Taxonomy-driven Computation of Product Recommendations

Cai-Nicolas Ziegler Georg Lausen Lars Schmidt-Thieme

Institut für Informatik, Universität Freiburg
Georges-Köhler-Allee, Gebäude 51
79110 Freiburg i.Br., Germany

{cziegler,lausen,lst}@informatik.uni-freiburg.de

ABSTRACT

Recommender systems have been subject to an enormous rise in popularity and research interest over the last ten years. At the same time, very large taxonomies for product classification are becoming increasingly prominent among e-commerce systems for diverse domains, rendering detailed machine-readable content descriptions feasible. Amazon.com makes use of an entire plethora of hand-crafted taxonomies classifying books, movies, apparel, and various other goods. We exploit such taxonomic background knowledge for the computation of personalized recommendations. Hereby, relationships between super-concepts and sub-concepts constitute an important cornerstone of our novel approach, providing powerful inference opportunities for profile generation based upon the classification of products that customers have chosen. Ample empirical analysis, both offline and online, demonstrates our proposal's superiority over common existing approaches when user information is sparse and implicit ratings prevail.

Categories and Subject Descriptors

H3.3 [**Information Storage and Retrieval**]: Information Retrieval and Search—*Information Filtering*; I.2.6 [**Artificial Intelligence**]: Learning—*Knowledge Acquisition*

General Terms

Algorithms, Experimentation, Performance, Human Factors

Keywords

Recommender systems, machine learning, taxonomies

1. INTRODUCTION

Recommendation systems [22] are becoming increasingly popular thanks to their great utility in providing people with recommendations of goods they might appreciate and thus purchase. Many electronic commerce sites already benefit from novel opportunities of personalized marketing leverage offered by these information systems [26].

Recommender systems learn from customers and recommend products they are expected to find most valuable from among all available goods. Hereby, common approaches are classified into two major categories, namely content-based and collaborative filtering. Purely content-based filtering systems compute personalized recommendations by comparing content representations of previously liked items with content descriptions of goods still unknown to the active user. Many of its ideas stem from information retrieval techniques.

Collaborative filtering works by collecting ratings about products pertaining to some given domain and matching together people with similar interests. Hereby, interest similarity implies having rated many items in common and having assigned similar ratings to each of them. Its huge advantage over content-based filtering lies in its ability to operate in environments where the extraction of relevant features cannot be accomplished easily by automated processes. For instance, Jester [9] recommends jokes to its users.

Hybrid approaches exploit both content-based and collaborative filtering facilities.

However, most systems of either type only work effectively when situated in those environments where information density is high [24], i.e., large numbers of users voting for small numbers of items and issuing large numbers of *explicit* ratings each. Small, decentralized and open Web communities, where ratings are mainly derived implicitly from user behavior and interaction patterns, therefore poorly qualify for blessings provided by recommender systems.

We propose a novel, hybrid filtering approach that exploits bulk taxonomic information designed for exact product classification. These large, domain-dependent corpora are made available through diverse electronic commerce sites and standardization organizations. For instance, Amazon.com provides comprehensive and detailed classification information for books published in most common languages, like English, Spanish, French, and German, relating content to some fine-grained taxonomy of more than 13,500 hierarchically arranged topics.

Our approach permits properly inferring profile similarity between two given users even when both agents do not have any products rated in common. Making use of the "collaboration via content" paradigm [20], high quality recom-

CIKM'04, November 8–13, 2004, Washington, D.C., USA.
Copyright 2004 ACM 1-58113-874-1/04/0011 ...$5.00.

mendations become feasible in communities suffering from information sparsity, too. Hereby, besides taxonomy-driven profile generation, topic diversification constitutes the second core contribution of our work.

We mined data from one such community, All Consuming (*http://www.allconsuming.net*), and conducted various experiments demonstrating the superior performance over benchmark approaches.

2. RELATED WORK

Recommender systems started attracting major research interest during the early nineties [8]. Resnick et al. [21] introduced Pearson correlation, still largely in use, to compute similarity between users of their GroupLens news recommender system. Along with Ringo [27], the latter system counts among the first "classical" collaborative filtering systems.

Purely content-based recommender systems are less common. Notable sample approaches are described by Middleton [16], Alspector [1], Ferman [7], and Mukherjee [18]. The effectiveness of the content-based information filtering paradigm has been proven for applications locating textual documents relevant to a topic, using techniques like vector-space queries.

Balabanović's Fab [3] counts among the first hybrid systems, which are becoming increasingly popular today. More recent approaches are depicted in [14], proposing a graph-based recommender system, and [15]. Ontological user profiling was explored by Middleton [17]. To the best of our knowledge, this approach is the only one bearing traits similar to taxonomy-driven recommendation generation. However, Middleton uses clustering techniques for categorization and does not exploit hand-crafted, large-scale product classification taxonomies.

Appropriate evaluation methods for measuring the performance of recommender systems are still in their infancy and subject to ample discussion. Herlocker et al. [13, 11] and Breese [5] offer in-depth information about diverse "in vitro" evaluation frameworks for recommender systems. Cosley [6] proposes an open framework for online, "in situ" benchmarking and comparison of filtering performance, positing that "accuracy does not tell the whole story".

3. PROPOSED APPROACH

Sticking to the "collaboration via content" paradigm [20], our approach computes content-based user profiles which are then used to discover like-minded peers. Once the active[1] agent's neighborhood of most similar peers has been formed, the recommender focuses on products rated by those neighbors and generates top-N recommendation lists. The rank assigned to a product hereby depends on the proximity of agents voting for the latter, and its content description with respect to the active user's interest profile. Hence the hybrid nature of our approach.

3.1 Information Model

Before delving into algorithmic details, we introduce the formal information model supposed:

- **Set of agents** $A = \{a_1, a_2, \ldots, a_n\}$. Set A contains all users part of the community.
- **Set of products** $B = \{b_1, b_2, \ldots, b_m\}$. All domain-relevant products are comprised in set B. Hereby, unique identifiers may refer to proprietary product codes from an online store, such as Amazon.com's ASINs, or represent globally accepted standard codes, like ISBNs.
- **User ratings** $R_1, R_2, \ldots, R_n$. Every agent a_i is assigned a set $R_i \subseteq B$ which contains its implicit product ratings. Implicit ratings, such as purchase data, product mentions, etc., are far more common in electronic commerce systems and online communities than explicit ratings [2], but more difficult to cope with when trying to compute personalized recommendations [19].
- **Taxonomy C over set $D = \{d_1, d_2, \ldots, d_l\}$.** Set D contains categories for product classification. Each category $d_e \in D$ represents one specific topic that products $b_k \in B$ may fall into. Topics express broad or narrow categories. The partial taxonomic order $C : D \rightarrow 2^D$ retrieves all immediate sub-categories $C(d_e) \subseteq D$ for topics $d_e \in D$. Hereby, we require that $C(d_e) \cap C(d_h) = \emptyset$ holds for all $d_e, d_h \in D, e \neq h$, hence imposing tree-like structuring, similar to single-inheritance class hierarchies known from object-oriented languages. Leaf topics d_e are topics with zero outdegree, formally $C(d_e) = \perp$, i.e., most specific categories. Furthermore, taxonomy C has exactly one top element $\top$, which represents the most general topic and has zero indegree.
- **Descriptor assignment function $f : B \rightarrow 2^D$.** Function f assigns a set $D_k \subseteq D$ of product topics to every product $b_k \in B$. Note that products may possess *several* descriptors, for classification into one single category generally entails loss of precision.

