

Collaborative feature-combination recommender exploiting explicit and implicit user feedback

Markus Zanker and Markus Jessenitschnig

University Klagenfurt

Intelligent Systems and Business Informatics Research Group

Klagenfurt, Austria

{markus.zanker;markus.jessenitschnig}@uni-klu.ac.at

Abstract—Collaborative filtering (CF) is currently the most popular technique used in commercial recommender systems. Algorithms of this type derive personalized product propositions for customers by exploiting statistics derived from vast amounts of transaction data. Traditionally, basic CF algorithms have exploited a single category of ratings despite the fact that on many platforms a variety of different forms of user feedback are available for personalization and recommendation. In this paper we explore a collaborative feature-combination algorithm that concurrently exploits multiple aspects of the user model like clickstream data, sales transactions and explicit user requirements to overcome some known shortcomings of CF like the cold-start problem for new users. We validate our contribution by evaluating it against the standard user-to-user CF algorithm using a dataset from a commercial Web shop. Evaluation results indicate considerable improvements in terms of user coverage and accuracy.

Keywords-Collaborative filtering, hybrid recommendation methods, cold-start problem

I. INTRODUCTION

Recommender systems (RS) are an important aspect of e-commerce infrastructures that help to deliver a personalized shopping experience to users. They support customers by retrieving items from a large product catalog that will most probably match their interests and/or needs. In contrast to general information filtering applications, RS provide product proposals to users based on preference information about what they are looking for [1]. Collaborative filtering (CF) relies on human judgements i.e. items' ratings to determine the proximity of users' tastes [2], [3]. Traditionally Pearson correlation or the cosine between rating vectors are used as similarity metrics. In a second step, recommendations are computed based on positively rated items of the most similar neighbors of a given user.

In many commercial situations these user judgements are collected automatically, requiring no explicit user input. For instance sales transactions or pageviews can be interpreted as implicit ratings and exploited by collaborative filtering systems. *Amazon.com* [4] is the most prominent example of this type of system, providing implicit advice to users browsing the online shop such as *Users who bought this item a, also bought items b and c.*

Although quite popular, such collaborative recommenders suffer from several shortcomings. For instance, when rating tables are rather sparse computed similarities between users have only limited meaning. Furthermore, observations of first-time or anonymous users can only be exploited to a limited degree making it difficult to determine similar peers. This is also known as the *cold-start* problem for new users.

However, most current research on CF systems focuses on optimizing system accuracy for users with 20 or more ratings which is rather impractical for application domains where returning online visitors cannot be recognized or where few ratings can be collected. Therefore, this paper focuses on an approach that simultaneously considers multiple features for personalization, namely implicitly and explicitly collected forms of user feedback, such as navigation actions and the context the user is currently in. Instead of exploiting only a single category of ratings for determining similar users and making recommendations, we present a hybrid recommendation approach that utilizes a diverse range of input data. Furthermore, it dynamically exploits only these types of rating input that optimize the predictive accuracy of the system. This approach was subsequently verified with transaction data collected over a period of 14 months from a real-world Web shop, exploiting relationships like *Users who were in the same situational context like you and who navigated like you have, actually bought item a* and demonstrating that the hybrid outperforms the standard approach in terms of user coverage as well as accuracy.

Section II gives an overview of related work and the hybrid algorithm framework is presented in Section III. Furthermore, the evaluation methodology and results obtained from tests on a commercial dataset from the cigar domain are described in Section IV. Finally, conclusions are presented in Section V.

II. RELATED WORK

According to Burke [5] five different paradigms of RS exist. Collaborative filtering [2], [3], [6] is the most prominent approach and exploits similarities inherent in explicit and implicit rating information derived from users. In comparison, demographics-based approaches determine user similar-

ity solely on user characteristics such as age, income or education [7]. While collaborative and demographic information filtering methods do not require any additional product information, content-based filtering techniques depend on item representations in the form of vectors of words or feature-value pairs [8]. Those items in the catalog that are most similar to a query or to the user's profile are then recommended. Knowledge and utility-based approaches also require product knowledge. While the previously mentioned methods typically follow a one-shot interaction style, i.e. items are recommended with or without explicit user request, knowledge and utility-based methods rely on preference elicitation dialogues that explicate the users' needs and their contextual requirements [9], [10], [11], [12]. Knowledge-based RS then interpret deep domain knowledge in the form of mappings between abstract user preferences and required product characteristics [12]. Utility-based RS are comparable to their knowledge-based counterparts in the sense that they produce recommendations based on user preferences and product knowledge. However, utility-based systems do not possess explicit mapping rules but require definitions of utility values that specify how product characteristics contribute to the fulfillment of given user requirements [5]. Note, that this paper focuses solely on collaborative filtering methods and how they can be extended to better address 'cold start' problems. A comparative analysis between different recommendation methods such as variants of knowledge-based RS as well as collaborative and content-based filtering has been done in [13].

