High Quality Recommendations for Small Communities: The Case of a Regional Parent Network

Sven Strickroth
Department of Informatics,
Clausthal University of Technology
Clausthal-Zellerfeld, Germany
sven.strickroth@tu-clausthal.de

Niels Pinkwart
Department of Informatics,
Clausthal University of Technology
Clausthal-Zellerfeld, Germany
niels.pinkwart@tu-clausthal.de

ABSTRACT
Traditional recommender systems are well established in scenarios in which "enough" items, users and ratings are available for the algorithms to operate on. However, automatic recommendations are also desirable in smaller online communities which only contain several hundred items and users. Collaborative filters, as one of the most successful technologies for recommender systems, do not perform well here. This paper argues that recommender systems can make use of contextual information and domain specific semantics in order to be able to generate recommendations also for these smaller usage scenarios.

The new hybrid recommendation approach presented in the paper enhances traditional neighborhood-based collaborative filtering techniques through the use of new kinds of data and a combination of different recommendation methods (rule, demographic, and average based). While the algorithmic techniques presented in this paper are suitable (especially) for smaller online communities, they can also be applied to improve the quality of recommendations in larger communities.

The approach was implemented and evaluated in a small regional bound parent education community. A multi-staged evaluation was conducted in order to determine the quality of recommendations: A cross-validation (recall), an expert questionnaire (recommendation quality) and a field study (user satisfaction). The results show that recommenders even for smaller communities are possible and can produce high quality recommendations.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Retrieval and Search—*Information Filtering*

Keywords
Collaborative Filtering, Recommender System, Small Communities, Sparsity, Implicit Ratings.

1. INTRODUCTION
Recommender systems are very popular both for E-Commerce (e.g. Amazon, Netflix) and the research community [3, 4, 17, 20], as these can calculate potential interesting items for users based on their interests. One of the most successful technologies for this task is Collaborative Filtering (CF) [4, 20]. These traditional recommender systems are well investigated and well established in scenarios in which a big amount of items, users and ratings are available for the algorithms to operate on (like MovieLens, Netflix, Amazon, …).

However, automatic recommendations are also desirable in smaller online communities which only contain several hundred items and users: Finding interesting items within several hundred alternatives already is not an easy, but time consuming task. It is often assumed that recommendation cannot work or performs badly in such scenarios without special adjustments [20]. Thus, there was little research on this field and there are no optimized approaches available.

We present a new hybrid approach especially suited for smaller online communities. Our approach enhances traditional neighborhood-based collaborative filtering techniques through the use of new kinds of data, contextual information from the community itself and a combination of different recommender methods. We will argue in this paper that recommender systems following this approach are able to generate recommendations also for these smaller usage scenarios.

We designed, implemented and evaluated our approach in a small regional bound parent education community.

In the next section we describe our usage scenario and why recommending interesting articles/items automatically is important in our scenario. The following section illustrates our algorithmic approach and describes the whole recommendation process in detail. The fourth section presents the evaluation of our hybrid recommender system prototype within our usage scenario.

2. USAGE SCENARIO
Our approach was implemented and evaluated in the context of the "Mobile2Learn" community in Germany [21]. This is a small parent community which focuses on education and upbringing of young children (ages up to 6), membership is free. Mobile2Learn is not a plain online community, but a community which combines "real world" events and workshops with new media (a Web 2.0 online platform and mobile phones).

The motivation for creating this community was that appropriate early childhood education is very important for the development of young children. Parents have a major influence on the process of early childhood education processes and development phases of their children. [2] (German context) and [5] suggested that the social origin and education of parents have a bearing on the future and academic achievements of their children. However, it is not easy to reach and motivate especially underprivileged parents – but it is well known that underprivileged parents need more assistance and normally do not take part in (offline) workshops about educational topics cf. [1] for a German report about parental training. Hence, the Mobile2Learn project was started by the community college Goslar, LEB (an institution for rural parent

RecSys'12, September 9–13, 2012, Dublin, Ireland.

education) and Clausthal University of Technology. The goal of this project was to combine the advantages of regional (offline) workshops/events and new media: parents could access pictures and content of the workshops online (anonymously) and could stay in contact with each other. Also, through the online portal, an intention of the project was to motivate parents to attend events.

In the Mobile2Learn project, the educational activities were structured into six areas (including "learning with all senses", musical education, nature discovery and "speaking and listening"). For each area, six independent thematically related events in different kindergartens were conducted; access to these events was open for everyone (not only parents whose children attend this facility). During an event, all parents were asked to sign on an attendance list. After all events of an area were held, taken photos and articles about the contents of the events were put online. So parents could look up all contents (again) and could find further information. Moreover, also parents who did not attend any events could view the photo galleries and could possibly be motivated to attend events in the future.

In order to reach parents of the target group, a network of partners was established: people and institutions which were in direct contact with parents and where parents go to if they need help (e.g. child doctors, youth welfare office and so on). Partners of this network helped to advertise the project. They were asked to invite and motivate parents to attend a workshop or to register on the online platform.

As already mentioned, the contents of the workshops were available as articles and interactive quizzes on the online platform. A few articles were available without registration on the platform to motivate parents to register. A registration in the system was required, however, to access most articles and all photo galleries. For registration users had to enter an e-mail-address, mobile phone number (optional), nickname, date of birth, gender and residence. Also date of birth and gender for their children could be entered optionally. Most of the articles and photos were provided by pedagogues, but it is possible for parents to submit their own articles or pictures (which were reviewed by pedagogues before publishing). All articles and photo galleries could be rated with one to five stars (dislike to like) and could be commented on by community members. Apart from this, many articles were annotated with ages of children to which they mainly apply to.

