Proceedings WEBKDD'2002, Eds. B. Masand, M. Spiliopoulou, J. Srivastava, O. Zaïane, Edmonton, Canada, July 2002, pages 100–114

Evaluation of Recommender Algorithms for an Internet Information Broker based on Simple Association Rules and on the Repeat-Buying Theory

Andreas Geyer-Schulz[1] and Michael Hahsler[2]

[1] Universität Karlsruhe (TH), D-76128 Karlsruhe, Germany,
andreas.geyer-schulz@em.uni-karlsruhe.de,
[2] Wirtschaftsuniversität Wien, A-1090 Wien, Austria,
Michael.Hahsler@wu-wien.ac.at

Abstract. Association rules are a widely used technique to generate recommendations in commercial and research recommender systems. Since more and more Web sites, especially of retailers, offer automatic recommender services using Web usage mining, evaluation of recommender algorithms becomes increasingly important. In this paper we first present a framework for the evaluation of different aspects of recommender systems based on the process of discovering knowledge in databases of Fayyad et al. and then we focus on the comparison of the performance of two recommender algorithms based on frequent itemsets. The first recommender algorithm uses association rules, and the other recommender algorithm is based on the repeat-buying theory known from marketing research. For the evaluation we concentrated on how well the patterns extracted from usage data match the concept of *useful recommendations* of users. We use 6 month of usage data from an educational Internet information broker and compare useful recommendations identified by users from the target group of the broker with the results of the recommender algorithms. The results of the evaluation presented in this paper suggest that frequent itemsets from purchase histories match the concept of *useful recommendations* expressed by users with satisfactory accuracy (higher than 70%) and precision (between 60% and 90%). Also the evaluation suggests that both algorithms studied in the paper perform similar on real-world data if they are tuned properly.

1 Introduction

Since recommender systems are becoming widely used by retailer Web sites (e.g. Amazon.com, Barnes & Noble.com), a careful evaluation of their performance gets increasingly important. However, recommender systems are complex applications that are based on a combination of several models (mathematical and psychological), algorithms and heuristics. This complexity makes evaluation efforts very difficult and results are hardly generalizable, which is apparent in the literature about recommender evaluation. In this paper we try to improve recommender evaluation by developing a more systematic approach in respect of what actually is evaluated. For this purpose we develop in section 2 a framework for the systematic evaluation of recommender systems which is in principle based on the process of knowledge discovery in databases by Fayyad et al. [10]. The expected benefit of this is that the evaluation of various aspects of a recommender system can be separated and thus more properly targeted for the different stakeholders responsible for the introduction of such systems and therefore is more suitable for continuous process improvement by focusing on the most promising areas of improvement. In section 3 we review common performance measures for recommender system evaluation and how they are used in the recommender systems literature.

For the rest of the paper, we focus on one specific aspect of the knowledge discovery process in databases, namely the comparison of data mining methods for the generation of recommendations. There are many possible sources for generating recommendations including expressed preferences, opinions of experts, characteristics of people and items, and many more. For recommendation services the information of all available sources should be combined to provide the best recommendations possible. However, for the performance evaluation in this paper our aim is not to present a complete recommender system, but we ask the often neglected question of how well the patterns extracted from usage data (frequent itemsets) match the concept of *useful recommendations* given by a human expert. Hence we restrict our investigation to mining only usage data. In sections 4 and 5 we give a brief introduction to the two data mining algorithms used in the paper. The first uses the in the KDD society well-known support-confidence framework, the second algorithm is based on Ehrenberg's repeat-buying theory originating from marketing research. In section 6 we describe the experimental setup and the data set. In section 7 we present and discuss the evaluation results. We conclude with a short summary of the findings and open research questions in section 8.

2 A Framework for the Evaluation of Recommender Systems

Recommender systems are complex applications which have to perform the whole process of knowledge discovery in databases (KDD). In [10] Fayyad et al. give an overview of the steps of the KDD process. An application of this model to recommender systems is straightforward. However, in order to separate the influence of the user interface from the effects of the choice of a data mining method we add *presentation* as additional step to the process model of Fayyad et al. (see figure 1). The five major steps are:

1. *Selection:* The data set used to produce recommendations can stem from various sources. For example these sources can be already existing transaction logs (e.g. point of sale data, Web server logs) or the data can be collected specially for the purpose of generating recommendations (e.g. ratings for movies).
2. *Preprocessing and transformation:* In these steps the data set is cleaned from noise, inconsistent data is removed and missing data is inferred. After this treatment the cleaned data is transformed into a representation suitable for data mining.

   For example, for collaborative filtering the data normally consists of explicit ratings by users which are collected for the purpose of creating recommendations (e.g. for music see [23]). Preparation mainly involves discovering and removing inconsistent ratings.

   For Web usage mining (see [24, 17]) data is collected by observing the behavior of users browsing a Web site. Since observation, especially server-side observation on the Web, is far from perfect, much effort has to be put on data preprocessing, cleaning and transformation. Problems involve the identification of users, dynamic (rotating) IP-addressing, session identification, missing data due to proxy servers and system crashes, requests by Web robots and many more.
3. *Data mining:* The objective of this step is to find interesting patterns in the data set that are useful for recommendation purposes. The output of data mining in recommender systems can be: groups of users with similar interests, items that are frequently used together, often used sequences of items,... Frequently, extracting patterns means learning the parameters of a specified model from the data set.
4. *Interpretation and evaluation:* In order to build knowledge, the found patterns (the model and its parameters) have to be understandable to humans. Only with this property the process

