Empirical Analysis of Attribute-Aware Recommender System Algorithms Using Synthetic Data

Karen H. L. Tso and Lars Schmidt-Thieme
Computer-based New Media Group(CGNM), Department of Computer Science, University of Freiburg, Freiburg, Germany
Email: {tso,lst}@informatik.uni-freiburg.de

***Abstract*— As the amount of online shoppers grows rapidly, the need of recommender systems for e-commerce sites are demanding, especially when the number of users and products being offered online continues to increase dramatically. There have been many ongoing researches on recommender systems and in investigating recommendation algorithms that could optimize the recommendation quality. However, adequate and public datasets of users and products have always been demanding to better evaluate recommender system algorithms. Yet, the amount of public data, especially data containing adequate content information (attributes) is limited. When evaluating recommendation algorithms, it is important to observe the behavior of the algorithm as the characteristic of data varies. Synthetic data would allow the application of systematic changes on the data which cannot be done with real-life data. Although studies on synthetic data for the use of recommender systems have been investigated, artificial data with attributes information are rarely looked into. In this paper, we review public and synthetic data that are applied in the field of recommender systems. A synthetic data generation methodology that considers attributes will also be discussed. Furthermore, we present empirical evaluations on existing attribute-aware recommendation algorithms and other state-of-the-art algorithms using real-life dataset as well as variable synthetic data to observe their behavior as the characteristic of data varies. In particular, the informativeness of attributes is being further investigated with both real-life datasets with augmented attributes sets as well as synthetic datasets with attributes. We have shown that a reasonably good overview of the behavior of attribute-aware algorithms can be obtained by using synthetic data compared to results done with real-life datasets.**

***Index Terms*— synthetic data, recommender systems, collaborative filtering, content-based filtering, attribute-aware**

I. Introduction

Recommendation problem has always been a great interest for both academic and industry. Especially, when online shopping continues to blossom after the internet-boom in the 90s, recommender systems for e-commerce are playing a more important role ever. As more users are purchasing via internet and more products are being offered online, recommender systems have been seen as a customization e-commerce tool to generate personalized recommendations or advertisements to boost up sales. Hence, recommender systems prevail in large commercial sites, such as amazon.com and ebay.com. Amazon, one of the pioneers in applying recommender systems in e-commerce, suggests products to a given user based on the purchase history of the given user and the one of other users who share similar interest to the given user. Ratings and/or comments can also be given to products by users and this information can then be used to help recommending products to other users. Typically, users are asked to identify themselves as a registered user of the site and their purchase history are stored as profiles in the system. Recommender system in Ebay is more known to generate recommendations using its feedback profile features. Usually, purchasers and sellers are allowed to provide feedbacks such as assessment of the satisfaction of the users. Most of the time, purchasers can then use these profile information as recommendations. Ebay also provides another tool called Gift Finder, which helps customers to find presents by matching the profile of the gift recipient. This type of recommender system is usually called relevance feedback, or anonymous recommender system, which purchase history or other profile information of the given user is not exactly know. Recommendations are usually offered by using information that has been provided at the time when the user uses the system. Typically, users are required to explicitly provide feedback by specifying keywords or answering questions about their interests. Implicit feedbacks of the users are usually unobtrusively obtain from the users by observing their interactions with the system by for instance recording the sequence of links clicked, printing, purchase, etc. [1].

To better evaluate algorithms for recommender system, adequate datasets of users and products have always been demanding. Yet, the amount of public datasets, especially data containing useful content information (attributes) is limited. In addition, the performance of recommender systems is highly dependent on various characteristics of the datasets. Evaluating algorithms based on only one or two datasets is often not sufficient. A more thorough

This paper is based on “Evaluation of Attribute-Aware Recommender System Algorithms on Data with Varying Characteristics,” by K. H. L. Tso and L. Schmidt-Thieme which appeared in the Proceedings of the 10th Pacific-Asia Conference, PAKDD 2006, Singapore, April 9-12, 2006.

This work was supported in part by Landesstiftung Baden-Württemberg.

analysis can be done by applying systematic changes to data, which cannot be done with real data. Thus, several researchers have already conducted studies on synthetically generated datasets to mimic the user product relationship. However, synthetic datasets that include attributes are rarely investigated. This paper is organized as follow:

- Section II discusses related works for synthetic data used in recommender systems.
- Section III talks about recommender systems and various popular techniques applied in recommender systems. Several existing hybrid methods that consider attribute information in collaborative filtering for predicting items will also be discussed.
- Section IV surveys various public datasets which are commonly used for evaluating recommender systems.
- Section V introduces a Synthetic Data Generator (SDG) which can produce user-item and user/item-attribute datasets. This SDG makes use of entropy to measure the randomness in the artificial data.
- Section VI presents our results on the empirical evaluations on several existing hybrid recommendation algorithms and other state-of-the-art algorithms using a real-life datasets as well as synthetic data. It also investigates the behavior of algorithms when the characteristic of attribute data varies.
- Section VII concludes the paper and discusses possible future works.

II. Related Works

Herlocker, Konstan, Terveen and Riedl [2] mentioned that synthetic data allow the detection for obvious flaws in the algorithms which can not be accurately modeled with real datasets. Thus, when evaluating a new domain where there is a significant research on the structure of user preferences, it maybe appropriate to first evaluate algorithm against artificial data. Hence, there have been ongoing researches in synthetic data generator for the use of recommender systems. One of the most widely known Synthetic Data Generators (SDG) in data mining is the one provided by the IBM Quest group [3]. It generates synthetic transactions to evaluate the performance of the algorithms over a large range of data characteristics. It mimics the real world transactions in the retailing environment and imitates the real world model which people tend to purchase sets of item at the same time. For example, people who buy chips also buy beer and popcorn, but other might buy only chips and beer. These synthetic data was originally intended for evaluating association rule algorithms. The characteristic of the synthetic data generated by the SDG can be controlled by changing different parameters such as the number of transactions, number of items, etc. Later on, Deshpande and Karypis used this SDG for evaluating their item-based top-N recommendation algorithm [4]. They have shown that the performance of recommendations for their algorithm is highly dependent on various characteristics of the dataset such as the number of items, the number of users, its sparsity, etc. Traupman and Wilensky tried to reproduce data by introducing skewed data to the synthetic data similar to a real dataset [5]. They used their synthetic data to evaluate their collaborative quality filtering algorithm, especially for the ability to recover the ground truth quality. They have generated two datasets which used different distributions of ground truth quality. This allowed them to measure how well their algorithm performs when presented with data whose distribution differs from what its model assumes. Another approach by Marlin, Roweis and Zemel [6] is to generate synthetic data according to the hierarchical Bayesian procedure. They focused on algorithms for unsupervised learning in the presence of non-ignorable missing data. They evaluated their work with a real-life dataset, EachMovie, which contains more than 95% of missing data. Synthetic data was generated for investating the performance of their algorithms when various strengths of missing data were presented. The synthetic data was produced by first sampling a complete dataset and then re-sampling these complete data again by missing data effect. Characteristic of the data can be changed by varying the strength of the missing data effect. Popescul, Unga, Pennock and Lawrence proposed a simple approach by assigning a fixed number of users and items into clusters evenly and then draw a uniform probability for each user and item in each cluster [7]. They generated several datasets with varying densities to examine the overfitting effect with their probabilistic models for collaborative and content-based recommendation algorithm. A similar attempt has been done for Usenet News [8], [9] as well as Aggarwal, Wolf, Wu and Yu for their horting approach [10]. Usenet News SDG cluster users together either by spreading them evenly or use other distributions. They used synthetic data to model news articles which have a fixed number of "properties" and assigned preferences of those properties to the users [2]. Aggarwal, Wolf, Wu and Yu had a more detailed description of their SDG. They introduced partitioning the synthetic data into a given number of hot items - items chosen based on their popularity to increase commonality, and a given number of cold items - items to increase coverage. Next, it assigned an average number of rated items to user and randomly sampled an equal amount of hot and cold itemsets to each user based on the hot and cold item distribution. For example, if the total number of items are set to 5000 and 4950 of these items are assigned as hotset and the rest, 50, as coldest. If the number of average rated items for each user is 20, then 10 hot and cold items will be randomly assigned to each user based on the predefined distribution of hot and cold items (e.g. P_{hot} = 0.2, P_{cold} = 0.00202). Average rating for each item was chosen from a uniform distribution from the given range of rating scale. An offset o_j, which was chosen from a normal distribution about a mean of 0, was assigned to each user j to adjust the effusiveness of the user. Users with positive offsets means they are more effusive than the average and negative means less.