A key point of the online platform was that regular personalized messages (campaigns) were sent to parents via SMS and/or e-mail. These campaigns informed parents about upcoming events, new articles or pictures. Also, automatically generated recommendations were included in the e-mail campaigns (recommendations generated by the approach we describe in the next section). This repetitive, direct way of contacting community members was chosen to continuously and actively "push" information about educational opportunities to parents. So, parents were regularly reminded that the project still exists and could easily access all new and "interesting" items.

The Mobile2Learn.de online community was launched in November 2010. As early as this date, users found a small number of 30 articles on that website, which could be commented and rated (with one to five stars). By April 2012 about 250 users were registered; 170 articles and 65 photo galleries were available.

3. ALGORITHMIC APPROACH

In the Mobile2Learn environment it is crucial that users find interesting articles easily – otherwise it is not possible to successfully deliver our educational contents to our users. Users of the target group have children (i.e. not much time) and might be unwilling to search for interesting articles. Thus, the addition of the website with a recommender system was decided and planned by April 2011. As rating of items was possible since the start of the website and users already read and rated some articles, we did not have to deal with a new system/community.

There is a set of some special properties in our scenario "Mobile2Learn" (which are probably symptomatic for many smaller communities): Small size (few users and items), data sparsity (sparsity level very close to 1; 60 ratings, 151 items, and 175 users at design time), different item types (articles, quizzes, photo galleries) and combination of online/offline items (events).

There seems to be a lack of research focusing on automatic recommendation for small communities. To our knowledge there are no implementations which handle all of these properties. However, there are approaches which handle single aspects: Knowledge-based filtering [3] can easily be used for small datasets (especially if they can be objectively classified on a set of different criteria). They do not rely on ratings or user profiles and allow users to pull recommendations. Filters can also use cold-user data (e.g. demographic data from user profiles) to generate recommendations for users who have not rated (many) items yet [15]. Content-based algorithms [17] can find and calculate similar items based on annotations/tags or content of the items (no ratings required). For recommendations, however, ratings or other (interest-)profiles are required. Collaborative filters [7, 19, 20] are domain independent and can incorporate different item types easily. Instead of using different algorithms, it is possible to (actively) reduce a high sparsity level: A possible approach is aggregating/clustering of articles [10]. Here, several articles are combined/clustered to one (dummy) article with a shared rating (e.g. (un)weighted arithmetic mean). (Content-)Boosting [12] employs a content-based recommender to automatically generate predictions for every single user. Then the generated predictions are used as user ratings and included in a collaborative filtering recommendation process (which generates the final recommendation for the users). [16] and [18] suggested to automatically create ratings for dummy users in order to reduce sparsity and overcome cold-start issues. Apart from relying on (few) explicit ratings, also implicit ratings based on user behavior can be used [9, 13].

Knowledge-based recommenders would work for our structured educational articles (e.g. by article types or prerequisites on age of the children), however, they require active cooperation from users (as well as some content-based approaches). The quality of content-based recommenders heavily depends on consistent tagging and/or good metrics to automatically calculate item similarities. In our scenario, tagging is not an easy task, since within a thematic area all articles require quite similar tags and multiple users are providing content. Furthermore, without good tagging automatic calculation of similarities of different item types (such as photo galleries, articles and quizzes) is impossible. Apart from this, recommendations only contain very similar items and are quite static [18]. Accordingly, those are not applicable in our scenario. As already mentioned, collaborative and demographic filters require special adjustments or "enough" data to work on or in order to generate high quality recommendations. CF based approaches like clustering are not appropriate for smaller scenarios, since these reduce the number of items again – to an even lower number. Also it is not possible to recommend single items, but only whole clusters. A combination of different approaches might be the key for smaller scenarios. Content-boosting, however, is not a viable option since it requires a good content-based recommender. As mentioned above, in the

Mobile2Learn.de scenario, there are few ratings and most articles are quite similar (due to their structure and tags).

We attempted to solve this task by using a hybrid recommender system which incorporates different approaches. The proposed recommender generates a TOP-5 ranking of articles, photo galleries and upcoming events which the user might like most (w/o the presentation of a prediction). To effectively overcome the high sparsity level, implicit ratings were taken into account (this way we gained 1438 ratings; sparsity level could be reduced to 0.94). For this reason, visit times of articles (based on the webserver log) were transformed into ratings. Additionally, our approach makes use of several different sources of data: Demographic information from the user profiles (residence, age, gender) and two types of contextual and sematic information (as [14] has demonstrated that using underlying semantic can improve recommendation quality): (entered and inferred) age of the user's children and location of residence together with information about visited events (based on attendance lists of presence events in kindergartens). Apart from this data, structural information about the articles is also included into the recommendation process to leverage the new item cold-start problem: System-internally, articles are structured as a tree. Whereas the leaves contain the main (educational) content and the inner nodes are (topic) overviews and link pages. It is assumed that the distance of articles in the tree reflects the relatedness of the content. Based on the distance pseudo ratings are generated automatically for all articles. Even if the scenario is a smaller one, we focused on item based CF [19] which performs better than the user-based approach [7]. The entire recommendation process is made up of two phases: The preparation and model building phase as well as the recommendation generation phase (section 3.3 explains our choice of parameters).