3.2 Taxonomy-driven Profile Generation

The computation of user profiles by exploiting taxonomies as powerful background knowledge represents our recommender system's most important cornerstone. Its applicability especially addresses very large product sets, e.g., the set of all published English books, etc.

Common collaborative filtering techniques represent user profiles by vectors $\vec{v}_i \in \mathbb{R}^{|B|}$, where v_{i_k} indicates the user's rating for product $b_k \in B$. Similarity between agents a_i and a_j is computed by applying Pearson correlation [27, 21] to their respective profile vectors. Clearly, for very large $|B|$ and comparatively small $|A|$, this representation fails by virtue of insufficient overlap of rating vectors. Even more advanced approaches, e.g., Sarwar's singular value decomposition [24], cannot reduce dimensionality satisfactorily for suchlike domains.

We propose another, more informed approach which does not represent users by their respective *product*-rating vectors of dimensionality $|B|$, but by vectors of interest scores assigned to *topics* taken from taxonomy C over product categories $d \in D$.

User profile vectors are thus made up of $|D|$ entries, which corresponds to the number of distinct classification topics. Moreover, making use of profile vectors representing interest in *topics* rather than product *instances*, we can exploit the hierarchical structure of taxonomy C in order to generate overlap and render the similarity computation more meaningful:

[1] The term *active* identifies the agent for which to perform recommendation computation.

for every topic $d_{k_e} \in f(b_k)$ of products b_k that agent a_i has implicitly rated, we also infer an interest score for all *super*-topics of d_{k_e} in user a_i's profile vector. However, score assigned to super-topics decays with increasing distance from leaf node d_{k_e}. We furthermore normalize profile vectors with respect to the amount of score assigned, according the arbitrarily fixed overall score s.

Hence, suppose that $\vec{v}_i = (v_{i_1}, v_{i_2}, \ldots, v_{i_{|D|}})^T$ represents the profile vector for user a_i, where v_{i_k} gives the score for topic $d_k \in D$. Then we require the following equation to hold:

$$\forall a_i \in A : \sum_{k=1}^{|D|} v_{i_k} = s \qquad (1)$$

By virtue of agent-wise normalization for a_i's profile, score for each product $b_k \in R_i$ amounts to $s \,/\, |R_i|$, inversely proportional to the number of distinct products that a_i has rated. Likewise, for each topic descriptor $d_{k_e} \in f(b_k)$ categorizing product b_k, we accord topic score $\mathrm{sc}(d_{k_e}) = s \,/\, (|R_i| \cdot |f(b_k)|)$. Hence, topic score for b_k is distributed evenly among its topic descriptors.

Let $(p_0, p_1, \ldots, p_q)$ denote the path from top element $p_0 = \top$ to descendant $p_q = d_{k_e}$ within the tree-structured taxonomy C for some given $d_{k_e} \in f(b_k)$. Hence, topic descriptor d_{k_e} has q super-topics. Score normalization and inference of fractional interest for super-topics imply that descriptor topic score $\mathrm{sc}(d_{k_e})$ may *not* become *fully* assigned to d_{k_e}, but in part to all its ancestors $p_{q-1}, \ldots p_0$, likewise. We therefore introduce another score function $\mathrm{sco}(p_m)$ that represents the eventual assignment of score to topics p_m along the taxonomy path leading from $p_q = d_{k_e}$ to p_0:

$$\sum_{m=0}^{q} \mathrm{sco}(p_m) = \mathrm{sc}(d_{k_e}) \qquad (2)$$

In addition, we require that interest score $\mathrm{sco}(p_m)$ accorded to p_m, which is super-topic to p_{m+1}, depends on the number of siblings, denoted $\mathrm{sib}(p_{m+1})$, of p_{m+1}. The less siblings p_{m+1} possesses, the more interest score is accorded to its super-topic node p_m:

$$\mathrm{sco}(p_m) = \kappa \cdot \frac{\mathrm{sco}(p_{m+1})}{\mathrm{sib}(p_{m+1}) + 1} \qquad (3)$$

We hereby assume that sub-topics have *equal shares* in their super-topic within taxonomy C. Clearly, this assumption may imply several issues and raise concerns, e.g., when certain sub-taxonomies are considerably denser than others [23].

Propagation factor κ permits fine-tuning for the profile generation process, depending on the underlying taxonomy's depth and granularity. For instance, we apply $\kappa = 0.75$ for Amazon.com's book taxonomy.

Computed scores $\mathrm{sco}(p_m)$ are used to build a profile vector $\vec{v}_i$ of user a_i, adding scores for topics in $\vec{v}_i$. The procedure is repeated for every product $b_k \in R_i$ and every $d_{k_e} \in f(b_k)$.

Example 1 (Profile assembly) Suppose taxonomy C as depicted in Figure 1, and propagation factor $\kappa = 1$. Let a_i have implicitly rated four books, namely Matrix Analysis, Fermat's Enigma, Snow Crash, and Neuromancer. For Matrix Analysis, five topic descriptors are given, one of them pointing to leaf topic Algebra within our small taxonomy.

Suppose that $s = 1000$ defines the overall accorded profile score. Then the score assigned to descriptor Algebra amounts to $s \,/\, (4 \cdot 5) = 50$. Ancestors of leaf Algebra are Pure, Mathematics, Science, and top element Books. Score 50 hence must be distributed among these topics according to Equation 2 and 3. Result computation yields score 29.087 for topic Algebra. Likewise, applying Equation 3, we get 14.543 for topic Pure, 4.848 for Mathematics, 1.212 for Science, and 0.303 for top element Books. These values are then used to build profile vector $\vec{v}_i$ of a_i.

3.3 Neighborhood Formation

Taxonomy-driven profile generation computes flat profile vectors $\vec{v}_i \in [0, s]^{|D|}$ for agents a_i, assigning score values between 0 and maximum score s to every topic d from the set of product categories D. In order to generate neighborhoods of like-minded peers for the active user a_i, a proximity measure is required.

3.3.1 Measuring Proximity

Sarwar names Pearson correlation [27, 21] and cosine similarity, widely known from information retrieval, as most popular approaches for measuring profile similarity. We have opted for Pearson correlation for its ability to discover *negative* correlation, too, which is not the case for cosine similarity.