A considerable amount of work addresses the problem of providing recommendations for anonymous Web users or the 'cold start' recommendation problems for new users of collaborative filtering in general. For example, Mobasher et al. [14] exploit Web usage data for personalized pageview recommendations and employ a clustering approach to increase system performance. This paper also focuses on binary Web usage data, however it differentiates between different types of usage information. Therefore, the approach is also related to the α -community spaces model, where different user features termed α_i are employed for determining similar users [15]. Nguyen et al. determine clusters of similar users according to feature α_i and use a rule-based induction approach to derive recommendations from the corresponding cluster. As the method in [15] employs 'cold' user features like demographics it is capable to recommend items to users that have not provided ratings to the system, yet. The method presented in this paper can be seen as an improvement to [15] as it dynamically decides for each user which features are exploited.

Schein et al. [16] propose a single probabilistic framework that combines content and collaborative data to address cold-start recommendations. However, they clearly focus on the 'new item' problem, while this approach addresses the 'new user' problem when deriving personalized recommendations

for anonymous Web users. Jin et al. [17] introduced the idea of an automated weighting scheme for ratings. Their algorithm exploits the variance of ratings within different clusters of similar users.

Adomavicius et al. [18] presented extensive work on the use of contextual information in recommender systems and developed a multi-dimensional data warehouse approach that allows ratings to be predicted according to user context. Our work is related in the sense that it introduces different categories of ratings that may also contain contextual information. However, while [18] have to cope with more sparse rating tables, the approach taken in this paper exploits additional types of rating information in order to address sparsity problems caused by anonymous Web users.

Breese et al. [19] and Sarwar et al. [20] conducted extensive evaluations on partly commercial datasets and compared different algorithm variants. The methodology and evaluation design used to validate the approach proposed here is based on these works as well as on [21].

III. COLLABORATIVE FEATURE-COMBINATION HYBRID

Many hybrid algorithm designs have been explored in order to overcome various shortcomings like the cold start problems or data sparsity. Burke's taxonomy [5] enlists seven techniques for creating hybrid algorithms based on pure recommendation paradigms. Combining different input features is one option while other hybridization designs propose for instance running several recommenders in parallel and aggregating their results or feeding the output of one recommender into another one.

The feature combination hybrid presented here consists of a single recommender that utilizes a diverse range of input data administered by a generic user modeling component [22]. Basu et al. [23] proposed a feature combination hybrid that combined collaborative features such as user's likes and dislikes with content features of catalog items. As a result they identified new features like *users who like dramas* to determine similar peers within the community.

In contrast, our approach does not invent new features but assumes an array of different feature categories like navigation actions or sales transactions and decides dynamically for each user which of these categories to exploit in order to achieve the best predictive behavior within the recommendation engine. Formally, we assume that $R_{dom,u}$ returns a set of ratings on the domain dom for user u . For instance navigation actions (*nav*), viewed items (*view*), items added to the shopping basket (*buy*) or the context the user is currently in (*ctx*) are examples of different rating domains. The latter may be derived from a user's input to preference elicitation dialogues or search forms that are interpreted as unary ratings on keywords and short phrases and constitutes a valuable source for personalization as shown in [24]. For instance in our cigar example domain if a user is searching for a *gift* or replies *no experience*

if asked about smoking experience these terms indicate the user's intention and context.

When presented with a new user, a traditional CF approach would only use ratings from a single category and compute similar users from the available community transactions using a similarity metric. In this paper we employ cosine similarity. Thus, similarity between users u and v is derived from the cosine of their respective rating vectors R_u and R_v .

$$\cos(\vec{R}_u, \vec{R}_v) = \frac{\vec{R}_u \times \vec{R}_v}{|\vec{R}_u| \times |\vec{R}_v|} \quad (1)$$

Note, that in case users explicitly rate items on a multi-point Likert scale Pearson coefficient would yield better results as the average rating values for each user are also taken into account.