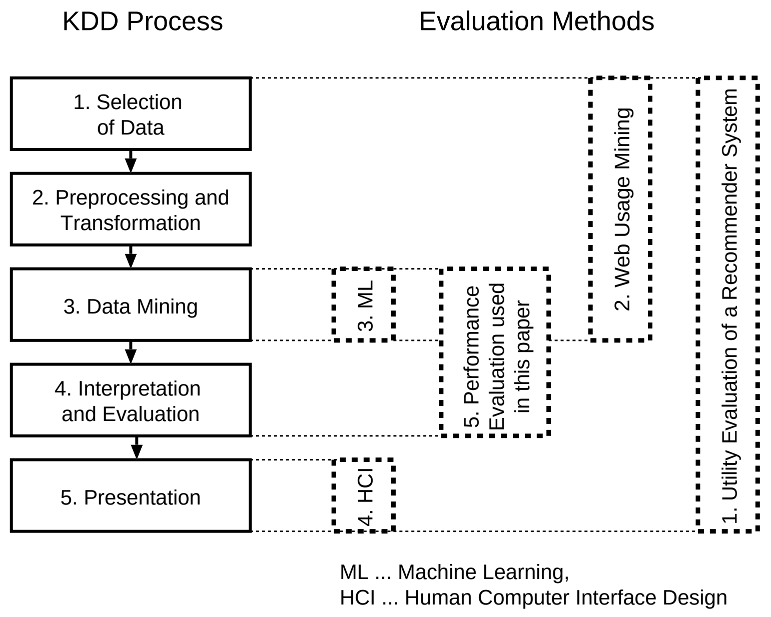
KDD Process
Evaluation Methods
1. Selection of Data
2. Preprocessing and Transformation
3. Data Mining
4. Interpretation and Evaluation
5. Presentation
3. ML
5. Performance Evaluation used in this paper
2. Web Usage Mining
1. Utility Evaluation of a Recommender System
4. HCI
ML ... Machine Learning,
HCI ... Human Computer Interface Design

Fig. 1. Mapping evaluation methods to the steps of the KDD process

can be called knowledge discovery and the results can be interpreted. A recommender system interprets found patterns for the user.
Finally the validity (patterns are also valid for new data), novelty (involves a new way of finding patterns), usefulness (potentially lead to useful action) and understandability (build and increase knowledge) of the patterns needs to be evaluated.

5. *Presentation:* A recommender system presents this interpretation in a suitable form as a recommendation. For example, the recommendation can be a top-n list of recommended items for a user, or a list of items that are similar to an item the user likes, or it can consist of information about how other users with similar interests rated a specific item.

Because of the complexity of recommender systems, and the many available choices for each step of the knowledge discovery process described above, detailed evaluation of the different aspects of this process is a necessity. In figure 1 we mapped five evaluation methods for recommender systems to the steps of the KDD process to provide a systematic reference framework. We summarize these evaluation methods in the following:

1. The evaluation of the utility of a recommender system is targeted to the stakeholders of the organization and evaluates the utility of the whole process as shown in figure 1. In practice the utility of the whole recommender system is usually evaluated by the impact of the recommender system on the overall performance of the process the recommender supports. If the process supported is the purchase process in a supermarket, the impact is measured by comparing sales of the product with and without recommender in two test markets. This approach is used in Lawrence et al. [15] for the evaluation of a personalized product recommender on personal digital assistants (PDAs) for the Safeway supermarket chain. The performance measure used for evaluation is the revenue from the sales volume triggered by the recommender. Lawrence et al. reported that they observed an increase in revenue of about 1.8% after introducing the recommender to a new store. This is in line with NetPerception's estimation of a 2

percent increase in sales volume [20]. NetPerception measures this by comparing the impact of random product lists versus recommended product lists on the buying behavior of the consumer. For non-profit sites, such an overall performance measure could be a conversion rate, a contact rate, a task achievement rate, ...

2. Web usage mining covers the first three steps of the KDD process. Therefore, in addition to the evaluation of the data mining method evaluation of the data selection and preprocessing steps is important for *Web usage mining* ([24, 17]). Correctly observing user behavior on the Web requires automatically handling difficulties e.g. with filtering automatic Web robots [25], detecting sessions [7], path completion [8], ... For example, for evaluating session identification heuristics Cooley recommends comparing the results of log files with session identification with the results of analyzing the same log files stripped (see [8, pp. 118-123]). Evaluation of path-completion heuristics requires instrumented Web browsers which log user interactions, so that the data collected with instrumented Web browsers can be compared to the results of path-completion algorithms applied to server logs. Another, last resort, evaluation method is testing preprocessing statistics on synthetic data. For episode identification this approach has been used by Cooley [8]. An expensive alternative are experiments in usability labs where the behavior of users is e.g. video-taped and then manually analyzed.

3. Most researchers in recommender systems focus the evaluation of mining algorithms with methods known from *machine learning.* A common way from machine learning for comparing the performance of several algorithms is to use a prepared data set and divide it into a set for training and a set for evaluation (cross-validation is often used). This approach only takes into account how well patterns in the data set can be learned and not how useful the patterns are for recommendation purposes. However, in addition, the underlying theoretical framework of these algorithms must be evaluated with respect to its consistency. For association rule algorithms, a recent discussion can be found in Adamo [1, pp. 151-184]. Furthermore, for model-based approaches the correspondence of the model with reality must be evaluated. For statistical models this is done by testing the underlying behavioral assumptions in a piece-meal fashion, by diagnostic-checking. For the repeat-buying theory used in this paper, see e.g. [9]. Model comparison is performed along several dimensions, most notably performance, robustness, parsimonity of parameters, sensitivity to misspecification of parameters, computational complexity, ease of use and understandability.

4. Evaluation of the presentation of recommendations to the consumer/user is dealt with in the area of *human-computer interface (HCI)* research and includes methods such as usability labs and field-experiments. In connection with recommender systems Herlocker et al. [13] compared 21 different representations of recommendations for movies. The findings were, that, similar to previous works with expert systems, suitable explanation of the recommendations to the users increases the acceptance of the recommender system.