Their SDG algorithm also took uncommon users, which they called "contrarian" into account. Random ratings, $RR_{i,j}$, are generated from a normal distribution with a fixed mean as well as the offset. For instance, if the rating scale range from 1 to 13, rating $r_{i,j}$ of items i for user j would be $min(13, max(1, RR_{i,j} + o_j))$. For contrarians, the reversed of ratings from normal users was considered. Thus, $r_{i,j}$ for contrarians is $r_{i,j}$ = $min(13, max(1, 14 - RR_{i,j} - o_j))$ [10]. However, Herlocker, Konstan, Terveen and Riedl [2] had commented that this approach tends to generate data which is unfair to other algorithms as it fits the algorithm proposed by the authors too well.

Although evaluating RS algorithms with synthetic data are not uncommon and various synthetic data generators for evaluating the behavior of data with various characteristic have been investigated, most of the SDG discussed above are mostly used for evaluating one specific algorithm. Furthermore, as SDG was never the main focus for most researches mentioned above, neither the SDG algorithms are described in detail nor it is generic enough to be reproducible and suitable for evaluating various RS algorithms. Besides, to the best of our knowledge, there is no prior attempt in examining SDGs for RS algorithms which considers attributes.

III. Recommender Systems

In general, recommender systems (RS) predict ratings of items or suggest a list of items that is unknown to the user. They take the users, items as well as the ratings of items into account. Thus, a recommender system consists of:

- A set of users U
- A set of items I
- An ordered set of rating values S
- A set of user ratings R, in triplets (u, i, r) where $u \in U$ and $i \in I$ and r is the rating value assigned by the user u to an item i.

Two different recommendation tasks are typically considered: (i) predicting the ratings, i.e. how much a given user will like a particular item, and (ii) predicting the items, i.e. which N items a user will rate, buy or visit next (topN). Most recommender systems derive recommendations to a user by using opinions from people who have alike tastes, called neighborhood, while concealing the real identity of the users neighborhood. When designing algorithms for recommender systems, researchers have to bear in mind some typical systematic disadvantages in the system.

- **Sparsity problem** – this has always been one of the most common challenges for recommender system. Most recommender systems suffer from having a dense database. In practice, it is rather unlikely to have a database of ratings where users will give an opinion to most of the items in the database. In fact, even active or loyal users have this problem. Sarwar, Karypis, Konstan and Reidl stated that the purchase rate of active users are lower than 1% [11]. And as the number of users and items increases, the amount of sparsity in the databases increases as well. However, many prominent RS algorithms are not able to handle spare database well. The accuracy of recommendations deteriorate as the database become sparser.
- **Shilling in recommender System** – shilling in recommender system occurs when a user attempt to provide biased ratings or opinions in order to make the recommender system to recommend a certain products more often than the others. For example, restaurant owner might trick the recommender system by giving only positive opinions of his/her restaurant. Recently, researches have been done to investigate the effectiveness of recommendation techniques to handle shilling attacks [12]–[14].

A. Recommender Systems Techniques

Many researches have attempted to come up with RS algorithms that could improve the performance of recommendations. There are various RS techniques:

Collaborative Filtering. The prevalent method in practice is Collaborative Filtering (CF) [15]. Its idea is basically the nearest neighbor method. Given some user profiles, it predicts whether a user might be interested in a certain item, based on a section of other users or items in the database. There are in general two types of collaborative filtering: user-based and item-based. Let

- $O_{u,i} \in \{0, 1\}$ specify whether item $i \in I$ occurred with user $u \in U$ (i.e., u has rated/bought/visited item i).

In user-based CF [16], recommendations are generated by considering solely the ratings of users on items, by computing the pairwise similarities between users, e.g., by means of vector similarity:

$$\text{usim}^{\text{ratings}}(u, v) := \frac{\langle R_{u,.}, R_{v,.} \rangle}{||R_{u,.}||_2 ||R_{v,.}||_2} \tag{1}$$

where u, $v \in U$ are two users and $R_{u,.}$ and $R_{v,.}$ are the vectors of their ratings.

A dualistic form of user-based CF is item-based CF [4], where similarities are computed between each pair of items i, $j \in I$.

$$\text{isim}^{\text{ratings}}(i, j) := \frac{\langle R'_{.,i}, R'_{.,j} \rangle}{||R'_{.,i}||_2 ||R'_{.,j}||_2} \tag{2}$$

To derive the recommendations for a target user u, usually only similarities of the k most-similar users are selected (neighborhood – N_u). For the case of predicting a rating of a given user u for an item i, the weighted sum of the other users are computed by:

$$\hat{r}(u, i) := \bar{r_u} + \frac{\sum_{v \in N_u \cap dom r_i} w(u, v)(r(v, i) - \bar{r_v})}{\sum_{v \in N_u \cap dom r_i} w(u, v)} \tag{3}$$

where dom r is the domain of r. For $i \in I$ we define $r_i := r|_{U \times \{i\}}$ and $w(u, v)$ is the similarity between each user v in user u's neighborhood and user u him/herself.