Consequently, recommendations are derived by a function $rec_{cf}(i, u)$ that computes a recommendation score for item i from u 's neighborhood of users N_u (i.e. the set of peers from the community that are most similar to u). The neighborhood size is typically limited by a parameter k , i.e. $|N_u| \leq k$, and an item's score is derived from the similarity of peers that rated it positively.

$$rec_{cf}(i, u) = \frac{\sum_{v \in N_u} score_{i,v}}{|N_u|}, \quad \text{where} \quad (2)$$

$$score_{i,v} = \begin{cases} \cos(\vec{R}_u, \vec{R}_v) & : i \in R_v \\ 0 & : \text{else} \end{cases}$$

Thus, for each user u , a recommendation system RS will thus output a list of the n top scoring items.

$$RS(u, n) = \{i_1, \dots, i_k, \dots, i_n\}, \quad \text{where} \quad (3)$$

$$\forall k \ score_{i_k, u} > 0 \wedge score_{i_k, u} > score_{i_{k+1}, u}$$

While the aforementioned pure collaborative filtering approach considers only a single set of ratings, a feature combination hybrid (fch) digests several rating sets $R_{d,u}$ from d different domains or categories and computes similarity between peers as a weighted sum over all d categories.

$$sim_{fch}(u, v) = \sum_d w_d \times \cos(\vec{R}_{d,u}, \vec{R}_{d,v}) \quad (4)$$

Note, that w_d represents a weighting factor for the rating domain d . The idea of associating some form of weight to specific user ratings is not new. For instance, Herlocker et al. [21] discuss the idea of associating a user's *confidence* and *strength* in a specific rating and Jin [17] propose the automated computation of weights depending on a rating's variance in different user clusters. However, both do not consider different types of rating categories per se.

In addition, some additional dynamic aspect was introduced. The algorithm implementation adaptively differentiates between the input from the different rating domains

based on an additional decision criteria. First, rating domains are prioritized based on their predictive power. Some domains are highly predictive like the user's context or actual purchases, but these rating categories are not always available in sufficient numbers. On the other hand, implicit user feedback like navigation actions is more abundant but more noisy and thus not so valuable for determining similar peers and making predictions. Therefore, an adaptive approach is proposed that selectively includes rating sets as recommendation input based on their predictive power. The latter can be determined either by insight and domain expertise or by empirical methods like offline experiments. In subsection IV-C1 we give experimental results for the employed dataset.

Formally, irrespective of the user R_{d_1} precedes (\prec) R_{d_2} if a CF recommender exploiting d_1 alone yields a higher accuracy than taking only d_2 as rating input. Initially the algorithm utilizes solely the input with highest priority, but if it cannot derive the required number of recommendations, additional lower priority rating sets are also included. Thus, the algorithm initially starts with the highest priority rating set as the threshold rating domain d_t and relaxes it to lower priority ones if necessary.

For instance, the system will attempt to compute n recommendations for the given user based on user context and actual purchases. It utilizes navigation actions only if context and purchase information is insufficient to generate the required number of recommendations.

Consequently, the enhanced feature combination hybrid (fch^*) algorithm computes similarities between users based on the given threshold domain d_t .

$$sim_{fch^*}(u, v, d_t) = \sum_{d \preceq d_t} w_d \times \cos(\vec{R}_{d,u}, \vec{R}_{d,v}) \quad (5)$$

Note that user neighborhood is determined analogously to that of traditional CF. Thus, when computing the recommendation score a given priority threshold on the rating categories d_t is assumed. In addition, when several different rating domains are combined as input, the output range of the recommendation system may also be varied, i.e. items rated by similar peers can in principle be recommended. Thus the output range of the recommendation function can be configured to reflect a specific rating domain such as viewed or bought items. The following equation therefore uses the parameter R_{rec} to restrict candidates for scored items to the desired rating set.

$$rec_{fch^*}(i, u, d_t, d_{rec}) = \frac{\sum_{v \in N_u} score_{i,v}}{|N_u|}, \quad \text{where} \quad (6)$$

$$score_{i,v} = \begin{cases} sim_{fch^*}(u, v, d_t) & : i \in R_{d_{rec}, v} \preceq R_{d_t} \\ 0 & : \text{else} \end{cases}$$