3.1 Preparation and model building phase

The first phase consists of three steps: First visit durations are extracted from the webserver log and transformed into the same scale as the explicit ratings. The visit durations were calculated as follows (only for articles, not for other entities): We wrote a small log file parser, which measured the time between two HTTP requests for HTML pages of a user (we ignored images, CSS, JavaScript and AJAX requests). This way we got all visit durations within a session, except for the last article of a session if there is no user logout or other page request (since there is no following request which could be used to calculate the visit duration). We stored the maximum visit duration for each article per user over all sessions which were between three seconds and ten minutes in order to reduce the noise. This is important because we have to remove pages which were just skipped and it is not possible to measure the real reading durations, since we do not know if the user went away from the computer (e.g. to collect a coffee). Apart from these fixed limits, we implemented a dynamic limit to reduce noise more properly: After determining all visit durations and normalizing them by dividing by the number of words of the regarding article, we calculate the standard deviation and remove all visit durations which deviate more than two standard deviations from the average. In order to transform these normalized visit durations into our rating scale, we calculate the standard deviation (σ) again and use the transformation instruction (where m is the mean of all visit durations of a user) shown in Table 1. Using this transformation, we achieve that moderately long visited articles (the majority) get a moderate rating of three. Longer and shorter visited pages get higher resp. lower ratings exploiting the whole rating interval.

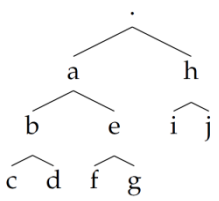
Table 1. Instructions for transforming normalized visit durations into ratings

Rating	Interval of normalized visit durations
1	$(-\infty; m - 1.64\sigma]$
2	$(m - 1.64\sigma; m - \sigma]$
3	$(m - \sigma; m + \sigma)$
4	$[m + \sigma; m+1.64\,\sigma)$
5	$[m + 1.64\sigma; \infty)$

[9] noted that one should not use negative feedback (e.g. ratings 1 and 2), because by observing the users behavior, one can only infer which items they probably like and thus chose to consume. In our scenario, however, the exclusive use of positive feedback led to excessive positive recommendations for some articles (especially longer ones). We think that in our context it is possible to infer by a small *maximum* visit duration that a user might not like an article, since an uninteresting article never gets read completely/for a long time without being skipped.

Contrary to [13], in our usage scenario it is necessary to normalize the durations with the length of the article before transforming them into ratings. Otherwise longer articles get higher ratings even if the normalized visit duration is the average or below. Too heterogeneous articles might be a reason for this.

The second step includes the generation of the pseudo ratings in order to integrate structural information of the article tree into the CF approach (inspired by [6]). This is done using an adapted filterbot approach [16]: We created filterbots/agents which rate items based on their position and relation in the article tree. On each node (which has at least one child) of the article tree we start a depth-first search which rates the node itself and all child nodes with a rating of 5. Fig. 1 shows an example article tree for which four filterbots are needed. Fig. 1 also shows the resulting ratings: Filterbot no. *1* starts in node *a* and creates ratings for *a* and all child nodes (*b*, *e*, *c*, *d*, *f*, *g*) of a. Filterbot no. 2 starts in node b and creates ratings for b, c and d. Filterbot no. 3 and 4 perform accordingly.



Filterbots:	1	2	3	4
r_a = (	5			)
r_b = (	5	5		)
r_c = (	5	5		)
r_d = (	5	5		)
r_e = (	5		5	)
r_f = (	5		5	)
r_g = (	5		5	)
r_h = (				5)
r_i = (				5)
r_j = (				5)

Figure 1. Generalized article tree and related filterbot ratings

By employing this schema (and using a decent metric, like the adjusted cosine similarity metric [19]), we achieve that articles, which are closely related (e.g. *c* and *d*) have a high similarity value ($s(r_c,r_d)=1$) while articles, which are not related (e.g. *f* and *i*) have a low similarity value ($s(r_f,r_i)=s(r_h,r_c)=0$). This schema is employed to be able to recommend new unrated items (new item cold-start problem) and to extend the neighborhood with articles of the same topic.

We also created one filterbot for each thematic area (e.g. "learning with all senses", see section 2) which generates ratings for all

articles and galleries of an area. This was done in order to create a link between the articles and the corresponding galleries.

Finally the similarity matrix for the collaborative filtering is calculated. Here the adjusted cosine similarity metric [19] is used. As [20] recommended, we only use positive similarity values (also using negative similarity values led to a worse recommendation quality). Being noisier than explicit ratings, implicit ratings are devalued (multiplied with 0.85) and overridden by explicit ones if available. Photo ratings are aggregated and averaged and stored as a rating for the photo gallery to which they belong. This aggregation is useful here, because it was not possible to generate reliable implicit ratings for photo galleries: Single photos are available through different URLs and change automatically after ten seconds in the slide mode. Also it does not make sense to recommend a single photo in our scenario.

Also, notable for the calculation of the similarity metric in small scenarios, the question how to handle items which have just one rating with exactly one user in common is notable. In smaller or real sparse scenarios those are not rare. If those items are not ignored, the similarity of two items will be 0 (pearson coefficient) or 1 (cosine based metrics) depending on the similarity metric. We evaluated different metrics with such similarities ignored and included (see section 3.3).

The preparation phase does not have to be executed for every recommendation request, but should be recalculated regularly (the system version reported on in this paper executes it once a day).

3.2 Recommendation generation phase

The recommendation generation is based on a multilevel approach with a decreasing degree of personalization and different approaches. If there were less than five candidates generated, the next (less personalized) level is executed until enough recommendations are available or the last level is reached. The criterion for considering an artifact a candidate for recommendation is a predicted rating that exceeds a threshold of 3 (out of 5).