For the case of predicting item problem, recommendations are a list of items that is ranked by decreasing frequency of occurrence in the ratings of his/her neighbors.

$$p^{\text{cf}}(O_{u,i}=1) := \frac{|\{v \in N_u \,|\, O_{v,i}=1\}|}{|N_u|} \qquad (4)$$

Due to its simplicity and rather promising performances [15], [17], collaborative filtering has been one of the most prominent methods used in recommender systems. However, there are several drawbacks of this method:

- **New user problem** – this problem was firstly introduced by Maltz and Ehrlich as the cold-start problem [18] and has been further investigated in RS by [19]. This scenario occurs when new users do not have any ratings or purchases in their profile. As not much information is made known from the users, it usually requires a training period before there is sufficiently enough amount of preference information from the users to produce reasonably accurate prediction.
- **New item problem** – similar to the new user problem, new items also suffers to be recommended. Items will not be able to be recommended until considerable number of users has rated the new item.
- **Non-transitive association** – this problem usually lead to poor and over specific recommendations. As recommendations are heavily based on the co-rated items of the neighborhood of the user, Balabanovic and Shoham [20] suggested the challenge in recommending items to users who have uncommon preferences compared to the rest of the population. This also brings about another related problem, which is the inability to capture the relationship between two similar items that have never been rated by the same user. In this case, these two items are not considered alike [21].

Content-based Filtering. On the other hand, there are methods that rely only on attributes/content information of the items and are commonly called the Content-Based Filtering (CBF). Although this type of methods can uniquely characterize each user, it only considers the profile of the given user and disregard information of the other users. Let

- B be a set of (binary) **item attributes**,
- $D_{i,b} \in \{0,1\}$ specify whether item $i \in I$ has attribute $b \in B$,

In Content-Based Filtering, recommendations are treated as text-categorization problem. Most of the time, content information of the items are viewed as bag-of-words and a naive Bayesian classifier is used. The classifier is trained for the binary target variable $O_{u,.}$ depending on the binary predictors $D_{i,b}\ \forall b \in B$:

$$\hat{p}^{\text{cb}}(O_{u,.}=1 \,|\, D_{.,b}, b \in B) := P(O_{u,.}) \cdot \prod_{b \in B} P(D_{.,b} \,|\, O_{u,.}) \qquad (5)$$

However, similar to CF, there are several drawbacks of this method:

- **New user problem** – as content-based filtering heavily based on the profile of the user, it faces the same problem as CF of not being able to recommend products to new users who do not have any ratings or purchases in their profile.
- **Overfitting** – Maltz and Ehrlich suggested that recommendations might be circumscribed by the profile [18]. Hence, recommendations may only contain items that share the same area of interest of the user profile, but ignore possible new areas of interest of the user.

In general, CF has shown to perform better compared to content-based filtering [22], [23]. Indeed, one of the most important advantages that CF has over content-based filtering is the potential for generating serendipitous recommendations [2]. However, combining the two RS techniques, CF and CBF, have shown to solve some of the drawbacks of both techniques [21], [22], [24]. Besides, content information of items usually contains valuable information; hence it makes it desirable to include attribute information in CF models, the so called hybrid collaborative/content-based Filtering methods.

Hybrid collaborative/content-based Filtering. There are many proposals on how to integrate attributes in CF for ratings. For instance, few others attempt linear combination of recommendation of CBF and CF predictions [21], [25]–[27]. There also exist methods that apply a CBF and a CF model sequentially, i.e. predict ratings by means of CBF and then re-estimate them from the completed rating matrix by means of CF [22]. There are also further proposals on how to integrate attributes when the problem is viewed as a classification problem [28]–[30]. As we lose the simplicity of CF, we do not consider those more complex methods here.

We have selected three basic methods [31] that predict items and try to keep the simplicity of CF, but still should improve prediction results. These algorithms will be evaluated using the data generated from the SDG discussed in Section V.

- Sequential CBF and CF (adapted content-boosted CF),
- Joint Weighting of CF and CBF, and
- Attribute-Aware Item-Based CF.

All three approaches recommend topN items that contain the highest frequency of their neighboring items. Similarity between two users is computed using Vector Similarity. The first two algorithms apply CBF and CF paradigms in two separate processes before combining them together at the point of prediction. Our third approach, however, does not employ CBF algorithm; instead item attributes are directly incorporated at the model-building stage.

Sequential CBF and CF

The first approach termed, "Sequential CBF and CF"

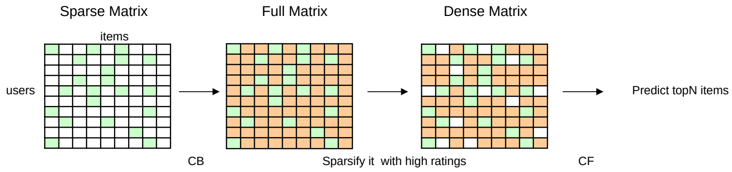

Figure 1. CF and CBF processes done in sequence

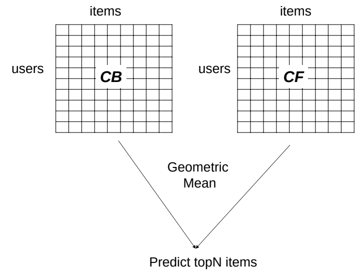

Figure 2. CF and CBF processes done in parallel

is an adapted form of [22]'s original hybrid model — Content-Boosted Collaborative Filtering (CBCF) . The reason why we do not use the CBCF directly is because the original model is intended for predicting ratings, whereas this paper focuses on the topN problem. Hence, the CBCF is adapted such that it will recommend N number of items to the user instead of inferring the rating of an item. This model is used as the hybrid baseline for evaluating the other two approaches. Recommendations are generated using CF.

As there exist no standard method in literature so far to adapt models from one prediction task to the other, we propose to do the adaptation as follow. CBCF first uses a naïve Bayesian classifier to build a content-based model for each user. Next the sparse matrix is filled by combining the actual ratings and the predicted ratings learned from the CBF predictor to form a full matrix. The adaptation takes place when applying CF. Instead of finding the weighted sum of ratings of other users to compute the prediction ratings for the current user, the full matrix is sparsified by considering solely items with high ratings. Next, standard CF (4) is applied using this dense matrix (Fig. 1).

Joint Weighting of CF and CBF

Similarly, the second approach also applies both CBF and CF. Again, naïve Bayesian classifier is utilized here. However, instead of inferring the class or rating of an item based on attributes, it predicts how much a user will like the attributes.

$$\hat{p}^{\text{cb}}(O_{u,.}=1\,|\,D_{.,b}, b\in B) := \frac{1}{k}P(O_{u,.})\cdot\prod_{b\in B}P(D_{.,b}\,|\,O_{u,.}) \tag{6}$$

where $k := P(D_{.,b}, b \in B)$.

Unlike the first approach where the two processes are done sequentially – content-based first then CF, the order of these processes is unimportant for the latter and serves as the complementary view for each other (Fig. 2). Equation 7 generates predictions using attributes (CBF) and this is joined with the outputs of CF by computing the geometric mean of the outputs. This mean combination is then used for performing the topN prediction.

$$\hat{p}(O_{u,i}=1) \sim \hat{p}^{\text{cb}}(O_{u,i}=1)^{\lambda}\cdot p^{\text{cf}}(O_{u,i}=1)^{1-\lambda} \tag{7}$$

with $\lambda \in [0, 1]$, and λ is used to weight the content-based and collaborative methods, e.g., for $\lambda = 0$, we get pure collaborative filtering and for $\lambda = 1$, pure content-based filtering.