Table I
DIFFERENT FORMS OF USER FEEDBACK

User	$R_{nav,u}$	$R_{view,u}$	$R_{ctx,u}$	$R_{buy,u}$
Alice	n_1, n_2, n_5	i_1, i_3, i_5	k_1, k_3	i_1
Bob	n_3, n_4	i_3, i_5, i_7	\emptyset	i_3
Carol	n_2, n_3, n_4	i_2, i_4, i_5	k_2, k_4	i_4

Table II
FEEDBACK FROM NEW USER

User	$R_{nav,u}$	$R_{view,u}$	$R_{ctx,u}$	$R_{buy,u}$
Doreen	n_3, n_4	i_5	k_5	\emptyset

If the recommender does not derive the required number of recommendations, the whole process is repeated after relaxing d_t , thus including additional rating input.

For illustrative purposes, let us consider an example based on the community data in Table I. It presents examples of the different feedback categories provided by several users. Navigation actions n are implicitly observed activities of the user, such as menu selections, while the user's context stems from entered keywords k that constitute explicit requirements. Views and purchases are both - again implicitly collected - ratings on items i from the product catalog. Obviously, views on the *more details* page of an item occur more often than actual purchases.

When a new user enters the system little user feedback is available as depicted in Table II. None of Doreen's different rating domains contain sufficient feedback to confidently determine which user in Table I is most similar. However, a feature combination approach may utilize all four types of ratings as input. For this example we assume that all rating types are uniformly weighted ($w_d = \frac{1}{4}$) and that the following precedence rule is applied: $R_{buy} \prec R_{ctx} \prec R_{view}, R_{nav}$.

Thus the algorithm initially uses ctx as the threshold domain to find similar peers and derive recommendations. Note, R_{ctx} is the highest ranked non-empty rating domain. It can be easily shown that when comparing solely R_{ctx} there is no overlap between *Doreen* and any other user and thus no similar peers can be determined. Therefore, the threshold domain has to be relaxed to include the two lower ranked domains $view$ and nav in order to extend the rating base considered for neighborhood formation. It turns out that based on the cosine similarity metric user *Bob* (B) is most similar to *Doreen* (D):

$$sim_{fch^*}(D, B, nav) = \sum_{d \leq nav} w_d \times \cos(\overrightarrow{R_{d,D}}, \overrightarrow{R_{d,B}}) =$$

$$\frac{1}{4} \times 0 + \frac{1}{4} \times 0 + \frac{1}{4} \times \frac{1}{\sqrt{3}} + \frac{1}{4} \times 1 = 0.39$$

Therefore, if the algorithm only recommends items that have been actually purchased by the similar peers, i.e. $d_{rec} = buy$, then item i_3 will be recommended to *Doreen*.

Table III
EVALUATION DATASET

	R_{buy}	R_{view}	R_{nav}	R_{ctx}	R
Users	1697	1697	1432	66	1697
Items	320	395	51	78	446
Ratings	2756	22985	6640	677	29625
Avg. ratings/user	1,62	13,54	4,64	10,26	17,46
Sparsity	0,9949	0,9657	0,9091	0,8685	0,9609
$ R_{d,u} \leq 2$	585	1306	1260	66	1540
$ R_{d,u} \leq 3$	248	1114	923	66	1435
$ R_{d,u} \leq 5$	60	916	571	66	1159
$ R_{d,u} \leq 8$	8	738	237	56	942
$ R_{d,u} \leq 12$	2	568	78	16	752

In the next section the approach is evaluated using an historical dataset collected from an e-shop for premium cigars [25].

IV. EVALUATION

A comparative offline analysis on an historical dataset was conducted in order to evaluate different feature combination hybrid configurations. The purpose was to research the following questions:

- 1) What is the accuracy of different forms of rating input like navigation actions, pageviews and user context when predicting purchases?
- 2) How does a feature-combination hybrid improve results in terms of accuracy and user coverage compared to standard collaborative filtering?
- 3) What effect do varying weights and incremental inclusion of additional rating input have on the accuracy and user coverage of a feature combination hybrid?

The dataset used here was obtained from a commercial Web shop offering luxury goods such as cigars, wine and selected coffee blends. In addition to an online product catalog navigable via a hierarchical menu, the shop environment also includes a conversational sales advisory system that guides its users through the cigar selection process and provides them with knowledgeable recommendations and explanations. Users can therefore browse the platform in a product-centric way and may disguise their needs and requirements context in an online conversation. Users anonymously interact with the site and provide their login data only when placing an order. Therefore, the evaluation exercise does not identify returning visitors and interprets each visit as a separate user model. In the following we give details on the dataset itself.