The first level generates rule-based recommendations: Based on location of residence and on visited and upcoming events, recommendations for events are generated. Event recommendations based on collaborative filtering approaches are not possible in this scenario because all events are independent and articles which describe events are published after the events are held. Hence, we constructed simple rules in which contexts (specific) events are recommended: Having a database with personal information, we calculate the distance (in km) between the residence of the user and the location of an upcoming event. If the distance is less than 7 km (average distance + standard deviation of people who already visited events) or the user already attended an event in a location, we insert a recommendation for the nearest event within one topic/area with a "predicted" rating of 5. Such a recommendation stays in the recommendation list until the user clicked on it or the event took place. We assume that each new workshop topic is interesting for all parents. Additionally, recommendations for overview articles and photo galleries of visited events are generated if the user has not accessed these before. These rule-based recommendations are based on generic assumptions about the interest of people (e.g., if you were at a kindergarten workshop, then it is very likely that you'll be interested in the photos). The predicted ratings for recommendations of this type are based on the number of elapsed days since the event took place: We employed the exponential time function $\exp\left(\frac{-t}{\tau_0}\right)$ for this task, with a value of 810 for the half-time parameter T_0. This value was chosen, so that the items can show up in the top-5 recommendation list for 413 days (i.e. the predicted rating for an item greater than 3; 413 was arbitrary chosen so that also the first evens could show up in the top-5 recommendation list when we performed the evaluation): $5 \cdot \exp\left(\frac{-t}{810}\right) = t_{\text{min. rating}} = 3 \Rightarrow\ t \approx 413$. This is based on the assumption that the interest in photo galleries and articles of a past event decays as time moves on and recommendations should not stay in the top-5 recommendation list forever.

The second level consists of a traditional neighborhood-based collaborative filtering algorithm (CF). This is the level which incorporates the ratings created by filterbots. We employed the item-based algorithm as described by [19]. Instead of setting a fixed limit of ratings a user must have done before he gets recommendations, we employed a dynamic limit based on the "significance weighting factor" (also called "devalue") [7]. If an item has less than 5 items in its neighborhood, the similarity/correlation is multiplied by *n*/*devalue* (where *n* is the size of the neighborhood). By employing a *devalue* >= max. possible rating, it is possible to establish a minimum number of ratings of the current user indirectly. We chose 5 in our scenario. On the one hand this algorithm is able to generate personalized and cross-category recommendations (e.g. items in different categories, the user normally would not have looked at, [18]). On the other hand it has a major drawback. A user must have rated some articles (explicitly or implicitly) in the past in order to get (good) recommendations. However, recommendations should also be generated for new or more passive users. That is why the next two levels were included.

The third level uses alpha-community spaces (AC) combined with the level of agreement method [15]. Here, similar users (according to a property alpha) are grouped together (into an alpha-community) and within each group the level of agreement method is applied: For each item which was rated by at least one member of a group, the average rating within the group is calculated. Afterwards, a threshold is applied and items which are rated by at least by 20 per cent of the members of the group and exceed a minimum average rating are recommended. Instead of using a partition (pairwise disjoint groups) of the space, we relaxed this prerequisite of [15] in order to allow dynamic groups: In our approach, we applied this alpha-community spaces approach considering age of the user (interval of 6 years), age of associated children (interval of age of youngest child to the age of the oldest child), users living close to each other (<= 7 km distance), and visit of same events as alpha properties.

The fourth and last level uses the average rating (AVG) among all users as a prediction of ratings for articles. Of course, thresholds are applied in this case, too: Items have to be rated by at least nine users (called "agreement"). So, recommendations without personalization are possible for "anonymous" users. Recommendations based on average ratings are not optimal, however, "relevant recommendations without personalization are extremely useful for the vast number of anonymous user[s] as there is no user profile" [22, p. 249].

After the last level has been reached or after a level at least five recommendation candidates are available, a final weighting is performed. This is done in order to assign a higher weight to recommendations generated by more personalized approaches. Thus, our hybrid recommendation approach follows the "weighting" approach of the taxonomy of [4]. For an item this was done using a linear combination with weighting factors and the predicted ratings of the different recommenders. We chose a weighting of 1.1 for collaborative prediction, 1.05 for the alpha-

community spaces prediction and 1 for all others (based on our experience with test runs). Furthermore, semantics are taken into account to optimize the recommendation candidates: Not only the entered age of the children of the profile is used but also the system tries to infer the child ages the user has major interest in (based on reading behavior of age annotated articles): The idea behind this works as follows: If the users read more than 75 % of articles which are annotated with "ages up to 3", articles for "ages 4 to 6" are devalued (if the user does not have a child in the profile in this age interval) and vice versa. Hence, it is possible to react on varying circumstances in which parents forget or do not enter new children (e.g., the system detects if a parent whose profile only shows a 6-year-old child now also cares for content about new-borns and adapts its recommendations accordingly). The fact if a user attended an event or not did not have an impact on the recommendation of articles, since we wanted to show interesting articles to all users (statistics show that 40 % of the users who attended an event also read articles associated with unattended events and vice versa).

Using this hybrid approach, it is possible to generate recommendations for all types of users (more passive and more active users). The more ratings a user has made, the better and more personalized the recommendations get. Recommendations are presented on the start page, and they are included in e-mail campaigns. Single recommendations (one item) are also sent one-time via SMS text messages if the rating exceeds a predicted rating of 4.

