical recommendation score is greater than 2.5. Our goal is to look into the test set and match items that both the system and the user deemed relevant. Then, *recall* is the proportion of relevant items that are classified as relevant, and *precision* is the proportion of items classified as relevant that are really relevant. The fact that both measures are critical for the quality judgment leads us to use a combination of the two. In particular, we use the standard *F1* metric (Sebastiani, 2002), which gives equal weight to them both. We also adopted the *Normalized Distance-based Performance Measure (NDPM)* (Yao, 1995) to evaluate the goodness of the items' ranking calculated according to a certain relevance measure. Specifically, *NDPM* was exploited to measure the distance between the ranking imposed on items by the user ratings and the ranking predicted by the system. Values range from 0 (agreement) to 1 (disagreement). Results of the experiments are divided into two parts: quality results and performance results. In assessing the quality of recommendations, we first analyze the results obtained in experiment 1 by UPE (Table 1).

Table 1 – Results obtained by UPE (averaged over 5 runs)

MAE	NDPM	Recall	Precision	F1-measure
0.421	0.066	0.942	0.905	0.923

Notice the high accuracy that can be achieved by the system on the whole dataset in predicting the ranking of the products according to the customers interests (the NDPM value is close to 0). The high value of the F1-measure and the balance between recall and precision demonstrates that the list of

recommendations presented to users by UPE contains relevant items correctly ranked.
In the second experiment, we examined separately the recommendation accuracy for users grouped according to their behavioral profiles. For each group, Table 2 reports the number of users classified as interested in that category, the number of users poorly, moderately, and strongly correlated, and the mean correlation value computed over each pair of users. A correlation coefficient between 0.3 and 0.6 reveals a moderate association, while values above 0.6 indicate a strong correlation. A coefficient at zero, or close to zero, indicates no relationship.
In Table 3, we reported the $U(PE)^2$ results. The values of MAE are positive given the small number of users belonging to each category (from 35 users in the category “kitchen utensils” to the 74 users in the category “underwear”). Only for two categories (“kitchen utensils” and “jewelry”) the value of MAE was over 2.
In particular, the computed MAE for the category “kitchen utensils” is greater than 8. In order to fully understand this result it is important to notice that this category contains the smallest number of users (35) and reported the lowest value of mean user correlation.
For the users in the category “kitchen utensils”, the computed value was 0.42 against at least 0.49 achieved in all the other categories (see Table 2 for more details). In general, NDPM results are very positive (values do not exceed 0.2), showing a strong correlation between the ranking imposed by the users and the ranking computed by the system, although there is a high degree of variation between different categories. NDPM is better for users strongly correlated and belonging to “more populated” categories: the best values have been found in the categories “underwear” (74 users) and “hardware”, which show the highest values of user correlation. For the F1 score, we consider the results as very positive. Overall, 8 out of the 11 categories reported values that exceed 0.80, while only for one category (“kitchen utensils” again) the system was not able to reach a value of at least 0.70.

The aim of the second experiment was to compare the results obtained by $U(PE)^2$, averaged over all the categories, with the results obtained by UPE (see Table 4).

As regards MAE, the value achieved by UPE is almost five times better than the value registered for U(PE)2. UPE outperforms U(PE)2 both for NDPM and F1-measure.

Table 2 – Statistics on the 11 products categories

DATASET	*# users*	*U.C.< 0.3*	*0.3≤U.C.≤0.6*	*U.C.> 0.6*	*Mean User Corr.*
UNDERWEAR	74	11 (15%)	10 (13%)	53 (72%)	0.50
FURNITURE	67	10 (15%)	14 (21%)	43 (64%)	0.51
PET SUPPLIES	69	10 (15%)	16 (23%)	43 (62%)	0.51
HOUSEHOLD ARTICLES	68	12 (18%)	17 (25%)	39 (57%)	0.49
KITCHEN UTENSILS	35	9 (26%)	5 (14%)	21 (60%)	0.42
SANITARY ARTICLES	70	9 (13%)	24 (34%)	37 (53%)	0.50
ELECTRONICS	59	9 (15%)	7 (12%)	43 (73%)	0.49
HARDWARE	65	12 (19%)	8 (12%)	45 (69%)	0.52
JEWELRY	70	14 (20%)	11 (16%)	45 (64%)	0.51
INFORMATICS	65	9 (14%)	13 (20%)	43 (66%)	0.51
BABYHOOD	70	8 (12%)	12 (17%)	50 (71%)	0.51
ENTIRE DATASET	**380**	**64 (17%)**	**41 (11%)**	**275 (72%)**	**0.65**

Table 3 – Results obtained by $U(PE)^2$

DATASET	MAE	NDPM	F1
UNDERWEAR	1,473	0,040	0,879
FURNITURE	0,900	0,166	0,947
PET SUPPLIES	1,228	0,062	0,888
HOUSEHOLD ARTICLES	1,395	0,135	0,847
KITCHEN UTENSILS	8,078	0,195	0,629
SANITARY ARTICLES	1,045	0,049	0,864
ELECTRONICS	1,130	0,061	0,799
HARDWARE	0,872	0,048	0,971
JEWELRY	2,496	0,109	0,724
INFORMATICS	0,939	0,155	0,862
BABYHOOD	1,734	0,072	0,822