For users a_i and a_j with profiles $\vec{v}_i$ and $\vec{v}_j \in [0, s]^{|D|}$, respectively, Pearson correlation is defined as below:

$$c(a_i, a_j) = \frac{\sum_{k=0}^{|D|} (v_{i_k} - \overline{v_i}) \cdot (v_{j_k} - \overline{v_j})}{\sqrt{\sum_{k=0}^{|D|} (v_{i_k} - \overline{v_i})^2 \cdot \sum_{k=0}^{|D|} (v_{j_k} - \overline{v_j})^2}} \qquad (4)$$

Hereby, $\overline{v_i}$ and $\overline{v_j}$ give mean values for vectors $\vec{v}_i$ and $\vec{v}_j$. In our case, because of profile score normalization, both are identical, i.e., $\overline{v_i} = \overline{v_j} = s \,/\, |D|$. Values for $c(a_i, a_j)$ range from -1 to $+1$, where negative values indicate negative correlation, and positive values positive correlation, respectively.

Clearly, people who have implicitly rated many products in common also have high similarity. For generic collaborative filtering approaches, the proposition's inversion also holds, i.e., people who have *not* rated many products in common have *low* similarity.

On the other hand, applying taxonomy-driven profile generation, high similarity values can be derived even for pairs of agents that have little or even no products in common. Common sense hereby tells that the measure's quality substantially depends on the taxonomy's design and level of nesting. According to our scheme, the more score two profiles $\vec{v}_i$ and $\vec{v}_j$ have accumulated in same branches, the higher their measured similarity.

Example 2 (Interest correlation) Suppose the active user a_i has rated only one single book b_m, bearing exactly one topic descriptor that classifies b_m into Algebra. User a_j has read a different book b_n whose topic descriptors point to diverse leaf nodes[2] of History, denoting history of mathematics. Then $c(a_i, a_j)$ will still be reasonably high, for both profiles have significant overlap in categories Mathematics and Science.

Negative correlation occurs when users have completely diverging interests. For instance, in our information base

[2] Leaf nodes of History are not shown in Figure 1.

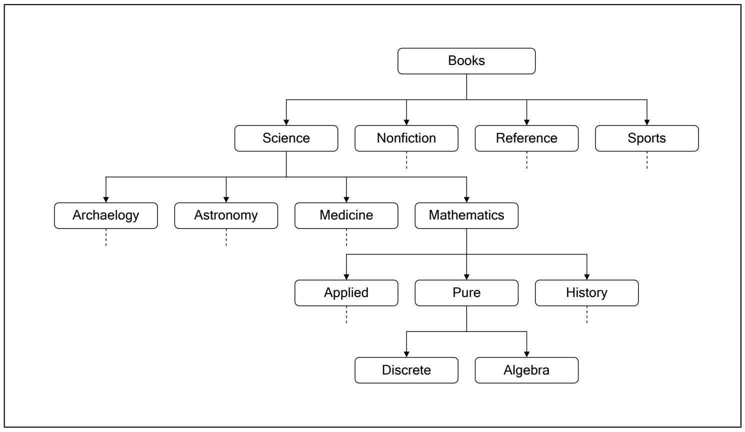

Figure 1: Fragment from the Amazon book taxonomy

mined from All Consuming, we had one user reading books mainly from the genres of Sci-Fi, Fantasy, and Artificial Intelligence. The person in question was negatively correlated to another one reading books about American History, Politics, and Conspiracy Theories.

3.3.2 Selecting Neighbors

Neighborhood formation is followed by computing proximity weights $c(a_i, a_j)$ for the active user a_i and agents $a_j \in A \setminus \{a_i\}$. Agent a_i's neighborhood, denoted by $\text{clique}(a_i)$, hereby contains most similar peers for use in computing recommendation lists [27].

Herlocker [11] names two techniques for neighborhood selection, namely correlation-thresholding and best-M-neighbors. Correlation-thresholding puts users a_j with similarities $c(a_i, a_j)$ above some given threshold t into $\text{clique}(a_i)$, whereas best-M-neighbors picks the M best correlates for a_i's neighborhood.

We opted for best-M-neighbors, since correlation-thresholding implies diverse unwanted effects when sparsity prevails [11].

3.4 Recommendation Generation

Candidate products for a_i's personalized recommendation list are taken from its neighborhood's implicit ratings, avoiding products that a_i already knows:

$$B_i = \bigcup \{R_j \mid a_j \in \text{clique}(a_i)\} \setminus R_i \tag{5}$$

Candidate products $b_k \in B_i$ are then weighted according to their *relevance* for a_i. Hereby, the relevance of products $b_k \in B_i$ for a_i, denoted $w_i(b_k)$, depends on various factors. Most important, however, are two aspects:

- **User proximity.** Similarity measures $c(a_i, a_j)$ of all those agents a_j that "recommend" product b_k to the active agent a_i are of special concern. The closer these agents to a_i's interest profile, the higher the relevance of b_k for a_i. We borrowed the latter intuition from common collaborative filtering techniques [12].
- **Product proximity.** Second, measures $c_b(a_i, b_k)$ of product b_k's closeness with respect to a_i's interest profile are equally significant. Being purely content-based, this measure supplements the overall recommendation generation process with more fine-grained filtering facilities: mind that even highly correlating agents may appreciate items beyond the active user's specific interests. Otherwise, these agents would have *identical* interest profiles, not just similar ones.

  The computation of $c_b(a_i, b_k)$ follows from user similarity detection. For this purpose, we create a "dummy" user a_θ with $R_\theta = \{b_k\}$ and define $c_b(a_i, b_k) := c(a_i, a_\theta)$.

Relevance $w_i(b_k)$ of product b_k for the active user a_i is then defined as follows:

$$w_i(b_k) = \frac{q \cdot c_b(a_i, b_k) \cdot \sum_{a_j \in A_i(b_k)} c(a_i, a_j)}{|A_i(b_k)| + \Upsilon_R}, \tag{6}$$

where

$$A_i(b_k) = \{a_j \in \text{clique}(a_i) \mid b_k \in R_j\}$$

and

$$q = (1.0 + |f(b_k)| \cdot \Gamma_T)$$

Hereby, variables Υ_R and Γ_T represent fine-tuning parameters that allow for customizing the recommendation process. Parameter Υ_R penalizes products occurring infrequently in rating profiles of neighbors $a_j \in \text{clique}(a_i)$. Hence, large Υ_R makes popular items acquire higher relevance weight, which may be suitable for users wishing to be recommended well-approved and common products instead of rarities. On the other hand, low Υ_R treats popular and uncommon, new products in exactly the same manner, helping to alleviate the latency problem [29]. For experimental analysis, we tried values between 0 and 2.5.

Parameter Γ_T rewards products b_k with extensive content descriptions, i.e., large $|f(b_k)|$. Variable Γ_T proves useful because profile score normalization and super-topic score inference may penalize products b_k containing several, detailed descriptors $d \in f(b_k)$, and favor products having few, more general topic descriptors indicating their content. Reward through Γ_T is assigned linearly by virtue of $(|f(b_k)| \cdot \Gamma_T)$. The implementation of exponential decay appears likewise reasonable, therefore reducing Υ_R's gain in influence when $|f(b_k)|$ becomes larger. However, we have not tried this extension yet.