3.3 Parameter optimization

The proposed algorithm is based on several parameters: First of all the threshold – the minimum predicted rating for items to be included into the final top-5 recommendation list –, the similarity metric, the significance weighting, the used alpha communities, the agreement parameter of the alpha communities, the number of minimum ratings for items to be considered in the average based recommendation, and the weight of each recommendation algorithm for the final rating. Apart from the final weighting factor (which is based on practical experience), all parameters have been determined by a parameter optimization based on cross-validations and measuring the recall (see section 4.1, [8]). We tried to optimize one parameter at a time. However, we found for the CF that the devalue parameter, the similarity metric and the inclusion/exclusion of items, which have just one rating of one user in common, were not unrelated. So we performed cross-validations for a set of sensible values. Exemplarily we want to show results of different metrics:

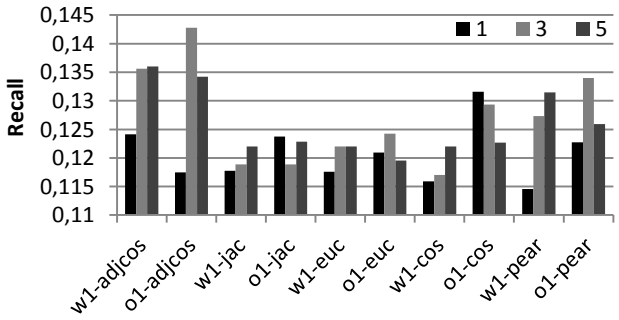


Figure 2. Recall based on different metrics and devalues

As Fig. 2 shows we evaluated different metrics as of: Adjusted cosine (adjcos), Jaccard based (jac, number of similar rated items divided by the number of co-rated items), Euclidean based (euc, 1/(distance + 1)), Cosine and Pearson correlation [19]. We performed cross-validations for each metric with three devalues (1, 3, and 5) and items which have exactly one co-rated item in common included (w1) and ignored (o1), as these are correlated and not independent. The adjusted cosine metric seems to work best, closely followed by the Pearson correlation and pure cosine metric. The peak of "o1-adjcos" with a devalue of 3 seems to be an anomaly (various test runs showed this). We chose the w1-adjcos with 5 as devalue parameter combination. In a manual evaluation we compared the top-5 recommendation lists for 10 % of our users and found that the "w1" variant produced recommendations with higher diversity (compared to the "o1" variant).

Fig. 3 shows the recall in the optimization process after each optimization and integration step. Remarkable is the recall of the plain CF approach (0.02) compared to the random selection of items (0.043). This result confirms the assumption that plain CF (and also alpha-community, Fig. 4) algorithms do not perform well in small scenarios. However, the combination of different algorithms and approaches performs better. We started our optimization by employing a plain CF algorithm and integrated the alpha-communities and average-based approaches (in two steps). Then we optimized the agreement for the average based algorithm, the similarity weight of the CF approach and the parameters of the alpha-communities. At last we integrated the age guessing (domain dependent optimization). With each optimization it was possible to raise the recall. The age guessing does not seem to have a high impact on the recall; however, its effects were clearly visible on the top-5 recommendation lists.

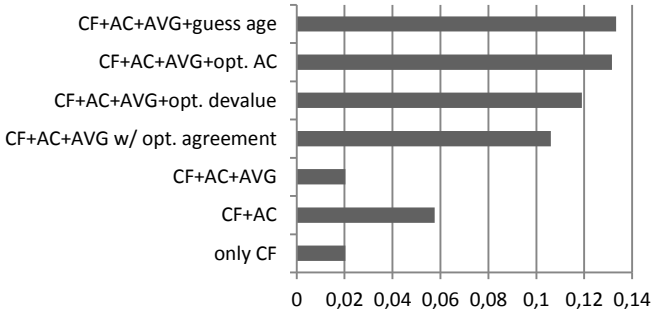


Figure 3. Recall improvements during parameter optimization

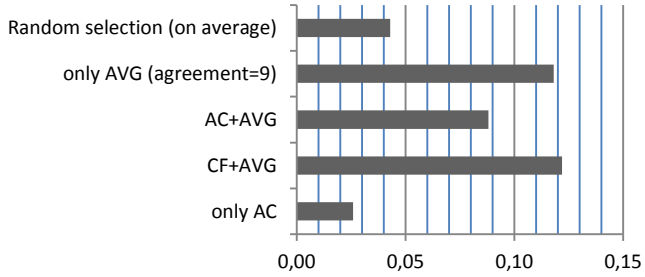


Figure 4. Recall of different recommender combinations

Fig. 4 also depicts the recall of different combinations of the recommendation algorithms. An examination shows that the CF and alpha-community approaches seem to be notably worse than the average-based approach or other combinations. Especially the average-based approach seems to stand out: We suppose the reason for this is our community in which a majority of users read and rated few articles.

4. EVALUATION

Evaluation of recommenders is not an easy task [8] especially for recommenders in smaller communities: A lot of recommenders were only evaluated using an offline evaluation with a metric and

cross-validations. This might be a viable option for bigger systems, but smaller systems suffer from a small number of ratings and items [22]. Also, an offline-evaluation cannot really measure user satisfaction [11, 20], but it is often used as a heuristic approximation for it. [11] found that a high prediction accuracy does not always correlate with high user satisfaction.

To face these problems, a multistage evaluation was conducted: An offline evaluation (cross-validation), an expert questionnaire, and a field study (online evaluation).

User satisfaction was central for us, even if it is hard and sophisticated to measure. In addition we also wanted to evaluate the quality of the recommendations as measured by domain experts.

Each stage evaluates different aspects, but also has characteristic weaknesses. There is no single method which combines all these aspects and is applicable to our target group. These three stages complement each other and give an overall view how the recommender performs as a whole. In the following sections we will describe the three evaluations and will present our results.