5. In this paper we focus on the evaluation of the performance of a simple association rule algorithm and a novel repeat-buying based algorithm. Our main research question is how well the patterns extracted from usage data by these algorithms match the concept of *useful recommendations* for users. We ask the question whether recommendations by algorithms based on observed frequency of co-purchases are similar to recommendations given by humans. Therefore, the performance evaluation presented in this paper combines the data mining step as well as part of the interpretation step of the KDD process shown in figure 1 in the sense that the performance of the data mining algorithms are evaluated with regard how well their output corresponds to a concept, namely the concept of *useful recommendations*.

actual / predicted	**negative**	**positive**
negative	*a*	*b*
positive	*c*	*d*

Table 1. 2x2 confusion matrix

3 Performance measures for Recommendation Algorithms

For measuring the performance of recommender algorithms measures originating from statistics, machine learning and information retrieval are used. To give definitions of the performance measures we first have to define the meaning of the terms item, recommendation and recommendation list. *Items* are the products under consideration that can be purchased (consumed/used) by the customer (user). For an item a *recommendation list* is a list of other items, that are recommended to be used together with the first item. A single *recommendation* is the relationship between two items, where the latter item is recommended to be used together with the first one. Therefore, a recommendation list consists of all recommendations for a specific item.

Since all measures use similar information, it is useful to lead them back to the so called *confusion matrix* depicted in table 1 (see [14]), which corresponds exactly to the outcomes of a classical statistical experiment. The confusion matrix shows how many of the possible recommendations were predicted as recommendations by the algorithm (column predicted positive) and how many of those actually were correct recommendations (cell *d*) and how many not (cell *b*). The matrix also shows how many of the possible recommendations the algorithm rejected (column predicted negative), were correctly rejected (cell *a*) or should have actually been recommended (cell *c*). Statistically cell *c* is known as type I error with probability α and cell *b* is known as type II error with probability β.

Performance measures from machine learning. For the data mining task of a recommender system the performance of an algorithm depends on its ability to learn significant patterns in the data set. Performance measures used to evaluate these algorithms have their root in machine learning.

Commonly used measures are accuracy and coverage. *Accuracy* is the fraction of correct recommendations to total possible recommendations (see formula 1). *Coverage* measures the fraction of items the system is able to provide recommendations for (see formula 2). We can not define coverage directly from the confusion matrix, since the confusion matrix only represents information at the level of recommendations (relationships between items) and not at the level of individual items with recommendation lists.

$$Accuracy = \frac{correct\ recommendations}{total\ possible\ recommendations} = \frac{a+d}{a+b+c+d} \tag{1}$$

$$Coverage = \frac{items\ with\ recommendations}{total\ number\ of\ items} \tag{2}$$

Another common measure is the *mean absolute error (MAE)* shown in formula 3. N is the length of the learned pattern from the training set (total number of items we produce recommendations for = items with recommendations in formula 2) and $|\varepsilon_i|$ is the absolute error of each component (number of incorrect classifications in the recommendation list for each item) of the pattern compared to the evaluation set.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\varepsilon_i| = \frac{b+c}{N} \tag{3}$$

Performance measures from information retrieval. Recommender systems help to find items of interest from the set of all available items. This can be seen as a retrieval task known from information retrieval. Therefore, standard information retrieval performance measures are frequently used to evaluate recommender performance.

Precision and *recall* are the best known measures used in information retrieval [21, 26] (see formula 4 and 5 for the definitions).

$$Precision = \frac{correctly\ recommended\ items}{total\ recommended\ items} = \frac{d}{b+d} \tag{4}$$

$$Recall = \frac{correctly\ recommended\ items}{total\ useful\ recommendations} = \frac{d}{c+d} \tag{5}$$

Often the number of *total useful recommendations* needed for recall is unknown since the whole collection would have to be inspected. However, instead of the actual *total useful recommendations* often the total number of known useful recommendations is used as an estimate.

Precision and recall are conflicting properties, high precision means low recall and vice versa. To find an optimal trade-off between precision and recall a single-valued measure like the *F-measure* can be used. The F-measure is defined as the harmonic mean of precision and recall given in formula 6. In the recommender evaluation literature the F-measure is often referred to as F1 measure.

$$F-measure = \frac{2*Precision*Recall}{Precision+Recall} = \frac{2}{1/Precision+1/Recall} \tag{6}$$

The F-measure is a special case of the *E-measure* which places the same weight on both, precision and recall [21]. The E-measure is defined in formula 7. The parameter α controls the trade-off between precision and recall.

$$E-measure = \frac{1}{\alpha(1/Precision)+(1-\alpha)(1/Recall)} \tag{7}$$

Other performance measures used for recommender systems. Another measure used in the literature is the *Receiver Operating Characteristic* (ROC). It is a measure used in signal detection and goes back to the Swets model [26]. The ROC-curve is a plot of the systems *sensitivity* (probability of detection, true positive rate) by its $1-specifity$ (probability of false alarm, $1-$ true negative rate). A possible way to compare the efficiency of two systems is by comparing the size of the area under the ROC-curve, where a bigger area indicates better performance.

Mobasher et al. introduced in [19] a new measure called *R*. It is obtained by dividing the coverage by the size of the recommendation set. Therefore it favors smaller recommendation sets.

In table 3 we summarize recent papers that evaluated recommender algorithms using the presented measures.