Eventually, product relevance weights $w_i(b_k)$ computed for every $b_k \in B_i$ are used to produce the active user a_i's recommendation list. The injective function $P_{w_i} : \{1, 2, \ldots, |B_i|\} \rightarrow B$ reflects recommendation ranking in *descending* order, i.e., $P_{w_i}(1) = b_h \Rightarrow \forall b_k \in B_i : w_i(b_h) \geq w_i(b_k)$. For top-$N$ recommendations, all entries $P_{w_i}(k), k > N$ are discarded.

3.5 Topic Diversification

An approach we call "topic diversification" constitutes another major contribution of our work. This technique represents an *optional* procedure to supplement recommendation generation and to enhance the computed list's utility for agent a_i.

To our best knowledge, no similar approaches exist or have been documented in literature affiliated with recommender systems. The underlying idea of topic diversification hereby refers to providing an active user a_i with recommendations from *all* major topics that a_i has declared specific interest in. The following example intends to motivate our method:

Example 3 (Topic overfitting) Suppose that a_i's profile contains books from Medieval Romance, Industrial Design, and Travel. Suppose Medieval Romance has a 60% share in a_i's profile, Industrial Design and Travel have 20% each. Consequently, Medieval Romance's predominance will result in most recommendations originating from this super-category, giving way for Industrial Design and Travel not before all books from like-minded neighbors fitting well into the Medieval Romance shape have been inserted into a_i's recommendations.

We observe the above issue with many recommender systems relying upon content-based and hybrid filtering facilities. For purely collaborative approaches, recommendation diversification according to the active user a_i's topics of interest becomes even less controllable. Remember that collaborative filtering does *not* consider the content of products but only ratings assigned. Hence, diversification and collaborative filtering intrinsically exclude each other.

3.5.1 Recommendation Dependency

In order to implement topic diversification, we assume that recommended products $P_{w_i}(o)$ and $P_{w_i}(p)$, $o, p \in \mathbb{N}$, along with their content descriptions, effectively *do* exert an impact on each other, which is commonly ignored by existing approaches: usually, only relevance weight ordering $o < p \Rightarrow w_i(P_{w_i}(o)) \geq w_i(P_{w_i}(p))$ must hold for recommendation list items, no other dependencies are assumed.

To our best knowledge, Brafman et al. [4] count among the only researchers recognizing the dependence between recommendations. Their approach considers recommendation generation as inherently *sequential* and uses Markov Decision Processes (MDP) in order to model interdependencies between recommendations. However, apart from the idea of dependence between items $P_{w_i}(o)$, $P_{w_i}(p)$, Brafman's focus significantly differs from our own, emphasizing the economic *utility* of recommendations with respect to past and future purchases.

In case of our topic diversification technique, recommendation interdependence signifies that an item b's current "dissimilarity" with respect to preceding recommendations plays an important role and may influence the "new" ranking order. Algorithm 1 depicts the entire procedure, a brief textual sketch is given in the next few paragraphs.

3.5.2 Topic Diversification Algorithm

Function P_{w_i*} denotes the new recommendation list, resulting from applying topic diversification. For every list entry $z \in [2, N]$, we collect those products b from the candidate products set B_i that do not occur in positions $o < z$ in P_{w_i*} and compute their similarity with set $\{P_{w_i*}(k) \mid k \in [1, z[\,\}$, which contains all new recommendations preceding rank z. We hereby compute this similarity, denoted $c^*(b)$, by applying our scheme for taxonomy-driven profile generation and proximity measuring presented in sections 3.2 and 3.3.1.

```
procedure diversify ($P_{w_i}$, $\Theta_F$) {
  $B_i \leftarrow \{P_{w_i}(k) \mid k \in [1, N]\,\}$; $P_{w_i*}(1) \leftarrow P_{w_i}(1)$;
  for $z \leftarrow 2$ to $N$ do
    set $B'_i \leftarrow B_i \setminus \{P_{w_i*}(k) \mid k \in [1, z[\,\}$;
    $\forall b \in B'$: compute $c^*(b, \{P_{w_i*}(k) \mid k \in [1, z[\,\})$;
    compute $P_{c^*} : \{1, 2, \ldots, |B'_i|\} \rightarrow B'_i$ using $c^*$;
    for all $b \in B'_i$ do
      $P_{c^*}^{\mathrm{rev}^{-1}}(b) \leftarrow |B'_i| - P_{c^*}^{-1}(b)$;
      $w_i^*(b) \leftarrow P_{w_i}^{-1}(b) \cdot (1 - \Theta_F) + P_{c^*}^{\mathrm{rev}^{-1}}(b) \cdot \Theta_F$;
    end do
    $P_{w_i*}(z) \leftarrow \min\{w_i^*(b) \mid b \in B'_i\}$;
  end do
  return $P_{w_i*}$;
}
```

Algorithm 1: Sequential topic diversification

Sorting all products b according to $c^*(b)$ in reverse order, we hence obtain dissimilarity rank $P_{c^*}^{\mathrm{rev}}$. This rank is then merged with the original recommendation rank P_{w_i} according to diversification factor Θ_F, yielding final rank P_{w_i*}. Factor Θ_F defines the impact that dissimilarity rank $P_{c^*}^{\mathrm{rev}}$ exerts on the eventual overall output. Large $\Theta_F \in [0.5, 1]$ favors diversification over a_i's original relevance order, while low $\Theta_F \in [0, 0.5[$ produces recommendation lists closer to the original rank P_{w_i}. For experimental analysis, we used parameterizations $\Theta_F \in [0.2, 0.4]$.

The effect of dissimilarity bears traits similar to that of "osmotic pressure" known from molecular biology [30]: steady insertion of products taken from one specific area of interest into the recommendation list increases the "pressure" for items from other domains. When pressure gets sufficiently high for one of these domains d, its best products b may "diffuse" into the recommendation list, even though their original rank $P_{w_i}^{-1}(b)$ might be inferior to candidates from the prevailing domain. Consequently, pressure for d decreases, paving the way for another domain whose pressure is about to reach its peak.

4. EXPERIMENTS AND EVALUATION

The following sections present empirical results obtained from evaluating our approach. Core engine parts of our system, along with vaious other tools for data collection and screen scraping, were implemented in Java, small portions in Perl. PHP frontends enable remote access via Web interfaces.

Besides our own, taxonomy-based approach, we also implemented three other recommender algorithms for comparison.

4.1 Data Acquisition

Experimentation, parameterization and algorithmic finetuning were conducted on "real-world" data, obtained from All Consuming[3], an open community addressing people interested in reading books. We extracted additional, taxonomic background knowledge, along with content descriptions of those books, from Amazon.com.

The entire dataset comprises $2,783$ users, representing either "real", registered members of All Consuming or personal weblogs collected by the community's crawlers, and $14,591$ ratings addressing $9,237$ diverse book titles. All ratings are implicit, i.e., non-quantifiable with respect to the extent of appreciation of respective books. On average, users provided 5.24 book ratings.