Attribute-Aware Item-Based CF

The third approach extends the Item-Based topN CF [4]. Rather than using CBF algorithms, it exploits the content/attribute information by computing the similarities between items using attributes thereupon combining it with the similarities between items using user ratings. This is shown in (9), where $\text{isim}^{\text{ratings}}$ corresponds to the item similarities computed using Vector Similarity with the ratings and $\text{isim}^{\text{attributes}}$, computed with the attributes.

$$\text{isim}^{\text{attributes}}(i,j) := \frac{\langle D_{i,.}, D_{j,.}\rangle}{||D_{i,.}||_2||D_{j,.}||_2} \tag{8}$$

$$\text{isim}^{\text{combined}} := (1-\lambda)\ \text{isim}^{\text{ratings}} + \lambda\ \text{isim}^{\text{attributes}} \tag{9}$$

with $\lambda \in [0, 1]$.

The same as standard CF, at the time of predication, only the similarities of the k most similar users are considered. As it applies to both $\text{isim}^{\text{ratings}}$ and $\text{isim}^{\text{attributes}}$, k has to be defined for both cases. In our experiments, k for computing the similarities with user ratings space various and k for attribute space is set to 10. Again, λ is used to adjust the corresponding weight on CBF and CF. In this case, setting λ to 0 is the same as computing pure item-based. This approach is very similar to Mobasher, Jin and Zhou [24], which also integrated the semantic similarities for items with rating similarities. The difference of these two approaches are that [24] used structured semantic information of items for similarity computation and their algorithm aimed at the predicting ratings problem, whereas attribute-aware item-based CF focuses on the predicting items problem.

IV. DATA FOR EVALUATING RS ALGORITHMS

When evaluating recommendation algorithms, suitable datasets of users and items have always been demanding, especially when diversity of public data is limited. In fact, one of the most significant challenges in the field of recommender system is the lack of variety in publicly available collaborative filtering datasets. It is not an easy task for most researchers to collect enough data to validate their research hypotheses. Thus, most researchers in the field of recommender systems are restricted to the few existing datasets available publicly [2].

Real-life datasets. Publicly available datasets which are suitable for evaluating recommender systems are:

- **MovieLens** (http://www.grouplens.org/data) - this data are collected from a web-based recommender system for movies. Currently, they have made two datasets publicly available. The first one consists of 100,000 ratings (from a scale of 1-5) for 1682 movies by 943 users. The second one consists of approximately 1 million ratings for 3900 movies by 6040 users.
- **EachMovie** (http://www.research.digital.com/SRC/eachmovie/) - this datasets are collected from EachMovie movie recommender by HP/Compaq Research (formerly DEC Research) group. This data is made available when the recommender service was terminated. The publicly released datasets contains 2,811,983 ratings (from a scale of 1-5) entered by 72,916 for 1628 different movies. The EachMovie Dataset remained available until October 2004 when it was retired.
- **Jester** (http://www.ieor.berkeley.edu/∼goldberg/jester-data/) - another ratings datasets has been provided by Ken Goldberg from UC Berkeley who released a dataset from the Jester Joke Recommender System. This dataset contains 4.1 million continuous ratings (from a scale of -10.00 to +10.00) of 100 jokes from 73,496 users.
- **Book crossing datasets** (http://www.informatik.uni-freiburg.de/ cziegler/BX/) – the Book Crossing (BX) dataset was collected by Ziegler, McNee, Konstan and Lausen [32] from the Book-Crossing community. It contains 1,149,780 ratings (from a scale of 1-10) of 271,379 books from 278,858 users.

Most research experiments are mainly based on these three domains of data. Other domains of datasets have also been looked into. For instance, music [33], digital cameras [34], other e-commerce site purchasing transactions [4], etc. However, more diverse datasets are usually corporate asset and thus are not made publicly available. In general, MovieLens and EachMovie datasets are the main experimental data sources for most researchers in the field related to recommender systems [4], [16], [19], [22].

Data with attributes. In general, there exist two types of attributes: user attributes and item attributes. Most public datasets contains both types of information. However, since a lot of users provide false personal information, such as name, age, address, etc, to protect their privacy [35], whether user attributes, such as demographic data, are reliable are often questionable. Therefore, in this paper we only consider item attributes. Although most public datasets comes with attributes information, the amount, quality as well as the variety of these attributes are limited and often require further means to extract additional attributes. Furthermore, in spite of the fact that attributes information are typically obtainable in practice, additional work has to be done to extract this information. Some publicly available sources for obtaining attributes that are commonly used for evaluating RS algorithms are:

- **Internet Movie Database (IMDB)** (http://www.imdb.com/) - it contains an enormous collection of movie information. For instance, title, genres, directors, awards, etc. As public movie datasets such as EachMovies and MovieLens usually only contains attributes information such as title and genres, IMDB has been a very popular source for collecting supplementary attributes information that does not come with those public datasets [21], [23], [24], [28], [36]. To extract this additional information from IMDB, one common approach is to use a web crawler. For instance, Melville, Mooney and Nagarajan [22] used a simple crawler to collect content information from IMDB. They used the links provided from the EachMovie dataset and collected information such as title, cast, director, plot summary, keywords, user comments, external reviews, etc. from the various links off the main URL.
- **Amazon.com** (http://amazon.com) - in addition to employing recommender system in their online shop, their product classification taxonomy and other products information provided from the sites could also be valuable attribute information. For instance, Ziegler, Schmidt-Thieme and Lausen [37] have used a crawler to extract Amazon.com's book classification taxonomy, which contains 13,525 distinct topics, as content information for their recommendation algorithms. Similar manner has been done for the Amazon.coms movie taxonomy, where 16,481 hierarchically arranged topics are extracted [38].

Beside having a collection of attributes, previous research has shown that the informativeness of attributes affects the quality of the recommendations greatly. Thus, special effort is needed for selecting an assortment of useful attributes. Otherwise, attributes could act as noises and thus lead to poor recommendations [39]. Therefore, it is desirable to use synthetic data to evaluate RS algorithms, especially attribute-aware RS algorithms, before using real-life datasets. Also, one should investigate the behavior of the algorithms as systematic changes are applied to the data. Although there are already a few attempts in generating synthetic data for the use in RS, to our best knowledge, there is no prior approach in generating synthetic data for evaluating recommender algorithms that incorporate attributes.

V. Synthetic Data Generator

The synthetic data consists of several components:

- **User Cluster** - cluster of users which can be viewed as the neighborhood or community of similar users.