A. Dataset

The Web shop offers a catalog of close to 400 products. Users may navigate through a two level menu hierarchy with 19 top-level product categories. Seven of the top-level categories are further structured into an additional 32 subcategories. Subsequently users must first click on menu categories before they can access pages with detailed

product descriptions. In addition, there is a conversational recommender system available that allows users to explicate their contextual requirements and retrieve appropriate product items. Over a period of 14 months we collected product-oriented pageviews (R_{view}), accesses to menu categories (R_{nav}), user input to the conversational recommender (R_{ctx}) as well as added products to their shopping basket (R_{buy}). As the evaluation of algorithms requires a success criteria, we restricted the more than 45,000 distinct user visits to those that concluded with an online purchase. Note that each anonymous online visit represents a distinct 'user' in our terminology. Table III gives size and structure of the evaluation dataset. The columns represent the different rating categories that were collected, while the rightmost column R describes the aggregated rating sets that encompass pageviews, menu navigation and user requirements. Clearly the number of user sessions with at least $n = 2, 3, 5, 8$ or 12 ratings during their visit is higher for the aggregated rating set than for any single rating category alone. Furthermore, the ratio of observations per user is highest when ratings are aggregated from the different categories. The sparsity of the evaluation dataset is quite high because the total number of items also increases. Note that both navigation actions and different explicit requirements are encompassed by the term *item*. The sparsity of each rating domain was computed with the following formula [21]:

$$sparsity = 1 - \frac{|R|}{|U| \times |I|}$$

Note, that for the purpose of replaying the experiments or comparing the proposed method to other approaches the dataset is available for download at <http://isl.ifit.uni-klu.ac.at>.

B. Methodology

In order to answer the research questions, the historic user data was utilized as input to the proposed algorithm. The evaluation followed the *Given n* method where n user ratings, i.e. the learning set, was randomly selected as input for determining similar users. The algorithms' goal was to correctly predict the actually purchased items of users, i.e. the test set.

$$testset_u = R_{buy,u}$$

$$\begin{aligned} |learnset_u| &= n \quad \wedge \\ testset_u \cap learnset_u &= \emptyset \end{aligned}$$

Obviously, $testset_u$ and $learnset_u$ may not overlap. Experiments were performed following a *leave-one-out* evaluation strategy where all but the current user are used for building a user/item matrix. For each user u , experiments were repeated ten times and given results are averages of all users and all

experiment runs. We used $k = 30$ as an upper limit for the k nearest neighbor approach. At most ten recommendations ($|recset_u| = 10$) were made, where only items from the recommendation domain d_{rec} were considered as candidates. Recommendations that were contained in the testset were assumed to be successful hits, i.e.

$$hits_u = recset_u \cap testset_u$$

For each experiment scenario we varied the learning set size such that $n = 3, 5, 8$ and 12 . Note, that the number of users that receive recommendations (User coverage) strongly depends on n . For instance in a Given 3 scenario on *views* observations exactly 3 viewed items are required for initializing the learning set and therefore all users with less than 3 *views* observations will definitely receive no recommendations (consult Table III for details of dataset observation frequencies).

The accuracy of recommendations was computed using Precision (P), Recall (R) and F1 metrics [6], [21].

$$\begin{aligned} P &= \frac{|hits_u|}{|recset_u|} \\ R &= \frac{|hits_u|}{|testset_u|} \\ F1 &= \frac{2 \cdot P \cdot R}{P + R} \end{aligned}$$

The Precision metric gives the share of successful recommendations from the total number of computed recommendations, while Recall metric computes the ratio of hits according to the testing set size, i.e. the size of the testing set constitutes the maximum number of hits. Therefore, we consider Recall to be more relevant in our situation. The average test set size is 1,6 (compare Table III - column R_{buy}). Consequently, if we could predict all items in the testing set, Recall would be 100% but Precision could be at most 16%. The F1 measure combines both Precision and Recall. Another important metric in our evaluation is User Coverage (Ucov).

$$Ucov = \frac{\sum_u |recset_u > 0|}{|U|}$$

It is defined as the share of users from the overall dataset that received a non-empty set of recommendations during the trial [21]. User Coverage is an especially important metric in this experimental setting due to the sparsity of user ratings and focuses on the algorithms' ability to make recommendations.