Amazon.com's book classification taxonomy, which is tree-structured and thus limited to "single inheritance" of concepts, contained $13,525$ distinct topics after application of various data cleansing procedures and duplicate removal. Moreover, our crawling tools collected $27,202$ topic descriptors from Amazon.com, relating $8,641$ books to this taxonomy. Consequently, for 596 of those $9,237$ books mentioned by All Consuming's users, no content information was obtained from Amazon.com, signifying only 6.45% defection. We eliminated these books from our dataset.

On average, 3.15 topic descriptors were found for books available on Amazon.com, thus making content descriptions sufficiently explicit and reliable for profile generation.

To make the analysis data obtained from our performance trials more accurate, we relied upon an external Web-service[4] to spot ISBNs referring to the same book, but different editions, e.g., hardcover and paperback. Those ISBNs were then mapped to one single representative ISBN.

4.2 Evaluation Framework

Evaluation methods for recommender systems are manifold, comprising statistical techniques to measure deviations of *predicted* and *actual* rating values [27], like MAE and ROC metrics [13], and approaches to estimate the *utility* of the recommendation list for the active user, e.g., precision and recall known from information retrieval, and Breese score [5], likewise. Since the prediction of product ratings only makes sense when dealing with *explicit* ratings, we have committed ourselves to the latter option, i.e., evaluating the quality of the generated recommendation lists.

4.2.1 Benchmark Systems

Besides our own, taxonomy-driven proposal, we implemented three other recommender algorithms: one "naive", random-based system offering no personalization at all and defining the bottom line, one purely collaborative approach, typically used for evaluations, and one hybrid method, exploiting content information provided by our dataset.

4.2.1.1 Bottom Line Definition.

For any given user a_i, the system randomly selects an item $b \in B \setminus R_i$ for $a_i^{\prime}s$ top-N list $P_i : \{1, 2, \ldots, N\} \rightarrow B$. Clearly, as is the case for every other presented approach, products may not occur more than once in the recommendation list, i.e., $\forall o, p \in \{1, 2, \ldots, N\}, o \neq p : P_i(o) \neq P_i(p)$ holds.

The random-based approach shows results obtained when no filtering takes place, constituting the base case that "non-naive" algorithms are bound to surpass.

4.2.1.2 Collaborative Filtering Algorithm.

The GroupLens project [21] first introduced an automated, purely collaborative system using a neighborhood-based algorithm, which commonly serves as baseline benchmarking system for evaluation purposes today.

The original GroupLens system used Pearson correlation to weight the similarity between the active user a_i and all other agents $a_j \in A \setminus \{a_i\}$, selected best-$M$ neighbors to form a_i's neighborhood $\text{clique}(a_i)$, and computed a final prediction by performing a weighted average of deviations from the neighbor's mean. Since the algorithm only works for scenarios featuring *explicit* ratings, Sarwar [24] proposed an adaptation known as "most frequent items".

We adopted Sarwar's version which computes relevance weights $w_i(b_k)$ for books b_k from $a_i^{\prime}s$ candidates set B_i according to the following scheme. Assume that $A_i(b_k) \subseteq \text{clique}(a_i)$ contains all neighbors of a_i who have implicitly rated b_k:

$$w_i(b_k) = \sum_{a_j \in A_i(b_k)} c(a_i, a_j) \tag{7}$$

Hereby, we measure user similarity $c(a_i, a_j)$ according to Pearson correlation, introduced in Section 3.3.1. Profile vectors $\vec{v}_i, \vec{v}_j$ for agents a_i, a_j, respectively, represent implicit ratings for every product $b_k \in B$, hence $\vec{v}_i, \vec{v}_j \in \{0, 1\}^{|B|}$.

4.2.1.3 Hybrid Recommender Approach.

The third competing system exploits collaborative as well as content-based filtering facilities, hence its hybrid nature. The algorithmic clockwork mimics Pazzani's "collaboration via content" proposal [20], representing user profiles $\vec{v}_i$ by collections of descriptive terms, along with their frequency of occurrence.

Hereby, descriptive terms for books b_k correspond to topic descriptors $f(b_k)$, originally relating book content to taxonomy C over categories D. Profile vectors $\vec{v}_i \in \mathbb{N}^{|D|}$ for agents a_i thus take the following shape:

$$\forall d \in D : v_{i_d} = |\{b_k \in R_i \mid d \in f(b_k)\}| \tag{8}$$

For neighborhood formation, standard Pearson correlation is applied to these content-driven profile vectors. Relevance is then defined as below:

$$w_i(b_k) = \frac{c_b(a_i, b_k) \cdot \sum_{a_j \in A_i(b_k)} c(a_i, a_j)}{|A_i(b_k)|} \tag{9}$$

Mind that Equation 9 presents a special case of Equation 6, assuming $\Gamma_T = 0$ and $\Upsilon_R = 0$. Essentially, the depicted hybrid approach constitutes a simplistic adaptation of our taxonomy-driven system. Notable differences pertain to the hybrid filtering algorithm's lack of super-topic score inference, one major cornerstone of our novel method. Furthermore, the simplisitc version lacks extensive parameterization and topic diversification.

[3] All Consuming is reachable via *http://www.allconsuming.net*.

[4] See *http://www.oclc.org/research/projects/xisbn/*.

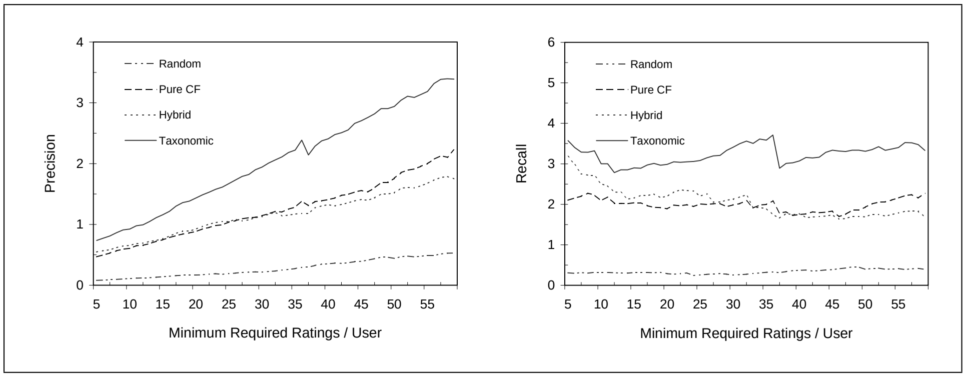

Figure 2: Unweighted precision and recall metrics

4.2.2 Experiment Setup

The evaluation framework we established intends to compare the "utility" of recommendation lists generated by all four recommender systems. Measurement is achieved by applying metrics well-known from information retrieval, i.e., precision and recall, implemented according to Sarwar's proposal [24], and Breese's half-life utility metric [13], known as Breese score [5] or weighted recall.