4 Recommender using Association Rules

The first recommender we use is based on association rules with thresholds for minimum support and minimum confidence. This is known as support-confidence framework. The problem of finding association rules for market basket data that satisfy minimum support and minimum confidence was first presented by Agrawal et al. [3]. Association rule algorithms require no model

Paper	Domain	Algorithm	Measures
Shardanand and Maes [23]	Music rating	Prediction algorithms based on similarity between user profiles	MAE
Herlocker et al. [12]	Movies rating	Neighborhood based prediction algorithms	MAE, ROC, Coverage
Sarwar et al. [22]	Movies rating, E-Commerce purchases	Dimensionality reduction	MAE, F-measure
Mobasher et al. [18]	Web-Site usage	Aggregate user profiles (clustering user transactions and pageviews)	Accuracy
Vucetic and Obradovic [27]	Movies rating	Regression based Item-to-item relationship	MAE, ROC
Yu et al. [28]	Movies rating	Instance selection for collaborative filtering	MAE, Accuracy
Mobasher et al. [19]	Web-Site usage	Aggregate user profiles (clustering user transactions and pageviews)	Precision, Coverage, F-measure, R
Lin et al. [16]	Movies rating	Adaptive-support association rule mining	Accuracy, Precision, Recall

Table 2. Recommender algorithm evaluation papers

assumptions. This makes the approach very attractive and simple, because checking whether the model assumptions are met is not required.

The problem is formalized in [3] as follows: Let $I = \{i_1, i_2, ..., i_m\}$ be a set of items. Let D be a set of transactions, where each T is a set of items such that $T \subseteq I$. A transaction T contains X if $X \subseteq T$ and Y if $Y \subseteq T$. An association rule is an implication in the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = 0$. X is called the antecedent and Y is called the consequent of the rule.

The rule $X \Rightarrow Y$ has support sup in the transaction set D if sup% of transactions in D contain X and Y. The rule $X \Rightarrow Y$ holds in the transaction set D with confidence $conf$ if $conf$% of transactions in D that contain X also contain Y. In formula 8 and formula 9 support and confidence are defined by probabilities.

$$sup(X \Rightarrow Y) = sup(Y \Rightarrow X) = P(X \wedge Y) \tag{8}$$

$$conf(X \Rightarrow Y) = \frac{P(X \wedge Y)}{P(X)} \tag{9}$$

Agrawal and Srikant presented in [4] the APRIORI algorithm to efficiently compute association rules with several items in the antecedent of the rule. With every pass the algorithm adds one item to the so-called large itemsets and prunes away the itemsets without sufficient support. The algorithm stops when no larger itemset can be produced without falling under minimum support. Then all large itemsets are used to produce the association rules.

For the simple recommender system used in this paper we only need association rules with one single item in the antecedent of the rule. Then for the item in the antecedent of each rule we use the items in the consequent as recommendations. The number of recommendations produced can be varied by changing the support and confidence thresholds.

Recently support and confidence were subject to criticism [6, 2]. The main point of criticism is that confidence ignores the frequency of the consequent in the data set and therefore spurious rules with items in the consequent that appear often in the data set cannot be separated from more accurate rules. *Lift* [5], also known as *interest* [6, 2] is an alternative measure for confidence that takes the frequency of the consequent into account. But instead of the implication it measures only the co-occurrence of X and Y as a symmetric measure. Lift is defined as the deviation of the co-occurrences of two items from the assumption that they occur independently. The definition of lift for the rules $X \Rightarrow Y$ and $Y \Rightarrow X$ is given in formula 10.

$$lift(X \Rightarrow Y) = lift(Y \Rightarrow X) = \frac{P(X \wedge Y)}{P(X)P(Y)} = \frac{conf(Y \Rightarrow X)}{sup(X)} \tag{10}$$

Another alternative measure for confidence is *conviction*. Conviction [6, 5] measures the deviation of the implication $X \Rightarrow Y$ from the assumption that X and Y occur independently. It is derived from the fact, that the implication $X \Rightarrow Y$ can logically be reformulated as $\neg(X \wedge \neg Y)$. If we divide this by the individual probabilities $P(X)$ and $P(\neg Y)$ and invert the ratio to compensate for the outside negation we reach the definition of conviction as given in formula 11.

$$conviction(X \Rightarrow Y) = \frac{P(X)P(\neg Y)}{P(X \wedge \neg Y)} = \frac{1 - sup(Y)}{1 - conf(X \Rightarrow Y)} \tag{11}$$

Although in the literature conviction is said to be superior to confidence [6] most papers still use the support-confidence framework. For an extensive survey of variants of association rule algorithms and a discussion of their intrinsic short-comings, see Adamo [1]. We will report results for confidence, lift and conviction in this paper.

Association rule algorithms require the specification of two parameters, namely support and confidence (or lift or conviction). The computational complexity is in general exponential. For simple association rules used in this paper, the computational complexity is of quadratic order. Production recommender systems in organizations require periodical updates. For association rule algorithms, we are currently not aware of association rule algorithms designed for incremental updates. Therefore, the update of the confidence and support of association rules requires a complete update of all support and confidence values which is of quadratic order in the number of items.

5 Recommender using the Repeat-buying Theory

The second recommender algorithm is based on the repeat-buying theory introduced by Ehrenberg [9]. In the context of this paper, buying an item means to use it. The idea behind the repeat-buying theory is, that in a stationary situation with the purchases of all items being independent from each other, the buying behavior follows a process that produces a specific distribution for the number of repeat buys, namely the negative binomial distribution (NBD). The statistical model is based on several strong behavioral assumptions about consumer purchase behavior which invite criticism of the model [9]. Ehrenberg and other authors empirically showed that this model holds for various consumer markets.