C. Results

1) *Experimental setup 1*: In our first experimental setup we addressed the question: *How appropriate are the different types of rating input like pageviews, menu navigation and*

Table IV
CF ON SINGLE RATING DOMAIN, $d_{rec} = buy$, $|U| = 1697$

Experiment	U_{cov}	R	P	$F1$
ctx.given3	4,03%	35,48%	5,46%	9,12%
ctx.given5	4,03%	35,86%	5,55%	9,28%
ctx.given8	3,30%	30,54%	4,91%	8,17%
ctx.given12	0,94%	34,69%	6,62%	10,63%
nav.given3	54,39%	29,09%	5,39%	8,69%
nav.given5	33,64%	27,28%	5,73%	8,99%
nav.given8	13,97%	24,78%	6,25%	9,42%
nav.given12	4,54%	21,86%	6,23%	9,02%

Table V
CF ON SINGLE RATING DOMAIN, $d_{rec} = view$, $|U| = 1697$

Experiment	U_{cov}	R	P	$F1$
ctx.given3	4,03%	37,92%	5,53%	9,32%
ctx.given5	4,03%	40,73%	5,82%	9,85%
ctx.given8	3,30%	35,39%	5,32%	8,94%
ctx.given12	0,94%	39,38%	7,50%	12,06%
nav.given3	54,39%	31,92%	5,96%	9,58%
nav.given5	33,64%	33,47%	6,96%	10,93%
nav.given8	13,97%	29,05%	7,29%	10,92%
nav.given12	4,54%	25,07%	7,31%	10,45%
view.given3	62,98%	34,26%	5,31%	8,83%
view.given5	58,27%	34,44%	5,61%	9,24%
view.given8	49,58%	34,34%	5,81%	9,49%
view.given12	40,18%	34,79%	6,20%	10,02%

contextual requirements for neighborhood formation and predicting shopping-cart actions?

As can be seen from Table IV-C1 and Table V all three partitions reach Recall values between 20% and 40%. Contextual requirements outperform all other forms of rating input, but this data is only available for very few users and thus user coverage is extremely low. We varied the recommendation domain d_{rec} between *buy* and *view* ratings, as shown in Table IV-C1 (using user's *buy* ratings) and in Table V (*view* ratings). Thus, user similarities were determined based on *ctx*, *nav* or *view* ratings, while *buy* or *view* ratings of their nearest neighbors were exploited to produce the actual recommendations. Note, however that the testset always consists of the *buy* ratings of the respective user. Interestingly, accuracy improved when using *views* as recommendation domain (see Table V). Although this may appear paradoxical at first glance, it can be explained by the much higher sparsity of the *buy* rating domain compared to the *view* ratings. Therefore, the *view* ratings in a user's neighborhood are more diverse and thus are more likely to produce a successful prediction. Therefore, we used $d_{rec} = view$ in all further experiments.

With respect to our research question, we can say that using user answers during preference elicitation dialogues for similarity calculation allows us to provide the best recommendations in terms of accuracy, followed by *views* and *menu* observations. Furthermore, requiring more than 5 observations does not improve accuracy results a lot, but obviously deteriorates User Coverage massively.

Table VI
FEATURE COMBINATION fch (UNIFORM WEIGHTS), $d_{rec} = view$, $|U| = 1697$

Experiment	U_{cov}	R	P	$F1$
ctx+nav+view.given3	72,72%	31,88%	5,05%	8,37%
ctx+nav+view.given5	61,17%	29,45%	5,00%	8,17%
ctx+nav+view.given8	50,52%	26,84%	4,71%	7,64%
ctx+nav+view.given12	40,31%	25,07%	4,58%	7,36%

Table VII
FEATURE COMBINATION fch (NON-UNIFORM WEIGHTS), $d_{rec} = view$, $|U| = 1697$

Experiment	U_{cov}	R	P	$F1$
ctx+nav+view.given3	72,72%	32,19%	5,10%	8,47%
ctx+nav+view.given5	61,11%	28,34%	4,89%	7,94%
ctx+nav+view.given8	50,56%	27,92%	4,78%	7,79%
ctx+nav+view.given12	40,31%	25,30%	4,61%	7,40%

2) *Experimental setup 2*: The second goal of our evaluation was to find out what improvements in terms of accuracy and user coverage can be obtained using a collaborative feature combination hybrid compared to a standard CF exploiting only a single type of rating input.