Hereby, we borrowed various ideas from machine learning cross-validation methods. First, we selected all users a_i with more than five ratings and discarded those having less, owing to the fact that reasonable recommendations are beyond feasibility for these cases.

Next, we applied K-folding, dividing every user $a_i's$ implicit ratings R_i into $K = 5$ disjoint "slices" of preferably equal size. Hereby, four randomly chosen slices constitute agent a_i's *training set* R_i^x, thus containing approximately 80% of implicit ratings $b \in R_i$. These ratings then define a_i's profile from which final recommendations are computed. For recommendation generation, a_i's residual slice $(R_i \setminus R_i^x)$ is retained and not used for prediction. This slice, denoted T_i^x, contains about 20% of a_i's ratings and constitutes the *test set*, i.e., those products the recommendation algorithms intend to "guess".

For our experiments, we considered *all five* combinations $(R_i^x, T_i^x), 1 \leq x \leq 5$ of user a_i's slices, hence computing five complete recommendation lists for every a_i that suffices the before-mentioned criteria, i.e., exactly those a_i having implicitly rated at least five books.

4.2.3 Parameterization

The size for neighborhood formation was set to $M = 20$, i.e., $|\,\text{clique}(a_i)| \leq 20$, and we provided top-20 recommendations for each active user's training set R_i^x. Similarity between profiles, based upon R_i^x and the original ratings R_j of all other agents a_j, was hereby computed anew for each training set R_i^x of a_i.

For performance trial purposes, we parameterized our taxonomy-driven recommender system's profile generation process by assuming propagation factor $\kappa = 0.75$, which encourages super-topic score inference. We opted for $\kappa < 1$ since Amazon.com's book taxonomy is deeply-nested and topics tend to have numerous siblings, which makes it rather difficult for topic score to reach higher levels.

For recommendation generation, we set parameter $\Upsilon_R = 0.25$, i.e., books occurring infrequently in ratings issued by the active user's neighbors were therefore not overly penalized. Generous reward was accorded for books b bearing highly explicit content descriptions, i.e., having large $|f(b)|$, by assuming $\Gamma_T = 0.1$. Hence, a 10% bonus was granted for every additional topic descriptor. For topic diversification, we adopted $\Theta_F = 0.33$.

No parameterizations were required for the random-based, purely collaborative, and hybrid approaches.

4.2.4 Evaluation Metrics

After computing top-20 lists $P_i^x : \{1, 2, \ldots, 20\} \rightarrow B$ for combinations (R_i^x, T_i^x), the actual evaluation of P_i^x's quality took place.

We adopted evaluation measures similar to precision and recall known from information retrieval. Remember that for some given number of returned items, recall indicates the percentage of relevant items that were returned, and precision gives the percentage of returned items that are relevant.

Sarwar [24] presents some adapted variant of recall, recording the percentage of test set products $b \in T_i^x$ occurring in recommendation list P_i^x with respect to the overall number of test set products $|T_i^x|$:[5]

$$\text{Recall} = 100 \cdot \frac{|T_i^x \cap \Im P_i^x|}{|T_i^x|} \tag{10}$$

Accordingly, precision represents the percentage of test set products $b \in T_i^x$ occurring in P_i^x with respect to the size of the recommendation list:

$$\text{Precision} = 100 \cdot \frac{|T_i^x \cap \Im P_i^x|}{|\Im P_i^x|} \tag{11}$$

Breese [5] further refines Sarwar's adaptation of recall by introducing *weighted* recall, or Breese score. Breese hereby proposes that the expected utility of a recommendation list is simply the *probability* of viewing a recommended product that is actually relevant, i.e., taken from the test set, times its utility, which is either 0 or 1 for implicit ratings.

Breese furthermore posits that each successive item in a list is less likely to be viewed by the active user with exponential

[5] Symbol $\Im P_i^x$ denotes the *image* of map P_i^x, i.e., all books part of the recommendation list.

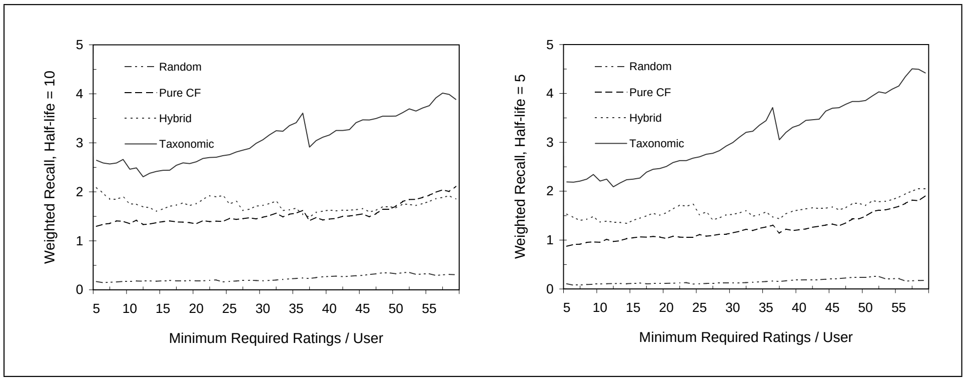

Figure 3: Weighted recall, using half-life $\alpha = 10$ and $\alpha = 5$

decay. The expected utility of a ranked list P_i^x of products is as follows:

$$H(P_i^x, T_i^x) = \sum_{b \, \in \, (T_i^x \, \cap \, \Im P_i^x)} \frac{1}{2^{(P_i^{x^{-1}}(b)-1)/(\alpha-1)}} \tag{12}$$

Parameter α denotes the viewing half-life. Half-life is the number of the product on the list such that there is a 50% chance that the active agent, represented by training set R_i^x, will review that product. Finally, the weighted recall of P_i^x with respect to T_i^x is defined as below:

$$\text{BScore}(P_i^x, T_i^x) = 100 \cdot \frac{H(P_i^x, T_i^x)}{\sum_{k=1}^{|T_i^x|} \frac{1}{2^{(k-1)/(\alpha-1)}}} \tag{13}$$

Interestingly, when assuming $\alpha = \infty$, Breese score is identical to Sarwar's definition of recall.

In order to obtain "global" metrics, i.e., precision, recall, and Breese score for the entire system and not only one single agent, we averaged the respective metric values for all evaluated users.

4.2.5 Result Analysis

Performance was mechanically measured by computing unweighted precision and recall according to Sarwar's definition, and Breese's weighted recall, first assuming half-life $\alpha = 5$, then $\alpha = 10$, for all four recommenders and all combinations of training sets and test sets. Results are displayed in Figure 2 and 3.

For each indicated chart, the horizontal axis expresses the *minimum number* of ratings that users were required to have issued so they were considered for recommendation generation and evaluation. Having discarded all users with less than five ratings during data preprocessing, our performance trials commence with all agents having at least five ratings. Note that larger x-coordinates hence imply that *less* agents were considered for computing the respective data points.