In [11] we showed how to apply a simplified form of the model (using the logarithmic series distribution (LSD), a special case of the NBD also described in [9]) to generate recommendations for the Internet information broker used in this paper. In this setting the LSD in formula 12 gives the probability of the observation that the same pair of two independent items are used together in a specified period of time a total of 1, 2, 3, ..., r times by pure chance. q is the only parameter of the distribution and can be easily estimated. First we calculate the mean of the observed usage frequency w for all item pairs that contain one specific item. And then we estimated q from w by their relationship stated in formula 13.

$$P(r) = \frac{-q^r}{r \ln(1-q)}, \quad r \geq 1 \tag{12}$$

$$w = \frac{-q}{(1-q)\ln(1-q)} \tag{13}$$

In [11] we showed that the LSD model can be fitted to co-occurrences of information products in our information broker reasonably well. However, the LSD model is only applicable to co-occurrence of items under the strict assumption of independence between the usage of all items. Of course, this assumption is not valid for some items in the data, caused by dependencies between the usage of items that cover the same topic or that are useful as complementing information sources. And exactly these dependencies are what we look for in a recommender system. The dependencies between these items violate the strict assumptions of the model and therefore produce outliers that do not follow the fitted LSD. For each pair of items we can calculate the probability that it comes from the LSD model by dividing the observed number of co-occurrences by the predicted number. This gives estimates for the type II error for each possible recommendation.

```
1  FORALL items in all transactions DO BEGIN
2    Generate the frequency distribution of co-occurrences
       from all transactions that include the item
3    Estimate the parameter q of the LSD distribution from
       the mean w of the frequency distribution
4    WHILE expected type II error rate is below a set
       threshold DO BEGIN
5      Add the item with the highest number of co-occurence
         to the list of recommendations
6      Compute the expected type II error rate for the
         new list
7    END
8  END
```

Table 3. Algorithm for computing recommendations based on repeat-buying

The algorithm in table 3 produces for each item a recommendation list that has an expected type II error rate (β) below a predefined threshold. By changing this threshold, recommendation lists with higher or lower expected β and potentially more or less recommendations can be produced by the algorithm.

The algorithm produces similar results like association rules using variable confidence and support thresholds. If we set support to 0 and we order the all association rules with the same antecedent by confidence we get the set of co-occurrences that is used to calculate the LSD. Since confidence preserves the rank order of the number of co-occurrences the selection of recommended items is made in the same sequence for both algorithms. For association rules only rules above the set minimum support and confidence are used, for the repeat-buying algorithm the set of rules is selected so that the total expected type II error rate is below a set threshold.

Like the simple association rule algorithm used in this paper the computational complexity of the repeat-buying algorithm is of quadratic order. However, an incremental update version has been developed. Its computational complexity is of quadratic order in the number of items updated, not in the total amount of items. Only actually updated LSD-models must be recomputed. This is an advantage over simple association rule algorithms which gets more pronounced as the ratio of the total number of items to the number of items updated increases. A real-world example where this difference matters is computing recommendations for the users of a large research library's online public access catalog (OPAC) with millions of items.

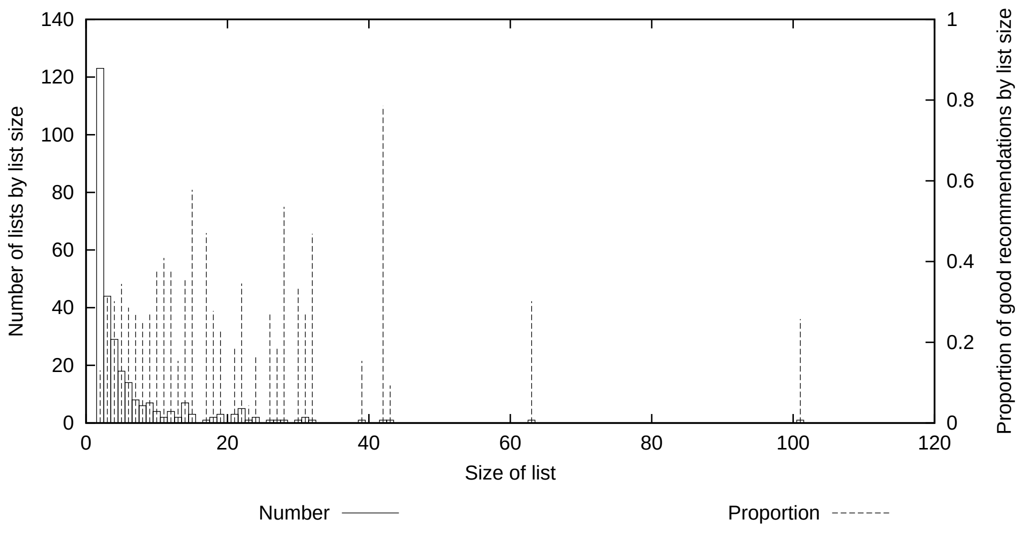

Fig. 2. Evaluated data set by size of list

6 Experimental Setup

To compare the different recommender algorithms we use a data set from the Virtual University information system at the Vienna University of Economics and Business Administration. This information system is an educational Internet information broker that provides access to online information products (lecture notes, research material, enrollment information,...) for students and researchers. An information product in this setting is defined as a set of Web pages (including other media like word processor files) that present a logical unit of information on a specific topic, e.g. a Web site on software development.

To generate recommendations for the information system we provide the recommender algorithms with a data set containing market basket data obtained from observed consumption patterns by anonymous users. A market basket contains all items used together in a single transaction (a session). We use 6 months of usage data (January to June 2001) containing 25522 transactions with more than one item. These transactions contain 3774 different items with 114128 co-occurrences of items that are possible recommendations.

For evaluation we produced for each item a list of other items that co-occurred together with the first item in at least one transactions. We selected randomly 300 lists and asked students and researchers if they would recommend the later items to a person interested in the first one. The possible answers were "recommend" or "don't recommend". In total we evaluated 1661 possible recommendations (co-occurrences of items). For 561 co-occurrences the interviewed persons said they would recommend the latter item, and for 1100 they would not. Figure 2 shows the distribution of the list size (number of items that co-occurred with one selected item) and the average proportion of items in each list that were evaluated as useful recommendations by the interviewed persons. More than 120 lists are of the size 2 (the smallest size analyzed) and only few lists have a size larger than 50. This is in line with the characteristics of other real-world datasets used in Zheng et al. [29]. The proportion of good items in the lists seems to be independent from the size of the list with an average of 0.338.