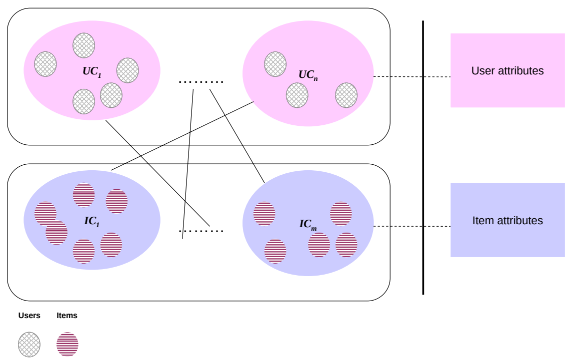

Figure 3. Overview structure of synthetic data.

- **Item Cluster** - cluster of items which can be viewed as the neighborhood or community of similar items.
- **User Attributes** - attributes/content information that belongs to the users. For example, demographic information of the users.
- **Item Attributes** - attributes/content information that describes the items. For example, the titles or genres of movies.

In general, the SDG can be divided into two phases: drawing distributions and sampling data. In the first phase, it draws distribution of User Cluster (UC) and Item Cluster (IC). Next it affiliates UC or IC with user/item attribute respectively as well as to associate the UC and IC. Using these generated correlations, the users, items, ratings and item/user-attribute datasets can then be produced in the second phase. Fig. 3 presents an overview of how the artificial data are generally structured.

A. *Drawing Distributions*

To create the ratings and attributes datasets, we generate five random distributions models:

- $P(UC)$, how users are distributed in N number of UC.
- $P(IC)$, how items are distributed in M number of IC.
- $P(A|UC)\ \forall\ UC$, how user attributes (A) are distributed in UC.
- $P(B|IC)\ \forall\ IC$, how item attributes (B) are distributed in IC.
- $P(UC|IC)\ \forall\ IC$, how UC are distributed in IC.
- q be the probability that an item in IC_i is assigned to UC_j

The SDG first draws $P(UC)$ and $P(IC)$ from a Dirichlet distribution (with parameters set to 1). This asserts that the sum of $P(UC)$ or $P(IC)$ forms to one. $P(B|IC)$ shows the affiliation of item attributes with the item clusters by drawing from a special Chi-square distribution rejecting values greater than 1. Likewise, the correlation between UC and IC, $P(UC|IC)$, as well as the correlation between user attributes and user clusters, $P(A|UC)$, are done with similar manner. However, the attribute-aware CF algorithms we discuss in this paper do not take user-attributes into account. The overall drawing distributions process is summarized in (Algo. 1).

Algorithm 1 Drawing distribution

Input: $|A|, |B|, N, M, \epsilon_A, \epsilon_B, \epsilon_C$
Output: $P(UC), P(IC), P(A|UC), P(B|IC), P(UC|IC)$
  $h = 0$
  $P(UC) \sim Dir_{a_1,a_2...,a_N}$
  $P(IC) \sim Dir_{b_1,b_2...,b_M}$
repeat
    $P(B|IC)_h = S\chi^2ED(|B|, M, h, \epsilon_B)$
    $P(UC|IC)_h = S\chi^2ED(N, M, h, \epsilon_{IC})$
    $P(A|UC)_h = S\chi^2ED(|A|, N, h, \epsilon_A)$
    $h = h + 0.1$
until $h < 1$

Algorithm 2 Drawing Special χ^2 distribution with specified entropy values

$S\chi^2ED(n, m, H_{XY}, \epsilon_{XY})$:
$d = 1$
repeat
  $P(X_i|Y_j) \sim \chi^2_d|_{[0,1]} \quad \forall i = 1...n, \forall j = 1...m$
  $d = d + 1$
until $|H(X|Y) - H_{XY}| < \epsilon_{XY}$
return $P(X|Y)$

By virtue of the randomness in those generated models, it is necessary to control or to measure the informativeness of these random data. Hence, we apply the Information Entropy and compute the average normalized entropy of the models.

$$H(X) = -\sum_{x\in \mathrm{dom}(X)} \frac{P(x)\log_2 P(x)}{\log_2|\mathrm{dom}(X)|}. \quad (10)$$

The conditional entropy for the item-attribute data therefore is:

$$H(B_i|IC) = -\sum_{b=0}^{1}\sum_{j\in \mathrm{dom}\ IC} \frac{P(B_i = b, IC = j)\cdot \log_2 P(B_i = b|IC = j)}{\log_2|\mathrm{dom}\ IC|} \quad (11)$$

In our experiment, $P(B|IC)$ is sampled eleven times for eleven different entropy values from 0 to 1 with 0.1 interval. By rejection sampling, $P(B \mid IC)$ is drawn iteratively with various Chi-square degrees of freedom until $H(B|IC)$ reaches desired entropies (Algo. 2). Other types of distribution have also been examined, yet, Chi-square distribution has shown to give the most diverse entropy range. We expect that as the entropy increases, which implies the data is less structured, the recommendation quality should decrease.

B. *Sampling Data*

Once these distributions have been drawn, users, items, ratings and item-attributes data are then sampled accordingly to those distributions. Firstly, users are

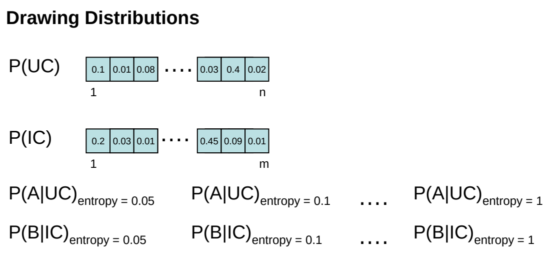

Figure 4. Example of drawing distribution

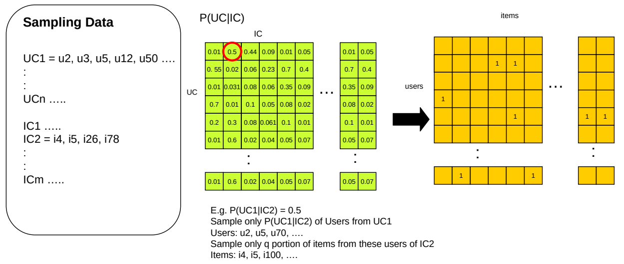

Figure 5. Example of sampling

assigned to user clusters by random sampling from $P(UC)$. Similar procedure, applies for sampling items. The user-item(ratings) data is generated by first sample $P(UC_l|IC_k)$ of users belonging to UC_l who prefer items in IC_k, then sample q portion of items of IC_k to these sampled users. The affiliation between items and attributes is done by sampling $P(B|IC)$ of items which contain attribute B. The same procedure can be applied to generate the user-attributes datasets. The overall sampling data process is summarized in (Algo. 3).

Algorithm 3 Sampling data

$uc_u \sim P(UC)$
$ic_i \sim P(IC)$
$oc_{l,k} \sim P(UC_l|IC_k)$
$o_{u,i} \sim binom(q) \quad \forall u, i : oc_{uc_u, ic_i} = 1$
$o_{u,i} = 0 \quad$ else
$b_{i,t} \sim P(B_t|IC = ic_i)$

First, the user cluster of user u , uc_u, and item cluster of item i, ic_i are sampled. Next, user of cluster l who prefer item of cluster k , $oc_{l,k}$, as well as the occurrence of user of uc_u prefers item of ic_i , $O_{u,i}$, are sampled. Finally, the sampling of item i contains attribute t, $b_{i,t}$ is also done. Fig. 4 and Fig. 5 show an example for drawing distributions and sampling the data.