4.1 Results of our offline evaluation

The offline evaluation was conducted to measure the recommendation quality using a metric. The result is heavily dependent on the employed metric. Common metrics are Mean absolute error (MAE), Rank-based metrics, click-through rate CTR, and Recall/Precision [8, 22]. MAE is not optimal for rank based recommendations, since they do not measure where differences of predicted and actual ratings occur (top or bottom of recommendation list). Rank-based metrics (like Spearmans correlation or Kendalls Tau) compare the order of the recommendation list with an order based on user ratings. Additionally to the position problem, here partial order is also problematic. [22] has shown that CTR and user experience are not consistent. Overall, recall/precision seemed to fit best into our scenario.

For the offline evaluation 2,000 cross-validation runs were carried out. The cross-validations were measured as described in [8]: Rated items by a user were randomly divided into two sets (test set and training set). The test set contained 20 per cent of the rated items (afterwards the test set was cleaned up so that only "relevant" items were in it, i.e. items with rating >= 3). Recall and precision were measured as follows: The recommendation system was initialized and executed on the training set and then recall and precision were measured. Recall is the number of recommended relevant items (i.e. items which are recommended and in the test set) divided by the number of items in the test set and precision is the number of relevant recommended items divided by the number of recommended items (5 here). Recall and precision were not independent on our small data sets (the test-set contained at most 5 items): both are linearly dependent. So it is irrelevant which one or which combination is selected; especially if the number of recommended items is fixed (5 in our scenario). As such, only recall was taken as a measure. The goal of this measurement was to compare the average recall of the proposed approach to the average recall of a random selection of items. We found that the algorithm (recall: 0.13) outperformed a random recommendation (recall: 0.04).

Moreover, we measured the recall for three classes of users, based on their read-behavior: "read < 5articles" (47 % of user base), "read > 20 articles" (14 % of user base) and users between these two (39 % of user base). Again all recalls (see Fig. 5) were higher than the random selection of items. Fig. 5 shows that the recall for people who read few articles (< 5) is very high (approx. twice as high as for users who read many articles) and that the recall for users who read many items (0.1) is below the overall recall of our prototype (0.13). A possible explanation for the first is that the test set for a lot of users of this class is empty (hence, these are ignored in the calculation) and a lot of users reflect the majority opinion. For users who read more articles, a reason for this result might be that it is hard or even impossible to measure relevance of recommended articles, which are not in the test set. Based on this initial evidence of system success, we went to the second stage of evaluation.

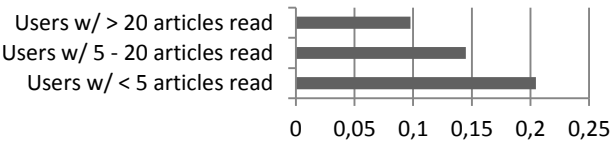


Figure 5. Recall for three different classes of users based on the number of articles read

4.2 Results of our expert questionnaire

In the second stage, an expert questionnaire was conducted to have four pedagogic experts evaluate the quality of the recommendations regarding pedagogic aspects. The experts were the pedagogues of the Mobile2Learn team. These knew the articles and the potential target group and could assess the recommendations best.

Twelve user profiles were taken randomly. However, the selection of profiles should be representative for the Mobile2Learn community. Therefore, we picked user profiles from the three classes mentioned before according to their occurrences in the community. Each profile showed anonymized demographic information about a user (residence, age, gender and children), attended events, all visited pages, visit durations (as "short", < 15 seconds; "mid"; "long", > 60 seconds) and explicit article ratings (if available). Also, for each profile, three different TOP-5 recommendation lists were shown. Two of these were generated by random selection of (unread) items while one was the output of the proposed recommender system. It was the design of the study, that the probability to choose a random selection is as twice as high as the calculated recommendation. The experts were asked to pick of the best of these three recommendation lists based on their own experience and the given user profiles.

The (normalized) results of this study show that three of the experts clearly preferred the recommendations of the algorithm. Overall the random recommendation lists where selected 18 times and the recommendation lists generated by our algorithm 30 times – i.e. each expert selected 7.5 generated recommendation lists on average. Two experts selected the generated recommendations more often than the random ones (11:1 and 9:3), one expert chose the two lists equally often (6:6) and the last expert chose 4 more random recommendations than the generated ones (4:8). Because of our study design, these numbers need to be normalized: This means that the four experts chose the generated recommendation lists to 96 %, 86 %, 67 % and 50 %. The reason for the 50 % of the last expert was that he had different opinions about when galleries should be recommended (only if a user has attended an event or also if the user did not attend any event to advertise them). Overall, for each profile at least one expert chose the recommendation of our algorithm and only for one profile three experts chose the same random generated recommendation list.

Biggest common sense of the experts with the calculated recommendations was for users who did not read many articles. Here, in four of six profiles all experts chose the recommendation

of our algorithm. The reason for this might be the small sample of profiles with mid to many read articles (> 5 articles) and the recommendations itself: Especially for users with a detailed profile (many read articles) it is difficult also for humans to interpret the data and imagine/assess which articles/recommendation list might fit best.

Concluding there seems to be a clear tendency for preference of the experts for the recommendations generated by our proposed algorithm.

4.3 Results of our field study

The last part of the evaluation was a field study in order to measure user satisfaction. When the prototype was put productive and a user clicked on a recommendation, we displayed a small feedback box on the recommended page asking if the page is interesting for the user or not. The binary answer, together with the type of recommendation and the rank of the top-5-recommendation list, were stored in a database. Within one month, 43 recommendations were clicked on by 28 unique users. We sent two newsletters to 189 and 199 users and sent 14 recommendations via SMS. We got feedback for 20 recommendations by 17 unique users. All feedback was purely positive. The feedback we got covers all kinds of item types (events, galleries and article) and recommendation methods (rule-, average-, alpha-community- and collaborative filtering-based). Even if the number of 17 users sounds very small, this is approximately 10 per cent of the user base and the same number as the number of "power users" in the community.