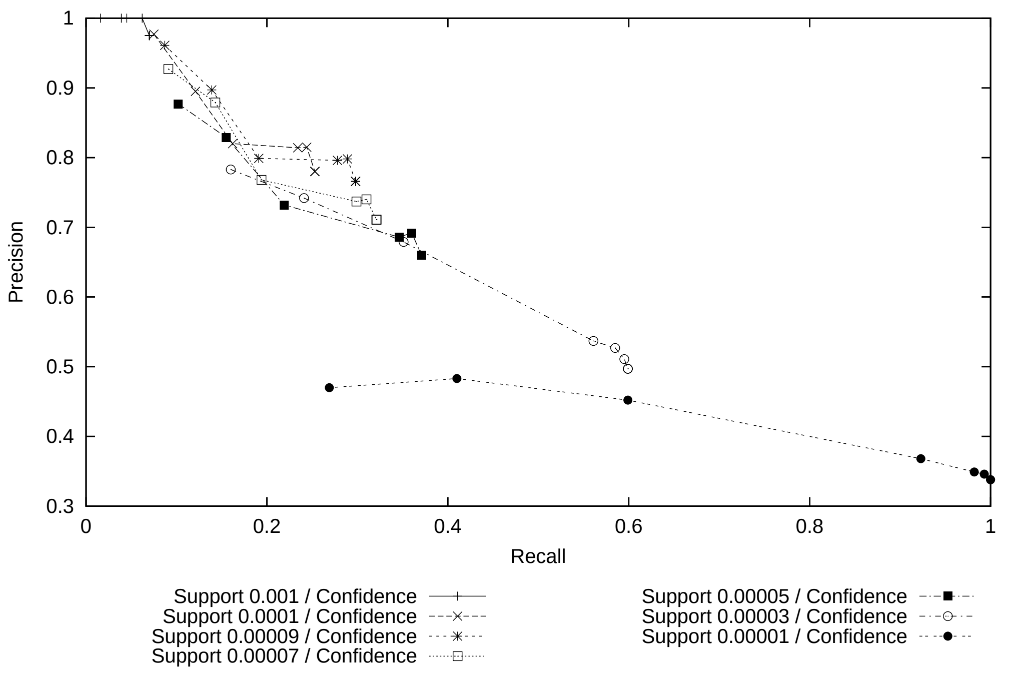

Fig. 3. Precision/recall plot for association rules with different minimum support

7 Evaluation Results

In this section we present and discuss the evaluation results for the different recommender algorithms. The algorithms used the data set described above as input and the produced recommendations where evaluated using the *useful recommendations* identified by the users.

To find sensible support values for the association rules based algorithms, we varied minimum confidence between 0.3 and 0.000001 (0.3, 0.2, 0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001) and plotted precision by recall for several support thresholds. Figure 3 depicts the precision/recall plots for the best performing support thresholds. Since the data set contains many, relatively short transactions and many items, the minimum support threshold has to be chosen relatively small (between 0.001 and 0.00001) to obtain reasonable recall.

Next we compare the performance of confidence with the alternative measures of interestingness lift and conviction. For recommender systems recommendation lists with low precision are problematic since recommending unrelated items annoys the users. Since reasonable precision starts with 0.5 and up, we use only minimum support of 0.0001 and 0.00003 (see figure 3) for the comparison. Figure 4 show the precision/recall plots for the two selected minimum supports. For the plots we varied lift between 1.5 and 1000 and conviction between 1.05 and 2. Since neither lift nor conviction perform significantly better than confidence on the data set, we use the support-confidence framework for all further investigations.

In figures 5 and 6 we compare the performance of the association rule algorithm and the repeat-buying algorithm in terms of precision/recall and accuracy by coverage. For the repeat-buying algorithm we varied the threshold between 0 and 1. For comparison, we included in both plots a recommender that chooses recommendations randomly from the co-occurrence list with the probability varying between 0 and 1. Both recommender algorithms perform significantly better in predicting the items that users qualify as useful recommendations than choosing recom-