VI. Evaluation and Experimental Results

In this section, we present the evaluation of the selected attributes-aware CF algorithms using a real-life dataset as well as artificial data generated by the SDG discussed in Section V. Performances of the these algorithms will be compared with their corresponding non-hybrid base models, which do not integrate attributes, i.e. user-based and item-based CF, as well as pure content-based (naïve Bayesian classifier). We will present our results on the experiments on sensitivity of attributes using a movie dataset with an augmented movie attribute set. In addition, the behavior of the algorithms after supplement of attributes with varying characteristics will also be observed.

Metrics. Our paper focuses on the item prediction problem, which is to predict a fixed number of top recommendations and not the ratings. Suitable evaluation metrics are Precision, Recall and F1. Similar to Sarwar, Karypis, Konstan and Reidl [16], our evaluations consider any item in the recommendation set that matches any item in the testing set as a “hit”. F1 measure is used to combine Precision and Recall into a single metric.

$$\text{Precision} = \frac{\text{Number of hits}}{\text{Number of recommendations}}$$

$$\text{Recall} = \frac{\text{Number of hits}}{\text{Number of items in test set}}$$

$$\text{F1} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

A. Results with real-life datasets with attributes

The three hybrid algorithms discussed in section III have been evaluated with the MovieLens (ml) datasets [31]. Since our algorithms do not take the actual ratings into account, the ratings are treated as binary values of whether the user has seen, rating of “1”, or not seen, rating of “0”, a movie. We have chosen the ml dataset containing approximately one million ratings of 3592 movies made by 6,040 users. In addition, the genres of each movie are provided. There are in total 18 different genres from the ml dataset. The genres of each movie, which are identical to the ones provided by the IMDB, are selected as the content information/attributes for each item. The experiments are tested on ten random subsets of the ml dataset with 1000 users and 1500 items each. The results we present here are the average of the ten random trials (Fig. 6). Confer [31] for further detail information on experiment.

B. Experiments on the effect of attributes

As the number of attributes increases, quantitative measures for attribute selection are needed to filter the irrelevant ones. Thus, it would be interesting to examine the performance of RS as the quality of the data decreases. We have done empirical analysis to investigate this matter by looking at an augmented sets of movie attributes from the Amazon taxonomy of 1074 different genres/classes

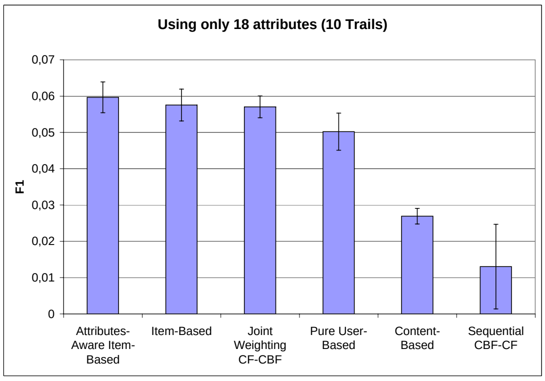

Figure 6. F1 with 18 attributes

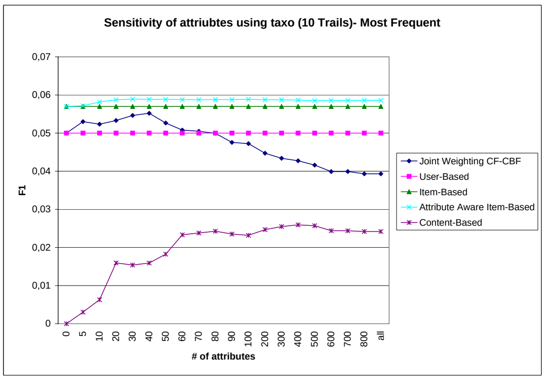

Figure 7. Vary # of Most Freq. attributes

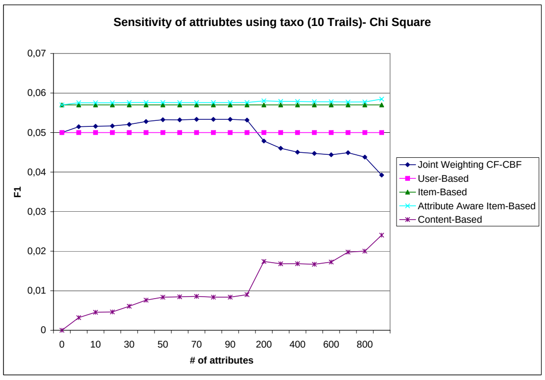

Figure 8. Vary # of best χ^2 attributes

provided by Ziegler [38]. Ten other subsets of ml data using two quantitative measures to select informative attributes are used in this experiment [39]. The first measure considers the total number of attribute occurrences (attribute frequency) and second one considers the χ^2 measure between item occurrences and attributes. Confer [39] for further information on the experiments. We expected that good attribute-aware algorithms are sensitive to attributes as this implies attributes could influence the performance of recommender systems. This means when a reasonable amount of relevant attributes are added to the system, the quality of recommendations should increase. In addition, we anticipated that the quality of the recommendations increases gradually as useful attributes are introduced to the system and when enough attributes are added to reach the maximum informativeness of the attributes, the quality should then decrease. As shown in both figures, Fig. 7 and Fig. 8, the selection of attributes does affect the quality of topN recommendation. Independent from the qualities measure, both results reflects similar trends. They have shown that in Joint-Weighting CF-CBF, the quality increases gradually, reaches its peak and decreases dramatically as more irrelevant attributes are appended. On the other hand, in attribute-aware item-based CF, the quality of attributes has almost no effect on the quality of the recommendations. Thus, Joint-Weighting CF-CBF is more attribute-aware or attribute-sensitive than Attribute-aware item-based CF (Attr-Item-based CF).

C. Results with synthetic data with attributes

From our previous findings [39], we have shown quality of attributes does have a great impact on the performance of the algorithms and that by adding useful attributes to the system, the performance of attribute-aware RS algorithms reaches its peak when the informativeness of attributes attain maximum. On this basis, we can examine this property further by using synthetic data with varying attribute informativeness. Five different trials of synthetic data are generated with the following parameter settings are summarized in Table I.

Parameters. Due to the nature of collaborative filtering, the size of neighborhood has significant impact on the recommendation quality [16], [40]. Thus, each of the randomly generated data should have an assorted neighborhood sizes for each method. In our experiments, we have selected optimal neighborhood sizes and λ parameters for the hybrid methods by means of a grid search. See Table II. Lambda is used to weight the contribution of attribute-dependent and rating-dependent models. Threshold and max, for the Sequential CBF-CF are set to 50 and 2 accordingly as chosen in the original model [22]. For Attr-aware item-based CF, the neighborhood size for computing the similarities in attribute space is set to 10. For more detail explanation of the parameters used in those algorithms, please refer to [31] and [22]. As our algorithms do not consider user attributes, the SDG only generates models for item attributes.