In order to create a comparative situation between CF using a single type of rating input (compare Table V) vs. using *ctx*, *nav* and *view*, n different ratings were randomly selected as the learning set and uniformly weighted, i.e. $\forall d \quad w_d = \frac{1}{3}$.

Table VI presents the performance of the feature combination hybrids. As can be easily seen, User Coverage is much higher and the feature combination hybrid used in the Given 3 experiment can produce recommendations for 72% of all user sessions which results in an increase of nearly 10% compared to the baseline algorithm CF on the *view* input (see Table V). Despite the fact that accuracy decreases slightly when different types of observations are arbitrarily combined, the number of overall hits (i.e. correct predictions for all users) was nevertheless increased by over 9% in this scenario.

However, when the number of user observations increases, *Given* $n \geq 5$, accuracy deteriorates sharply for the feature combination hybrid and the baseline experiment design (see Table V) catches up in terms of User Coverage.

We subsequently explored **how weighting and prioritizing the rating domains** impacts the overall performance of the collaborative feature combination hybrid. As can be observed in Table V, utilizing user's contextual requirements (*ctx*) resulted in the highest predictive accuracy followed by *view* and *nav* rating domains. Therefore, we used the following weights when combining different rating input: $w_{ctx} = 0.43$, $w_{view} = 0.35$ and $w_{nav} = 0.22$. Thus, the user's context is twice as important as navigation actions and pageviews lie in between. Table VII presents the performance of the feature combination hybrid using a non-uniform weighting of input ratings. However, introducing

Table VIII
FEATURE COMBINATION fch^* (NON-UNIFORM WEIGHTS),
 $d_{rec} = view, |U| = 1697$

Experiment	U_{cov}	R	P	$F1$
ctx+nav+view.given3	72,72%	35,13%	5,56%	9,22%
ctx+nav+view.given5	61,17%	34,61%	5,82%	9,52%
ctx+nav+view.given8	50,56%	35,53%	5,99%	9,79%
ctx+nav+view.given12	40,31%	34,77%	6,17%	9,98%

weights led only to very slight improvements compared with the experiments given in Table VI. Finally, the enhanced feature combination hybrid fch^* that exploits priorities on the rating input is evaluated. The assigned ordering attributed contextual requirements with the highest priority followed by pageviews and navigation actions: $R_{ctx} \prec R_{view} \prec R_{nav}$. Based on this algorithm configuration, the feature combination hybrids not only achieved significantly higher user coverage but also outperformed the baseline CF in terms of accuracy (see Table VIII). Neither Recall nor Precision deteriorated as n increased but remained approximately at the same level. The reason for this lies in the priority scheme that only takes lower priority rating input into account if higher priority rating domains do not suffice to compute the required number of recommendations. When analyzing the results for the single rating domain input (Table V) we can observe that accuracy deteriorates when users provide more nav actions as implicit feedback. This is in contrast to other rating domains where typically more information about a user's behavior does improve recommendation results. Therefore, the fch algorithm variants (Tables VI and VII) that exploit all three rating domains (ctx , nav and $view$) show a comparable deterioration with increasing n for every user. Due to the prioritization of rating domains where nav actions are assigned the lowest priority, fch^* only considers users' navigation actions if no recommendations would otherwise be computable.

Concluding, this evaluation of the proposed collaborative feature combination approach showed that using different types of implicit and explicit rating input yields better results in terms of User Coverage and also in terms of accuracy. The first result appeared likely from the beginning, i.e. additional rating input increases a recommender's potential to derive recommendations. The second improvement is the result of adaptive algorithm behavior which selectively combines inputs with different predictive accuracy. It was achieved by prioritizing rating domains and only exploiting additional rating input when necessary as proposed by fch^* algorithm.

V. CONCLUSIONS

This paper contributed an adaptive collaborative feature combination hybrid that generalizes the popular collaborative filtering recommendation approach to multiple categories of ratings. It combines several user inputs based on a weighting and priority scheme that includes lower priority

ratings only when necessary to boost the algorithm's User Coverage. An evaluation on a commercial dataset clearly supports the applicability of the approach and demonstrates the possible improvements in terms of user coverage and accuracy. The evaluation of this principle for other types of rating input as well as in different application domains remains for authors' future work.