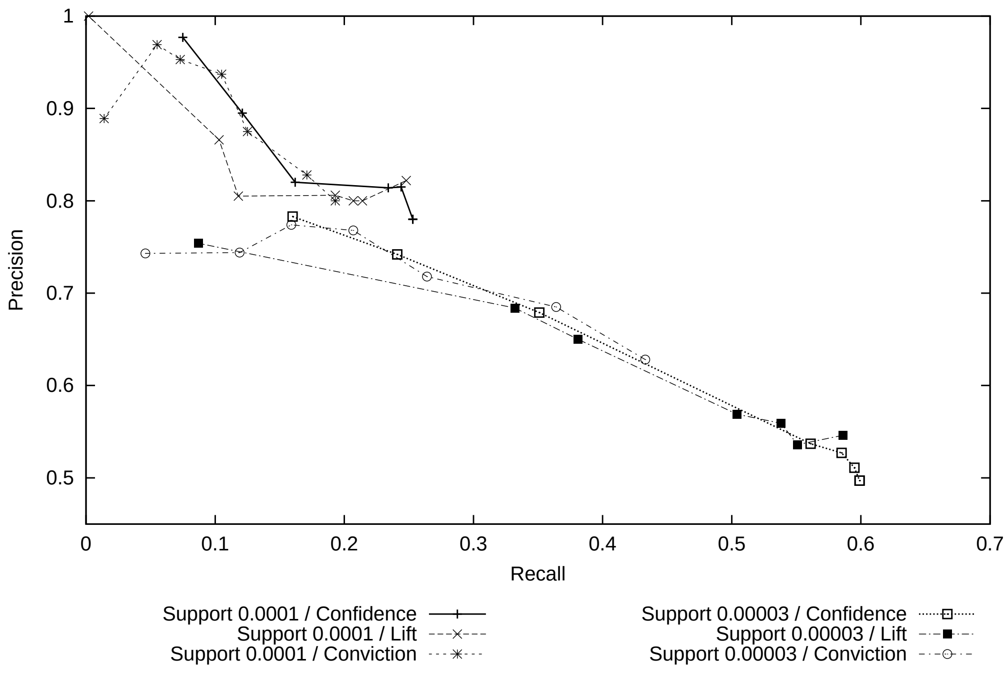

Fig. 4. Precision/recall plots for confidence, lift and conviction

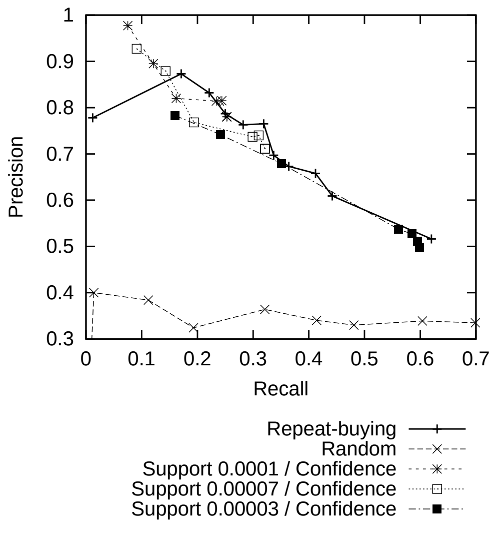

Fig. 5. Precision/recall plot

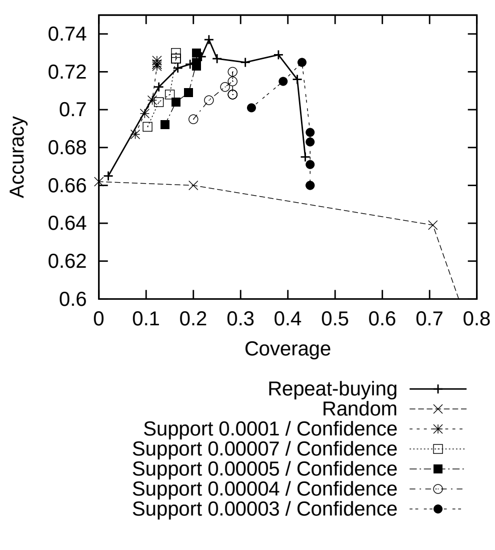

Fig. 6. Accuracy by coverage

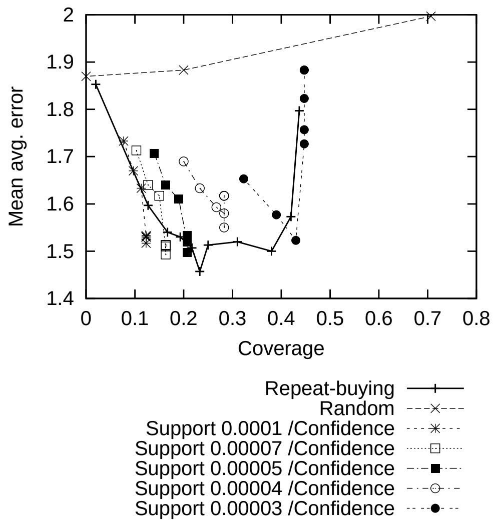

Fig. 7. Mean average error by coverage

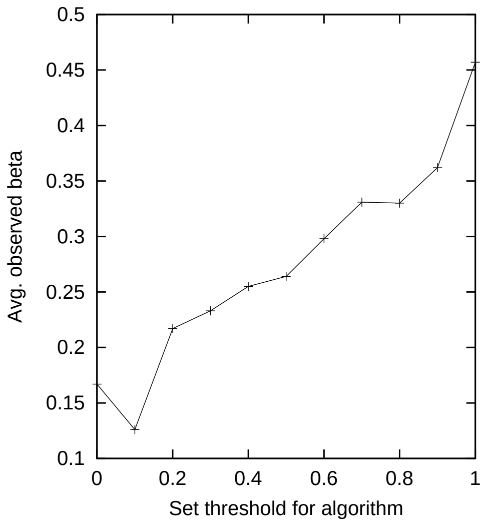

Fig. 8. β by threshold of the repeat-buying algorithm

mendations randomly. The repeat-buying recommender performs similar to the association rule algorithm with the support-confidence framework. Figure 5 shows that at reasonable recall (between 20% and 50%) both algorithms reach a precision between 60% and 80% which is acceptable for most applications. Both algorithms provide accuracy above 70% (see figure 6), however, the support-confidence framework is very brittle in respect to changes of minimum confidence and also the optimal value changes with minimum support.

Figure 7 shows the mean average error of the recommendations by the coverage produced by the algorithms for different parameters. Again, for the support-confidence framework performance deteriorates significantly for very small misspecifications of the confidence threshold. The repeat-buying algorithm shows a more robust behavior with respect to changes to its one threshold which represents the maximum expected type II error rate β in the produced recommendation lists. In contrast to minimum support and minimum confidence, β is independent of the structure and the properties of the analyzed data set.β is a very convenient parameter since it can be interpreted as the proportion of incorrect recommendations in a recommendation list. Figure 8 shows the dependence of the *average observed* β on the threshold of the repeat-buying algorithm. With the exception of the threshold values 0, 0.1, and 0.2 (where only few lists are produced) the *average observed* β is, as expected by the model, below the threshold (the *maximum expected* β).

8 Conclusion

Evaluation is an important part of the knowledge discovery process. However, for the data mining part of recommender systems the question of how well found patterns match the user's concept of *useful recommendations* is often neglected. In this paper we studied this question by comparing recommendations by human experts with the results of two recommender algorithms. In the following we summarize the results and remaining research questions:

– The results of the presented evaluation supports the widely in the KDD society accepted assumption that frequent itemsets from purchase histories or from Web usage data represent useful recommendations. With a well-parameterized algorithm an accuracy of more than 70%

and a precision between 60% and 90% have been reached in this study. Further evaluation studies on other data sets are planned and needed.