For each trial, we produce one dataset of user-item (ratings) and eleven different item-attributes datasets with increasing entropy from 0-1 with 0.1 intervals, by rejection sampling. In addition, to reduce the complexity of the experiment, it is assumed that the correlation between the user and item clusters to be fairly well-structure and have constant entropy of 0.05. The results of the average of five random trials where only item-attributes with entropy of 0.05 are presented in Fig. 9. As shown in Fig. 9, Joint-Weighting CF-CBF achieves the highest Recall value by around 4% difference with respect to its base method. On the other hand, Attr-Item-based CF does not seem to be

TABLE I.
THE PARAMETERS CHOSEN FOR THE RESPECTIVE ALGORITHMS.

Description	Symbol	Value
Number of users	n	250
Number of items	m	500
Number of User Clusters	N	5
Number of Item Clusters	M	10
Number of Item Attributes	$\|B\|$	50
Probability of i in IC assigned to a UC	q	0.2

TABLE II.
THE PARAMETERS CHOSEN FOR THE RESPECTIVE ALGORITHMS.

Method	Neighborhood Size	λ
user-based CF	35-50	–
item-based CF	40-60	–
joint weighting CF–CBF	35-50	0.15
attr-aware item-based CF	40-60	0.15

effective at all as attributes are appended to its base model. It also has a very high standard deviation. This suggests that the algorithms to be rather unstable and unreliable. Although Melville, Mooney and Nagarajan [22] reported that Content-Boosted CF performed better than user-based and pure CBF for ratings, it is not able to provide quality top-N recommendations for items in our experiments. Therefore, we will focus our evaluation on the other two algorithms in the rest of the paper.

As one of the aims of the paper is to examine the behavior of the models as the characteristic of data varies, what is more important is to observe the performance as entropy varies. As anticipated, the recommendation quality increases, when more structure are presented in the data. The results of an average of five random trials of item-attribute datasets with eleven various entropies are presented in Fig. 10. We can see that for both Attr-Item-based CF and Joint-Weighting CF-CBF algorithms, the quality of recommendation reaches its peaks when the entropy approaches zero and it gradually decreases as entropy increases. As for Attr-Item-based CF, although it carries the right entropy trend, its peak does not surpass its base model and the quality drops gradually below its base model, which does not make use of attributes. On the other hand, for Joint-Weighting CF-CBF, the value of recall descends gradually as the entropy raises, still the recall maintain above its base-model until entropy approaches 1 where recall plummets to below its base-line score.

The results shown in this experiment confirm our findings with the real-life datasets that Joint-Weighing CF-CBF is more attribute sensitive than Attr-Item-based CF. This shows that reasonably good overview of the behavior of attribute-aware algorithms can be obtained by using synthetic data generated by the SDG discussed here by varying the entropies of attributes in the system.

VII. Conclusion and Future Works

Due to the lack of diversity in publicly available collaborative filtering datasets, most researchers in the field of RS are confined to a few public datasets. However, to compare the recommendations quality of different algorithms, it is often not enough to just evaluate the

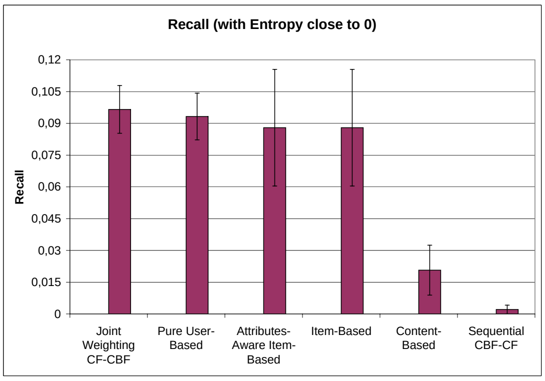

Figure 9. Recall by selecting item-attributes with entropy $\leq$ 0.05

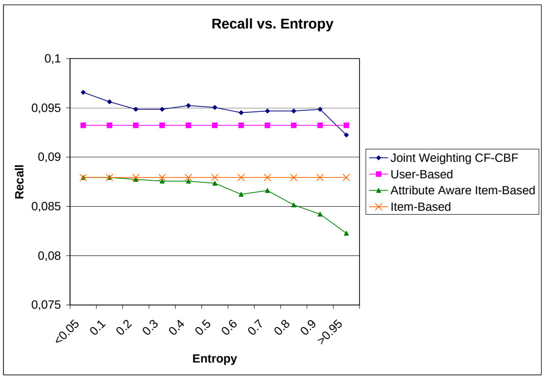

Figure 10. Recall vs. Entropy ranging from 0-1

algorithms on only one or two datasets. In this paper, we first have discussed empirical evaluations on existing attribute-aware recommendation models and other state-of-the-art algorithms with real-life datasets. Next, we have investigated the sensitivity of attributes by using real-life datasets with augmented attributes sets. We have discovered that the quality of attributes affects the performance of recommendations greatly. We have shown that Joint-Weighting CF-CBF is sensitive to attributes, and that when reasonable amount of informative attributes are added to the system, it should improve the recommendations quality. Finally, evaluations with synthetic data have also been done to further investigate this matter with different entropies of attributes. We have tested the hybrid CF algorithms by varying informativeness of the attribute. Again, we expected that good attribute-aware RS algorithms should improve the quality of recommendations as the informativeness of attributes increases. Experiments with synthetic data have validated our previous findings with real-life datasets that Joint-Weighting CF-CBF is sensitive to attributes. The other algorithms do not seem to be sensitive to attributes. As for future work, the SDG could be ameliorated by adding more structural dependency between clusters. In addition, currently the data are only controlled by the entropy of item-attribute datasets; however, other distributions such as the user-item data should also be investigated when various entropies are considered. Furthermore, more extensive experiments should be done to examine the effect of varying other parameters settings and to conduct an empirical evaluation with models that predict ratings.

ACKNOWLEDGMENT

We thank the reviewers for their detailed comments. This work was supported in part by a grant from Landesstiftung Baden-Württemberg on Learning Attribute-Aware Recommender Systems.

REFERENCES

[1] J. J. Rocchio, "Relevance feedback in information retrieval," in *The Smart retrieval system - experiments in automatic document processing*, G. Salton, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1971, pp. 313–323.

[2] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.

[3] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," *Readings in database systems (3rd ed.)*, pp. 580–592, 1998.

[4] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms." *ACM Transactions on Information Systems. Springer-Verlag*, vol. 22/1, 2004.

[5] J. Traupman and R. Wilensky, "Collaborative quality filtering: Establishing consensus or recovering ground truth?" in *Proceedings of the KDD Workshop on Web Mining and Web Usage Analysis, in conjunction with the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004)*, 2004.

[6] B. Marlin, S. Roweis, and R. Zemel, "Unsupervised learning with non-ignorable missing data," in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2005, pp. 222–229.

[7] A. Popescul, L. Ungar, D. Pennock, and S. Lawrence, "Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments," in *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, 2001, pp. 437–444.