Results obtained seem to prove our hypothesis that taxonomy-driven recommendation generation outperforms common approaches when dealing with sparse product rating information. All four metrics position our novel approach significantly above its purely collaborative and hybrid counterparts.

Hereby, we observe one considerable cusp common to all four charts and particularly pronounced for the taxonomy-based curves. The sudden drop happens when users bearing exactly 36 implicit ratings become discarded. On average, for taxonomy-driven recommendation generation, these agents have considerably high ranks with respect to all four metrics applied. Removal thus temporarily lowers the respective curves.

More detailed, metric-specific analysis follows in subsequent paragraphs.

4.2.5.1 Precision.

Surprisingly, precision increases even for the random recommender when ignoring users with fewer ratings. The reason for this phenomenon lies in the nature of the precision metric: for users a_i with test sets T_i^x smaller than the number $|P_i^x|$ of recommendations received, i.e., $|T_i^x| < 20$, there is not even a chance of achieving 100% precision.

Analysis of unweighted precision, given on the left-hand side of Figure 2, shows that the gap between our taxonomy-driven approach and its collaborative and hybrid rivals becomes even larger when users are required to have numerous books rated. Agents with small numbers of ratings tend to interfere prediction accuracy as no proper "guidance" for neighborhood selection and interest definition can be provided.

Differences between the collaborative and the hybrid method are less significant and rather marginal. However, the first increasingly outperforms the former when making recommendations for agents with numerous ratings.

4.2.5.2 Unweighted and Weighted Recall.

Unweighted recall, shown on the right side of Figure 2, presents a slightly different scenario: even though the performance gap between taxonomy-driven recommender and both other, non-naive methods still persists, this gap does not become larger for increasing x. Collaborative filtering, slightly inferior to its hybrid pendant at first, overtakes the latter when considering agents with numerous ratings only. Similar observations have been made by Pazzani [20].

Figure 3 allows more fine-grained analysis with respect to the accuracy of rankings. Remember that unweighted recall is equivalent to Breese score when assuming half-life $\alpha = \infty$. While pure collaborative filtering shows largely insensitive to decreasing α, hybrid and taxonomy-driven recommenders do not. Assuming $\alpha = 10$, the first derivation of the latter two approaches improves over their corresponding recall curves for increasing x-coordinates. This notable development becomes even more obvious when further decreasing half-life to $\alpha = 5$.

Consequently, in case of content-exploiting methods, actually relevant products $b \in \Im P_i^x \cap T_i^x$ have the tendency to appear "earlier" in recommendation lists P_i^x, i.e., have comparatively small distance from the top rank. On the other hand, relevant products seem to be more evenly distributed among top-20 ranks for collaborative filtering.

5. DEPLOYMENT AND ONLINE STUDY

On February 9, 2004, our taxonomy-driven recommender system was deployed into the All Consuming community[6] and now computes personalized recommendations for registered users, based upon their book rating profile. Access facilities are offered through diverse PHP scripts that query an RDBMS containing rating profiles, neighborhood information, and precomputed recommendations, likewise.

5.1 Online User Satisfaction Study

Besides our taxonomy-driven approach, we furthermore implanted both other non-naive approaches documented before into All Consuming. Registered users hence may access three distinct lists of top-20 recommendations, customized according to their personal rating profile. We utilized the latter system setup to conduct online "in situ" performance comparisons, going beyond offline statistical measures. Offline evaluation methods are useful, though not able to measure *real* user satisfaction [10].

Online evaluations of recommender systems performance have already been made before by Swearingen and Sinha [28], comparing human perception, i.e., approval or disapproval, of recommendation lists provided by several popular recommenders. Studies were based on 19 people assessing six different commercial systems.

5.1.1 Evaluation Setup

For online evaluation, we demanded All Consuming members to rate all recommendations provided on a 5-point likert scale, ranging from -2 to $+2$. Hereby, raters were advised to give maximum score for recommended books they had already read, but not indicated in their reading profile. Moreover, users were given the opportunity to return an "overall" satisfaction verdict for each recommendation list. The additional rating served as an instrument to also reflect the make-up and quality of list composition. Consequently, members could provide 63 rating statements each.

5.1.2 Result Analysis

51 All Consuming members, not affiliated with our department and university, volunteered to participate in our study by August 29, 2004. They provided 2,041 ratings about recommendations they were offered, and 123 additional, overall list quality verdicts. Since not every user rated all 60 books recommended by our three diverse systems, we assumed neutral votes for recommended books not rated. Furthermore, in order not to bias users towards our taxonomy-driven approach, we assigned letters "A", "B", "C" to recommendation lists, not revealing any information about the algorithm operating behind the scenes.

While 48 users rated one or more recommendations computed according to the purely collaborative method, dubbed "A", 42 did so for the taxonomy-driven approach, labelled "B", and 39 for the simplistic hybrid algorithm. In a first experiment, depicted on the left side of Figure 4, we compared the overall recommendation list verdicts and average ratings of personalized top-20 recommendations for each rater and each recommender system. Results were averaged over all participating users. In both cases, the taxonomy-driven system performed best and the purely collaborative worst.

Second, we counted all those raters perceiving one specific system as best. Again, comparison was based upon the overall verdict and average recommendation rating, likewise. In order to guarantee fairness, we discarded users not having rated all three systems for each metric. The right chart of Figure 4 shows that the appreciation of the taxonomy-driven method significantly prevailed.

Eventually, we may conclude that results obtained from the online analysis back offline evaluation results. In both cases, our taxonomy-driven method has been shown to outperform benchmark systems.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented a novel, hybrid approach to automated recommendation making, based upon large-scale product classification taxonomies which are readily available for diverse domains today. Cornerstones of our approach are hereby the generation of profiles via inference of super-topic score and topic diversification.

Thorough "in vitro" performance trials were conducted on "real-world" data in order to demonstrate our algorithm's superiority over less informed approaches when rating information sparseness prevails. Moreover, we provided "in situ" online evaluation, asking All Consuming community members to rate diverse recommender systems.

Next steps include testing our method for domains other than books and analyzing the impact that taxonomic structure, nesting and average taxonomy depth may have on results obtained. Amazon.com offers an immense taxonomy for movie classification, too, containing more than 16,400 at the time of this writing. We envision performance trials running on top of the two well-known EachMovie and MovieLens [25] datasets. Hence, this analysis would also allow us to test the suitability of our approach for rather dense data.

7. REFERENCES

[1] ALSPECTOR, J., KOLCZ, A., AND KARUNANITHI, N. Comparing feature-based and clique-based user models for movie selection. In *Proceedings of The Third ACM Conference on Digital Libraries* (Pittsburgh, PE, USA, 1998), ACM Press, pp. 11–18.

[2] AVERY, C., AND ZECKHAUSER, R. Recommender systems for evaluating computer messages. *Communications of the ACM 40*, 3 (March 1997), 88–89.

[3] BALABANOVIĆ, M., AND SHOHAM, Y. Fab - content-based, collaborative recommendation. *Communications of the ACM 40*, 3 (March 1997), 66–72.