REFERENCES

- [1] H. H. Sung, "Helping Customers Decide through Web Personalization," *IEEE Intelligent Systems*, vol. 17(6), pp. 34–43, 2002.
- [2] P. Resnick, N. Iacovou, N. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in *Computer Supported Collaborative Work (CSCW)*, Chapel Hill, NC, 1994, pp. 175–186.
- [3] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, pp. 230–237.
- [4] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *Internet Computing, IEEE*, vol. 7, no. 1, pp. 76–80, 2003.
- [5] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12(4), pp. 331–370, 2002.
- [6] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based Collaborative Filtering Recommendation Algorithms," in *10th International World Wide Web Conference*, 2001, pp. 285–295.
- [7] D. Billsus and M. Pazzani, "Learning collaborative information filters," in *International Conference on Machine Learning*, 1998, pp. 46–54.
- [8] M. Balabanovic and Y. Shoham, "Fab: Content-based, collaborative recommendation," *Communications of the ACM*, vol. 40(3), pp. 66–72, 1997.
- [9] R. D. Burke, K. J. Hammond, and B. C. Young, "The findme approach to assisted browsing," *IEEE Expert*, vol. July/Aug., pp. 32–40, 1997.
- [10] H. Shimazu, "Expert clerk: Navigating shoppers' buying process with the combination of asking and proposing," in *17th International Joint Conference on Artificial Intelligence (IJCAI)*, 2001, pp. 1443–1448.
- [11] F. Ricci, "Travel recommender systems," *IEEE Intelligent Systems*, vol. 17(6), pp. 55–57, 2002.
- [12] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker, "An integrated environment for the development of knowledge-based recommender applications," *International Journal of Electronic Commerce*, vol. 11, no. 2, pp. 11–34, 2006.
- [13] M. Zanker, M. Jessenitschnig, D. Jannach, and S. Gordea, "Comparing Recommendation Strategies in a Commercial Context," *IEEE Intelligent Systems*, vol. 22, no. May/Jun, pp. 69–73, 2007.

- [14] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, "Improving the effectiveness of collaborative filtering on anonymous web usage data," in *International Workshop on Intelligent Techniques for Web Personalization held in conjunction with IJCAI-01*, 2001.
- [15] A.-T. Nguyen, N. Denos, and C. Berrut, "Improving new user recommendations with rule-based induction on cold user data," in *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*. New York, NY, USA: ACM, 2007, pp. 121–128.
- [16] A. Schein, A. Popescul, L. Ungar, and D. Pennock, "Methods and metrics for cold-start recommendations," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002, pp. 253–260.
- [17] R. Jin, J. Chai, and L. Si, "An automatic weighting scheme for collaborative filtering," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004, pp. 337–344.
- [18] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Transactions on Information Systems*, vol. 23(1), pp. 103–145, 2005.
- [19] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *14th Conference on Uncertainty in Artificial Intelligence (UAI)*, 1998, pp. 43–52.
- [20] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," in *ACM Conference on e-Commerce (EC)*, 2000, pp. 158–167.
- [21] S. Jung, K. Harris, J. Webster, and J. Herlocker, "Serf: integrating human recommendations with search," in *Thirteenth Conference on Information and Knowledge Management (CIKM)*, 2004.
- [22] M. Jessenitschnig and M. Zanker, "A generic user modeling component for hybrid recommendation strategies," in *11th IEEE Conference on Commerce and Enterprise Computing (CEC)*. Vienna, Austria: IEEE, 2009.
- [23] C. Basu, H. Hirsh, and W. Cohen, "Recommendation as classification: using social and content-based information in recommendation," in *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI)*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1998, pp. 714–720.
- [24] M. Zanker and M. Jessenitschnig, "Case-studies on exploiting explicit customer requirements in recommender systems," *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*, A. Tuzhilin and B. Mobasher (Eds.): *Special issue on Data Mining for Personalization*, vol. 19, no. 1-2, pp. 133–166, 2009.
- [25] M. Zanker, M. Bricman, S. Gordea, D. Jannach, and M. Jessenitschnig, "Persuasive online-selling in quality & taste domains," in *7th International Conference on Electronic Commerce and Web Technologies (EC-Web)*. Krakow, Poland: Springer, 2006, pp. 51–60.