[8] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "GroupLens: applying collaborative filtering to Usenet news," *Communications of the ACM*, vol. 40, no. 3, pp. 77–87, 1997.

[9] B. N. Miller, J. Riedl, and J. A. Konstan, "Experiences with grouplens: Making usenet useful again," in *1997 USENIX Technical Conference*, 1997.

[10] C. C. Aggarwal, J. L. Wolf, K. L. Wu, and P. S. Yu, "Horting hatches an egg: A new graph-theoretic approach to collaborative filtering," in *Proceedings of the ACM Knowledge Discovery and Data Mining Conference (KDD'99)*, 1999, pp. 201–212.

[11] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*. New York, NY, USA: ACM Press, 2001, pp. 285–295.

[12] R. Burke, B. Mobasher, and R. Bhaumik, "Limited knowledge shilling attacks in collaborative filtering systems," in *Proceedings of the Third Workshop on Intelligent Techniques for Web Personalization (ITWP'05)*, 2005.

[13] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams, "Effective attack models for shilling item-based collaborative filtering systems," in *Proceedings of the Workshop on Knowledge Discovery in the Web (WebKdd-05)*, 2005.

[14] S. K. Lam and J. Riedl, "Shilling recommender systems for fun and profit," in *Proceedings of the 13th international conference on World Wide Web*. New York, NY, USA: ACM Press, 2004, pp. 393–402.

[15] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[16] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," in *Proceedings of the Second ACM Conference on Electronic Commerce (EC'00)*, 2000, pp. 285–295.

[17] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in *Proceedings of the ACM1994 Conference on Computer Supported Cooperative Work*. Chapel Hill, North Carolina: ACM, 1994, pp. 175–186.

[18] D. Maltz and K. Ehrlich, "Pointing the way: Active collaborative filtering," in *Proceedings of CHI'95 Human Factors in Computing Systems*, Denver, CO, 1995, pp. 202–209.

[19] A. Schein, A. Popescul, L. Ungar, and D. Pennock, "Generative models for cold-start recommendations," in *Proceedings of the 2001 SIGIR Workshop on Recommender Systems*, 2001.

[20] M. Balabanovic and Y. Shoham, "Fab: Content-based, collaborative recommendation," *Commun. ACM*, vol. 40, pp. 66–72, 1997.

[21] Q. Lin and B. M. Kim, "An approach for combining content-based and collaborative filters," in *Proceedings of the Sixth international workshop on Information retrieval with Asian languages (ACL-2003)*, 2003, pp. 17–24.

[22] P. Melville, R. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering," in *Proceedings of Eighteenth National Conference on Artificial Intelligence (AAAI-2002)*, 2001, pp. 187–192.

[23] L. Si and R. Jin, "Unified filtering by combining collaborative filtering and content-based filtering via mixture model and exponential model," in *Proceedings of the Thirteenth Conference on Information and Knowledge Management (CIKM-04)*, 2004, pp. 156–157.

[24] B. Mobasher, X. Jin, and Y. Zhou, "Semantically enhanced collaborative filtering on the web," in *Proceedings of the First EuropeanWeb Mining Forum – EWMF 2003*. Springer, 2003, pp. 57–76.

[25] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin, "Combining content-based and collaborative filters in an online newspaper," in *Proceedings of ACM SIGIR Workshop on Recommender Systems*, August 1999.

[26] N. Good, J. B. Schafer, J. J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, "Combining collaborative filtering with personal agents for better recommendations," in *Proceedings of the 1999 Conference of the American Association of Artifical Intelligence (AAAI-99)*, 1999.

[27] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artif. Intell. Rev.*, vol. 13, no. 5-6, pp. 393–408, 1999.

[28] J. Basilico and T. Hofmann, "Unifying collaborative and content-based filtering," in *Proceedings of the twenty-first international conference on Machine learning*. New York, NY, USA: ACM Press, 2004, pp. 65–72.

[29] C. Basu, H. Hirsh, and W. Cohen, "Recommendation as classification: Using social and content-based information in recommendation." in *Proceedings of the 1998 Workshop on Recommender Systems. AAAI Press, Reston, Va.*, 1998, pp. 11–15.

[30] L. Schmidt-Thieme, "Compound classification models for recommender systems," in *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM 2005)*, 2005, pp. 378–385.

[31] K. H. L. Tso and L. Schmidt-Thieme, “Attribute-aware collaborative filtering,” in *Proceedings of 29th Annual Conference of the German Classification Society 2005. Magdeburg, Germany*, 2005.

[32] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, “Improving recommendation lists through topic diversification,” in *Proceedings of the 14th international conference on World Wide Web*. New York, NY, USA: ACM Press, 2005, pp. 22–32.

[33] P. Cano, M. Koppenberger, and N. Wack, “An industrial-strength content-based music recommendation system,” in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM Press, 2005, pp. 673–673.

[34] M. Stritt, K. H. L. Tso, L. Schmidt-Thieme, and D. Schwarz, “Anonymous recommender systems,” *ÖGAI Journal Jahrgang 2005*, vol. 4, pp. 4–11, 2005.

[35] J. B. Schafer, J. A. Konstan, and J. Riedl, “E-commerce recommendation applications,” *Data Min. Knowl. Discov.*, vol. 5, no. 1-2, pp. 115–153, 2001.

[36] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, “Methods and metrics for cold-start recommendations,” in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM Press, 2002, pp. 253–260.

[37] C.-N. Ziegler, L. Schmidt-Thieme, and G. Lausen, “Exploiting semantic product descriptions for recommender systems,” in *Proceedings of the 2nd ACM SIGIR Semantic Web and Information Retrieval Workshop*, Sheffield, UK, July 2004.

[38] C.-N. Ziegler, “Towards decentralized recommender systems,” Ph.D. dissertation, University of Freiburg, 2005, department of Computer Science.

[39] K. H. L. Tso and L. Schmidt-Thieme, “Sensitivity of attributes on the performance of attribute-aware collaborative filtering,” in *Proceedings of the 5th biennial meeting of CLAssification and Data Analysis Group (CLADAG’05)*, 2005.

[40] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, “An algorithmic framework for performing collaborative filtering,” in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM Press, 1999, pp. 230–237.

Karen H. L. Tso received her Bachelors in Electrical Engineering at the University of British Columbia in 2002, a Master in Communication and Media Engineering at University of Applied Science Offenburg (FH Offenburg) in 2004. Currently she is pursuing her Ph.D. in Computer Science at Albert-Ludwigs-University of Freiburg, researching on recommender systems incorporated with attributes. Her project is funded by Landesstiftung Baden-Wuerttemberg and supervised by Prof. Dr. Dr. Lars Schmidt-Thieme.

Lars Schmidt-Thieme received his Masters in Mathematics at the University of Heidelberg and his Ph.D. in Business Engineering at the University of Karlsruhe. He is now professor for computer science at University of Freiburg, Germany, where he leads the Computer Based New Media Group that works on machine learning and data mining as well as on web technologies.