18 clicks (42 %) on recommendations came from our newsletters. We cannot definitely say how many users accepted recommendations from SMS messages. However, two users visited the Mobile2Learn website and clicked on the same recommendation which was sent via SMS a few minutes ago.

We cannot know if users who did not give any feedback did not want to give feedback, did not notice the feedback box or if the recommended page was interesting for them or not. Anyway, we analyzed the log files and found out, that even if users did not give any feedback, they often visited a recommended gallery extensively or explored related articles (to the one which was recommended). We suppose that this is a sign for good recommendations, since our users were directed to a right direction which was interesting for them.

Additionally, we observed that the average number of total read articles per user increased from 8.9 (before the prototype was activated) to 9.2 (one month later), despite the fact that ten new users registered in the same time span.

5. DISCUSSION

In this section we present our experiences with explicit ratings and the click behavior of our users. Then we discuss other possible filterbot implementations and the recall evaluation metric we used for our parameter optimization and evaluation. Finally we point out further research tasks for recommendation systems in smaller communities and transferability to other scenarios.

The recommendation algorithm supports explicit ratings, however, this whole recommendation process is mainly based upon implicit ratings (which were generated out of visit times): At the design time, there were only very few explicit ratings (even a lottery for ratings did not encourage people to submit much more ratings). Also in the evaluation period, no noteworthy amount of new ratings was made.

In the evaluation period, we analyzed which ranks the items of the top-5 recommendation had which were clicked on: Clicks on items were not limited to the first recommended items of the top-5 ranking – but we found a clear preference for the first recommended items: 40 % of all clicks were on the first item, 60 % of all clicks were on the first two items and approx. 70 % on the first three items. This means that users did not just click on the first recommended item, but choose on which to click.

We also investigated different implementations of Filterbots in our scenario (see Fig. 6): Three different variants were implemented and tested. Each implementation differs in how similarities are calculated. The first variant is the implementation we described in section 3.1. On closer examination one can see a possible problem: The idea of the design was that the similarity metric reflects the distance in the article tree (see Fig. 1). $s(r_a,r_b) = s(r_a,r_e) = 1/2$ and $s(r_b,r_e) = 1/3$ should hold, but using the adjusted cosine similarity metric (with absence of other ratings) $s(r_a,r_b) = s(r_a,r_e) = s(r_b,r_e) = 1$ holds. Hence, we developed two other filterbot variants, which calculate the similarity of the filterbot ratings and the user ratings separately. Both variants (2 and 3) use the jaccard coefficient to calculate the similarity of the full filterbot rating vectors. For this metric the distance of articles in the tree as described before holds (i.e. $s(r_a,r_b) = s(r_a,r_e) = 1/2$). The difference of variant 2 and 3 lies in the way they combine the filterbot rating similarity with the user ratings similarity: Variant 1 uses the unweighted arithmetic mean. Variant 2 uses a weighted arithmetic mean where the user ratings count twice as high as the filterbot rating similarity. Fig. 6 shows that variant 1 seems to work best in our scenario and that the filterbot rating integration does not corrupt the quality of the recommendations (compared by recall). A possible explanation why variant 2 and 3 are worse than variant 1 might be a weight bias: Variant 2 and variant 3 use a ratio of 1:1 and 1:2 of filterbot rating similarity to user rating similarity – this might be a too high weight for the filterbot ratings. This opens room for further investigations with dynamic weights (based on the number of user ratings).

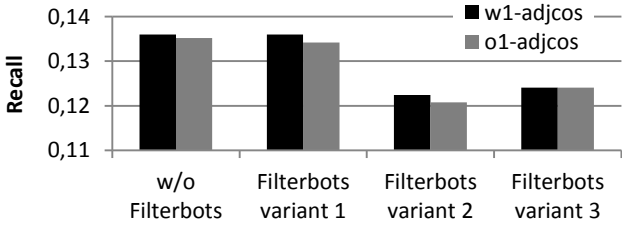


Figure 6. Recall of different Filterbots implementations

The recall metric restricts itself to measure if already good rated items from the test set are recommended based on the items of the training set. So, there is no statement about the quality of items which are recommended, but which are not in the test set. In addition, it is not obvious if a user notices small improvements of the accuracy or if this increases the user satisfaction. Certainly this depends of the concrete situation: When we integrated the age-detection into the final ranking, the recall just improved very slightly, but the impact on the recommended items was clearly visible during test-runs. All in all we increased the quality based on the recall of the recommendations from 0.02 up to 0.13 (factor 6.5) and got positive feedback from both, the domain experts (pedagogues) and users of our system.

Instead of just focusing on tagging the bag-of-words approach combined with TF/IDF weighting could also be considered for classifying articles or determining article similarities. However,

this only works for text-based items and cannot be used for articles, which just contain an image, or other item types (like photo galleries or music).

A smaller number of items might lead to decreasing recommendation quality for users who read/rated many items over time. For these users the number of unrated items is even smaller or shrinks and items with higher predictions are recommended first. So, items with a less high prediction are recommended. However, items must have a minimum predicted rating of 3. We have not experienced this so far, but here further research is possible.

One possible problem with CF and also with this approach is diversification: it might happen, that two or more items of the top-5 recommendation are closely related. Further steps might be needed to evaluate and improve diversification.