[4] BRAFMAN, R., HECKERMAN, D., AND SHANI, G. Recommendation as a stochastic sequential decision problem. In *Proceedings of ICAPS 2003* (Trento, Italy, 2003).

[5] BREESE, J., HECKERMAN, D., AND KADIE, C. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence* (Madison, WI, USA, July 1998), Morgan Kaufmann, pp. 43–52.

[6] Our recommenders are reachable through All Consuming's *News*-section, see *http://cgi.allconsuming.net/news.html*.

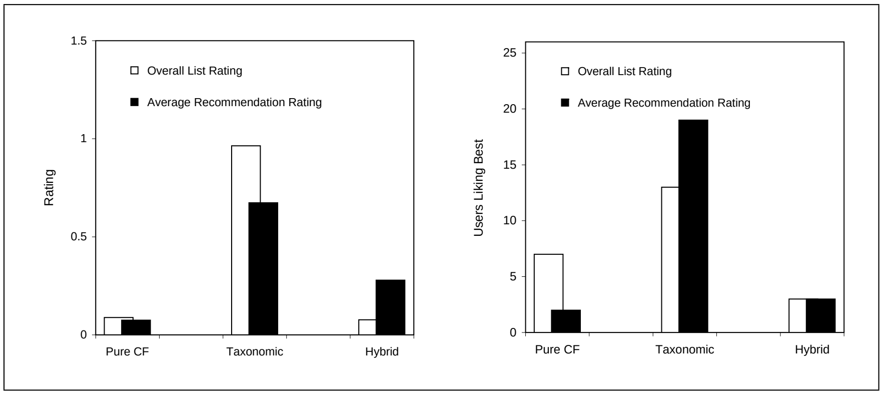

Figure 4: Results obtained from online evaluation

[6] COSLEY, D., LAWRENCE, S., AND PENNOCK, D. REFEREE: An open framework for practical testing of recommender systems using ResearchIndex. In *28th International Conference on Very Large Databases* (Hong Kong, China, August 2002), Morgan Kaufmann, pp. 35–46.

[7] FERMAN, M., ERRICO, J., VAN BEEK, P., AND SEZAN, I. Content-based filtering and personalization using structured metadata. In *Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries* (Portland, OR, USA, 2002), ACM Press, pp. 393–393.

[8] GOLDBERG, D., NICHOLS, D., OKI, B., AND TERRY, D. Using collaborative filtering to weave an information tapestry. *Communications of the ACM 35*, 12 (1992), 61–70.

[9] GOLDBERG, K., ROEDER, T., GUPTA, D., AND PERKINS, C. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval 4*, 2 (2001), 133–151.

[10] HAYES, C., MASSA, P., AVESANI, P., AND CUNNINGHAM, P. An online evaluation framework for recommender systems. In *Workshop on Personalization and Recommendation in E-Commerce* (Malaga, Spain, May 2002), Springer-Verlag.

[11] HERLOCKER, J., KONSTAN, J., BORCHERS, A., AND RIEDL, J. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Berkeley, CA, USA, 1999), ACM Press, pp. 230–237.

[12] HERLOCKER, J., KONSTAN, J., AND RIEDL, J. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work* (Philadelphia, PA, USA, 2000), pp. 241–250.

[13] HERLOCKER, J., KONSTAN, J., TERVEEN, L., AND RIEDL, J. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems 22*, 1 (2004), 5–53.

[14] HUANG, Z., CHUNG, W., ONG, T.-H., AND CHEN, H. A graph-based recommender system for digital library. In *Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries* (Portland, OR, USA, 2002), ACM Press, pp. 65–73.

[15] MELVILLE, P., MOONEY, R., AND NAGARAJAN, R. Content-boosted collaborative filtering for improved recommendations. In *Eighteenth National Conference on Artificial Intelligence* (Edmonton, Canada, 2002), American Association for Artificial Intelligence, pp. 187–192.

[16] MIDDLETON, S., ALANI, H., SHADBOLT, N., AND DE ROURE, D. Exploiting synergy between ontologies and recommender systems. In *Proceedings of the WWW2002 International Workshop on the Semantic Web* (Maui, HW, USA, May 2002), vol. 55 of *CEUR Workshop Proceedings*.

[17] MIDDLETON, S., SHADBOLT, N., AND DE ROURE, D. Capturing interest through inference and visualization: Ontological user profiling in recommender systems. In *Proceedings of the Third International Conference on Knowledge Capture* (Sanibel Island, FL, USA, September 2003), ACM Press, pp. 62–69.

[18] MUKHERJEE, R., DUTTA, P., AND SEN, S. MOVIES2GO - a new approach to online movie recommendation. In *Proceedings of the IJCAI Workshop on Intelligent Techniques for Web Personalization* (Seattle, WA, USA, August 2001).

[19] NICHOLS, D. Implicit rating and filtering. In *Proceedings of the Fifth DELOS Workshop on Filtering and Collaborative Filtering* (Budapest, Hungary, 1998), ERCIM, pp. 31–36.

[20] PAZZANI, M. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review 13*, 5-6 (1999), 393–408.

[21] RESNICK, P., IACOVOU, N., SUCHAK, M., BERGSTORM, P., AND RIEDL, J. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work* (Chapel Hill, NC, USA, 1994), ACM, pp. 175–186.

[22] RESNICK, P., AND VARIAN, H. Recommender systems. *Communications of the ACM 40*, 3 (1997), 56–58.

[23] RESNIK, P. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (Montreal, Canada, 1995), pp. 448–453.

[24] SARWAR, B., KARYPIS, G., KONSTAN, J., AND RIEDL, J. Application of dimensionality reduction in recommender systems - a case study. In *ACM WebKDD Workshop* (Boston, MA, USA, August 2000).

[25] SARWAR, B., KARYPIS, G., KONSTAN, J., AND RIEDL, J. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference* (Hong Kong, China, May 2001).

[26] SCHAFER, B., KONSTAN, J., AND RIEDL, J. Recommender systems in e-commerce. In *Proceedings of the First ACM Conference on Electronic Commerce* (Denver, CO, USA, 1999), ACM Press, pp. 158–166.

[27] SHARDANAND, U., AND MAES, P. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of the ACM CHI'95 Conference on Human Factors in Computing Systems* (1995), vol. 1, pp. 210–217.

[28] SINHA, R., AND SWEARINGEN, K. Comparing recommendations made by online systems and friends. In *Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries* (Dublin, Ireland, June 2001).

[29] SOLLENBORN, M., AND FUNK, P. Category-based filtering and user stereotype cases to reduce the latency problem in recommender systems. In *Proceedings of the Sixth European Conference on Case-based Reasoning* (Aberdeen, UK, September 2002), vol. 2416 of *LNCS*, Springer-Verlag, pp. 395–405.

[30] TOMBS, M. *Osmotic Pressure of Biological Macromolecules*. Oxford University Press, New York, NY, USA, 1997.