– Association rules are free of model assumptions, whereas the repeat-buying algorithm requires several quite strong model assumptions on user behavior which are hard to verify and which invite critic on the validity of the model. However, the assumptions tend to hold approximately for a wide range of applications for consumer products [9]. An investigation of how well the repeat-buying models for consumer goods apply to information products like Web sites is planned.
– The results of both algorithms when properly tuned are quite similar. However, the repeat-buying algorithm uses only one parameter which is independent of the data set and has a simple interpretation, whereas the association rule algorithm uses two parameters which strongly depend on properties of the data set. Furthermore, the repeat-buying algorithm seems to be more robust with regard to a misspecification of its parameter, whereas the association rule algorithm is more flexible and allows fine tuning.
– Both algorithms studied in this paper have a computational complexity of quadratic order. However, for incremental updates, the repeat-buying algorithm has a computational complexity of quadratic order in the number of updated items, whereas the simple association rule algorithm is of quadratic complexity on all items. For applications with millions of items the repeat-buying algorithm's possibility to perform efficient incremental updates is vital. Strategies for incremental updates of association rules need to be developed.

Since we only used one data set from one application in this paper, additional studies of other data sets are needed to confirm our findings.

References

1. Jean-Marc Adamo. *Data Mining for Association Rules and Sequential Patterns*. Springer, New York, 2001.
2. C. C. Aggarwal and P. S. Yu. A new framework for itemset generation. In *PODS 98, Symposium on Principles of Database Systems*, pages 18–24, Seattle, WA, USA, 1998.
3. R. Agrawal, T. Imielinski, and A. Swami. Mining associations between sets of items in large databases. In *Proc. of the ACM SIGMOD Int'l Conference on Management of Data*, pages 207–216, Washington D.C., May 1993.
4. Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499, Santiago, Chile, Sept 1994.
5. R. Bayardo, R. Agrawal, and D. Gunopulos. Constraint-based rule mining in large, dense databases. *Data Mining and Knowledge Discovery*, 4(2/3):217–240, 2000.
6. Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Tsur. Dynamic itemset counting and implication rules for market basket data. In *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data*, pages 255–264, Tucson, Arizona, USA, May 1997.
7. Robert Cooley, Bamshad Mobasher, and Jaidep Srivastava. Data preparation for mining world wide web browsing patterns. *Journal of Knowledge and Information Systems*, 1(1), 1999.
8. Robert Walker Cooley. Web usage mining: Discovery and application of interesting patterns from web data. Ph. d. thesis, Graduate School of the University of Minnesota, University of Minnesota, 2000.
9. A. S. C. Ehrenberg. *Repeat-Buying: Facts, Theory and Application*. Charles Griffin & Company Ltd., London, 1988.
10. Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. *From Data Mining to Knowledge Discovery: An Overview*, pages 1–36. MIT Press, Cambridge, MA, 1996.
11. Andreas Geyer-Schulz and Michael Hahsler. A customer purchase incidence model applied to recommender systems. In *ACM WebKDD 2001 Workshop on Mining Web Log Data Accross All Customer Touch Points*, 2001.

12. Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 1999 Conference on Research and Development in Information Retrieval*, pages 230–237, 1999.
13. Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the ACM 2000 Conference on Computer Supported Cooperative Work*, pages 241–250, December 2000.
14. Ron Kohavi and Foster Provost. Glossary of terms. *Machine Learning*, 30(2–3):271–274, 1988.
15. Richard D. Lawrence, George S. Almasi, Vladimir Kotlyar, Marisa S. Viveros, and Sastry Duri. Personalization of supermarket product recommendations. *Data Mining and Knowledge Discovery*, 5(1/2):11–32, 2001.
16. Weiyang Lin, Sergio A. Alvarez, and Carolina Ruiz. Efficient adaptive-support association rule mining for recommender systems. *Data Mining and Knowledge Discovery*, 6:83–105, 2002.
17. B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. *Communications of the ACM*, 43(8):142–151, 2000.
18. B. Mobasher, H. Dai, T. Luo, M. Nakagawa, Y. Sun, and J. Wiltshire. Discovery of aggregate usage profiles for web personalization. In *ACM WebKDD 2000 Web Mining for E-Commerce Workshop*, 2000.
19. Bamshad Mobasher, Honghua Dai, and Miki Nakagawa Tao Luo. Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery*, 6:61–82, 2002.
20. André Quadt. Personalisierung im e-commerce. Diplomarbeit, AIFB, Universität Karlsruhe (TH), D-76128 Karlsruhe, Germany, 2001.
21. G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.
22. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender systems–a case study. In *ACM WebKDD 2000 Web Mining for E-Commerce Workshop*, 2000.
23. U. Shardanand and P. Maes. Social information filtering: Algorithms for automating 'word of mouth'. In *Proceedings of the Computer-Human Interaction Conference (CHI95)*, Denver, CO, May 1995.
24. Myra Spiliopoulou. Web usage mining for web site evaluation. *Communications of the ACM*, 43(8):127–134, 2000.
25. Pang-Nin Tan and Vipin Kumar. Discovery of web robot sessions based on their navigational patterns. *Data Mining and Knowledge Discovery*, 6:9–35, 2002.
26. C. van Rijsbergen. *Information retrieval*. Butterworth, London, 1979.
27. S. Vucetic and Z. Obradovic. A regression-based approach for scaling-up personalized recommender systems in e-commerce. In *ACM WebKDD 2000 Web Mining for E-Commerce Workshop*, 2000.
28. Kai Yu, Xiaowei Xu, Martin Ester, and Hans-Peter Kriegel. Selecting relevant instances for efficient accurate collaborative filtering. In *Proceedings of the 10th CIKM*, pages 239–246. ACM Press, 2001.
29. Zijian Zheng, Ron Kohavi, and Llew Mason. Real world performance of association rule algorithms. In *Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining (ACM-SIGKDD)*. ACM Press, 2001.