By applying a multilayered approach, this hybrid recommender algorithm is (or parts of it are) also applicable to bigger or growing systems. If the amount of data grows, the CF layer will generate a full top-5 recommendation without the need to proceed to (possibly worse performing) deeper layers. However, deeper layers can still be reached for new users for which a less amount data is available. Independently, filterbots, alpha-communities, and rule based recommendations can be used to enhance existing recommender systems – these just have to be adapted to the needs of the environment. Of course for scenarios with a bigger amount of data optimizations for speed and performance are becoming more important.

6. SUMMARY

We propose a new recommender approach which was designed for smaller, regionally bound communities. Apart from only using traditional recommendation techniques, it makes use of different kinds of context and semantics: The recommender infers content types which the user is interested in (refers to family context) and also used data about visited "real world" events. This context-awareness and domain specific semantics turned out to be critical for developing the recommendation system in our scenario, a local parent community. We implemented our proposed approach as a prototype and performed a multi-staged evaluation (offline and online). Each stage of the evaluation showed a clear tendency that our approach and recommendation systems can work – even in small scenarios.

Even if the set of special properties sounds special (e.g. different item types, combination of online/offline items), these are quite common for smaller communities. So, the approach presented in the paper is transferable to different smaller communities, but is not bound to that. Aspects of it can also be used to improve the quality of recommendation in larger systems.

7. REFERENCES

[1] Bauer, U. and Bittlingmayer, U. H. 2005. Wer profitiert von Elternbildung? In *Zeitschrift für Soziologie der Erziehung und Sozialisation 25* (3). 263–280.

[2] Büchner, P. 2003. Stichwort: Bildung und soziale Ungleichheit. In *Zeitschrift für Erziehungswissenschaft 6* (1). 5–24.

[3] Burke, R. 2000. Knowledge-based recommender systems. In: Dekker, M. ed. *Encyclopedia of Library and Information Systems 69*. New York, NY, USA. 180–200.

[4] Burke, R. 2007. Hybrid Web Recommender Systems. In Brusilovsky, P., Kobsa, A., Nejdl, W. eds. *The Adaptive Web, LNCS 4321*. Springer Berlin/Heidelberg, 377–408.

[5] Davis-Kean, P. E. 2005. The Influence of Parent Education and Family Income on Child Achievement: The Indirect Role of Parental Expectations and the Home Environment. In *Journal of Family Psychology 19* (2). 294–304.

[6] Ganesan, P., Garcia-Molina, H., and Widom, J. 2003. Exploiting hierarchical domain structure to compute similarity. In: *ACM Trans. Inf. Syst. 21* (1). 64–93.

[7] Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. 1999. An algorithmic framework for performing collaborative filtering. In *Proc. ACM SIGIR '99*. 230–237.

[8] Herlocker, J. L., Konstan J. A., Terveen, T., and Riedl, J. 2004. Evaluating collaborative filtering recommender systems. In *ACM TOIS '04 22* (1). 5–53.

[9] Hu, Y., Koren, Y., and Volinsky, C. 2008. Collaborative filtering for implicit feedback datasets. In *ICDM'08*. 263–272

[10] Li, Q. and Kim, B. M. 2003. An approach for combining content-based and collaborative filters. In: *Proc. AsianIR'03 11*. Stroudsburg, PA, USA. ACL. 17–24

[11] McNee, S. M., Riedl, J., and Konstan, J. A. 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *ACM CHI EA '06*. 1097–1101

[12] Melville, P., Mooney, R. J., and Nagarajan, R. 2002. Content-boosted collaborative filtering for improved recommendations. In *Proc. of AAAI'02*. 187–192.

[13] Morita, M. and Shinoda, Y. 1994. Information filtering based on user behavior analysis and best match text retrieval. In *ACM SIGIR '94*. 272–281.

[14] Moshfeghi, Y., Agarwal, D., Piwowarski, B., and Jose, J. M. 2009. Movie Recommender: Semantically Enriched Unified Relevance Model for Rating Prediction in Collaborative Filtering. In *ECIR'09 LNCS 5478.* 54–65.

[15] Nguyen, A.-T., Denos, N., and Berrut, C. 2007. Improving new user recommendations with rule-based induction on cold user data. In *Proc. of ACM RecSys'07*. 121–128.

[16] Park, S.-T., Pennock, D., Madani, O., Good, N., and Decoste, D. 2006. Naïve filterbots for robust cold-start recommendations. In *ACM SIGKDD'06*, 699–705.

[17] Pazzani, M. J. and Billsus, D. 2007. Content-based recommendation systems. In Brusilovsky, P., Kobsa, A., and Nejdl, W. eds. *The adaptive web. LNCS 4321*. Springer-Verlag, Berlin, Heidelberg. 325–341.

[18] Sarwar, B., Konstann, J. A., Borchers, A., Herlocker, J., Miller, B., Riedl, J. 1998. Using filtering agents to improve prediction quality in the GroupLens research collaborative filtering system. In *Proc. of CSCW'98*. 345–354.

[19] Sarwar, B., Karypis, G., Konstan, J. A., and Riedl. J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proc. of ACM WWW'01*. 285–295.

[20] Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. 2007. The adaptive web. Springer Berlin/Heidelberg. 291-324

[21] Strickroth, S., Pinkwart, N., and Müller, J. P. 2011. Neue Medien und Präsenzveranstaltungen: Ein didaktisches Modell für die Elternbildung? In Friedrich, S., et al. eds. *DeLFI'11*. Dresden, Germany. TUDpress.

[22] Zheng, H., Wang, D., Zhang, Q., Li, H., and Yang, T. 2010. Do clicks measure recommendation relevancy?: an empirical user study. *ACM RecSys '10*. New York, NY, USA. 249–252.