Table 4 – Comparison UPE vs. $U(PE)^2$

MAE			NDPM			F1-measure		
UPE	$U(PE)^2$	|Diff.|	UPE	$U(PE)^2$	|Diff.|	UPE	$U(PE)^2$	|Diff.|
0.421	1.936	1.515	0.066	0.099	0.033	0.923	0.839	0.084

This result is to be expected, as the collaborative filtering algorithm implemented by UPE generates recommendations based on the strength of the association among users and it is adversely affected by reduced training sets containing poorly correlated users. Only 3 categories (“underwear”, “electronics”, “babyhood”) reported at least 70% of users strongly correlated, as in the original dataset, and that the mean user correlation observed in each category is always lower than in the entire dataset. Nevertheless, the results achieved using behavioral profiles are satisfactory: NDPM is still very close to 0 and F1-measure shows a classification accuracy in recognizing relevant items that is almost 84%. This means that $U(PE)^2$ is able to recommend “good” items, although the individual item prediction gets worse. When we focus on performance issues, we find the main advantage of grouping users according to their behavioral profiles before computing recommendations: the time requested by UPE to produce recommendations on the whole dataset of 380 users was 5h 47min, while the time requested by $U(PE)^2$ was 57min for computing recommendations and 1h 27min for classifying users into 11 categories. The total time for completing the process was 2h 24min.

5 CONCLUSIONS

Recommender systems are a powerful technology that allows a company to get additional value from its user database. A real problem is that these systems are being stressed by the huge volume of user data in existing corporate databases. A strong research issue is to develop methods that can improve the scalability of recommender systems, still producing high-quality recommendations. In this paper, we have presented a new approach for collaborative-based recommender systems. It integrates knowledge about customers stored in behavioral profiles into the collaborative filtering algorithm in order to reduce the computational time required for generating recommendations. The final goal of the work has been to identify some measures for evaluating the quality of recommendations. For this purpose, we have presented the empirical evaluation of the $U(PE)^2$ hybrid recommender system. Our results have highlighted the actual improvement of the proposed hybrid approach with respect to a pure collaborative approach. We can conclude that the proposed technique holds the promise of allowing collaborative-based algorithms to scale to large data sets, still producing high-quality recommendations.

REFERENCES

Balabanovic, M. and Shoham, Y., 1997. Fab: Content-Based, Collaborative recommendation. *Communications of the ACM*, 40 (3), 66-72.

Buono, P., Costabile, M. F., Guida, S., Piccinno, A., 2002. Integrating User Data and Collaborative Filtering in a Web Recommendation Systems. In S. Reich, M.M. Tzagarakis and P.M.E. De Bra (Eds.), *Hypermedia: Openness, Structural Awareness and Adaptivity*, LNCS, vol. 2266, 315-321. Springer, Berlin.

Burke, R., 2002. Hybrid Recommender Systems: Survey and Experiments. In *User Modeling and User-Adapted Interaction*, 12, 331-370. Kluwer Academic Publishers, the Netherlands.

Frank, E. and Witten, I., 1998. Generating accurate rule sets without global optimization. In *15th International Conference on Machine Learning*, 144–151. Morgan Kaufmann.

Konstan, J. A., Riedl, J., Borchers, A. and Herlocker, J. L., 1998. Recommender Systems: A GroupLens Perspective. In *Recommender Systems. Papers from 1998 Workshop*. Technical Report WS-98-08, 60-64. AAAI Press.

Pazzani, M., 1999. A Framework for Collaborative, Content-Based and Demographic Filtering. In *Artificial Intelligence Review*, 393-408.

Resnick, P. and Varian, H. R., 1997. Recommender Systems. *Communications of the ACM*, 40 (3),56-58.

Sarwar, B. M., Karypis, G., Konstan, J. and Riedl, J., 2002. Recommender Systems for Large-Scale E-Commerce: Scalable Neighborhood Formation Using Clustering. In *Proceedings of the 5th International Conference on Computer and Information Technology* (ICCIT 2002).

Schafer, J. B., Konstan, J. and Riedl, J., 1999. Recommender Systems in E-Commerce. In *EC ’99: Proceedings of the First ACM Conference on Electronic Commerce*, Denver, CO, 158-166.

Sebastiani, F., 2002. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34 (1), 1–47.

Semeraro, G., Abbattista, F., Degemmis, M., Licchelli, O., Lops, P., Zambetta, F., 2003. Agents, Personalisation, and Intelligent Applications. In Corchuelo, R., Ruiz Cortés, A. and Wrembel, R. (Eds.), *Technologies Supporting Business Solutions, Part IV: Data Analysis and Knowledge Discovery*, Chapter 7, 163-186. Nova Sciences Books and Journals.

Shardanand, U. and Maes, P., 1995. Social Information Filtering:Algorithms for Automating’Word of Mouth’. In *Proceedings of CHI ’95*, Denver, CO., 210-217.

Terveen, L. and Hill, W., 2001. Human-Computer Collaboration in Recommender Systems. In J. Carroll (Ed.), *Human Computer Interaction in the New Millenium*, Addison-Wesley, New York.

Tuzhilin, A. and Adomavicius, G., 1999. Integrating user behaviour and collaborative methods in recommender systems. In *CHI' 99 Workshop Interacting with Recommender Systems*, Pittsburgh, PA, USA.

Witten, I. H. and Frank, E., 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann Publishers, San Francisco.

Yao, Y. Y., 1995. Measuring Retrieval Effectiveness Based on User Preference of Documents. *Journal of the American Society for Information Science*, 46, 2, 133–145.