

Semantic Web Recommender Systems

Cai-Nicolas Ziegler

Institut für Informatik, Group DBIS,
Universität Freiburg, Germany
cziegler@informatik.uni-freiburg.de

Abstract. Research on recommender systems has primarily addressed centralized scenarios and largely ignored open, decentralized systems where remote information distribution prevails. The absence of superordinate authorities having full access and control introduces some serious issues requiring novel approaches and methods. Hence, our primary objective targets the successful deployment and integration of recommender system facilities for Semantic Web applications, making use of novel technologies and concepts and incorporating them into one coherent framework.

1 Introduction

Automated recommender systems intend to provide people with recommendations of products they might appreciate, taking into account their past product ratings profile and history of purchase or interest. Most successful systems apply so-called social filtering techniques [9], dubbed collaborative filtering [6]. These systems identify similar users and make recommendations based upon products people utterly fancy.

Unfortunately, common collaborative filtering methods fail when transplanted into decentralized scenarios. Analyzing the issues specific to these domains, we believe that two novel approaches may alleviate the prevailing problems, namely trust networks, along with trust propagation mechanisms, and taxonomy-driven profile generation and filtering. One aspect of our work hence addresses the conception of suitable components, specifically tailored to suit our decentralized setting, while another regards the seamless integration of these latter building bricks into one single, unified framework. Empirical analysis and performance evaluations are conducted at all stages.

2 Research Issues

Deploying recommender systems into the Semantic Web implies diverse, multi-faceted issues, some of them being inherent to decentralized systems in general, others being specific. Hereby, our devised Semantic Web recommender system performs all recommendation computations *locally* for one given user. Its principal difference from generic, centralized approaches refers to information storage, supposing all user and rating data *distributed* throughout the Semantic Web. Hence its decentralized nature. We thus come to identify several research issues:

- **Ontological Commitment.** Basically, the Semantic Web is made up of machine-readable content distributed all over the Web. In order to ensure that agents can understand and reason about the respective information, semantic interoperability via ontologies or common content models must be established. For instance, FOAF [5], an acronym for “Friend of a Friend”, defines an ontology for establishing simple social networks and represents an open standard agents can rely upon.
- **Interaction Facilities.** Decentralized recommender systems have primarily been subject to multi-agent research projects. In suchlike settings, environment models are agent-centric, enabling agents to directly communicate with their peers and thus making synchronous message exchange feasible. The Semantic Web, being an aggregation of distributed metadata, constitutes an inherently data-centric environment model. Messages are exchanged by publishing or updating documents encoded in RDF, OWL, or similar formats. Hence, the communication becomes restricted to asynchronous message exchange only.
- **Security and Credibility.** Closed communities generally possess efficient means to control the users’ identity and penalize malevolent behavior. Decentralized systems, among those peer-to-peer networks, open marketplaces and the Semantic Web, likewise, cannot prevent deception and insincerity. Spoofing and identity forging thus become facile to achieve [22]. Hence, some subjective means enabling each individual to decide which peers and content to rely upon are needed.
- **Computational Complexity and Scalability.** Centralized systems allow for estimating and limiting the community size and may thus tailor their filtering systems to ensure scalability. Note that user similarity assessment, which is an integral part of collaborative filtering [6], implies some computation-intensive processes. The Semantic Web will once contain millions of machine-readable homepages. Computing similarity measures for all these “individuals” thus becomes infeasible. Consequently, scalability can only be ensured when restricting these computations to sufficiently narrow neighborhoods. Intelligent filtering mechanisms are needed, still ensuring reasonable recall, i.e., not sacrificing too many relevant, like-minded agents.
- **Low Profile Overlap.** Interest profiles are generally represented by vectors indicating the user’s opinion for every product. In order to reduce dimensionality and ensure profile overlap, some centralized systems like Ringo [20] require users to rate *small subsets* of the overall product space. These mandatory assessments, provisional tools for creating overlap-ensuring profiles, imply additional efforts for prospective users. Other recommenders, among those GroupLens and MovieLens [14], operate in domains where product sets are comparatively small. On the Semantic Web, virtually no restrictions can be imposed on agents regarding which items to rate. Hence, new approaches to ensure profile overlap are needed in order to make profile similarity measures meaningful.

3 Proposed Approach

Endeavors to ensure semantical interoperability through ontologies constitute the cornerstone of Semantic Web conception and have been subject to numerous research projects.

We do not concentrate our efforts on this aspect but suppose data compatibility from the outset. Our interest rather focuses on handling computational complexity, security, data-centric message passing, and profile vector overlap. Hereby, our approach builds upon two fundamental notions, namely taxonomy-driven interest profile assembly and trust networks. The exploitation of synergies of both intrinsically separate concepts helps us leverage recommender system facilities into the Semantic Web.

3.1 Information Model

The infrastructure of the Semantic Web defines interlinked XML-documents made up of machine-readable metadata. Our information model presented below well complies with its design goals and allows facile mapping into RDF, OWL, etc.:

- **Set of agents** $A = \{a_1, a_2, \dots, a_n\}$. Set A contains all agents part of the community. Globally unique identifiers are assigned through URIs.
- **Set of products** $B = \{b_1, b_2, \dots, b_m\}$. All products considered are comprised in set B . Hereby, unique identifiers may refer to product descriptions from an online shop agreed upon, such as Amazon.com, or globally accepted codes, like ISBNs in case of books.
- **Set of partial trust functions** $T = \{t_1, t_2, \dots, t_n\}$. Every agent $a_i \in A$ has one partial trust function $t_i : A \rightarrow [-1, +1]^\perp$ that assigns continuous trust values to its peers. Functions $t_i \in A$ are partial since agents generally only rate small subsets of the overall community, hence rendering t_i sparse:

$$t_i(a_j) = \begin{cases} p, & \text{if } \text{trust}(a_i, a_j) = p \\ \perp, & \text{if no trust statement for } a_j \text{ from } a_i \end{cases} \quad (1)$$

We define high values for $t_i(a_j)$ to denote high trust from a_i in a_j , and negative values to express distrust, respectively. Values around zero indicate the absence of trust, not to be confused with explicit distrust [11].

- **Set of partial rating functions** $R = \{r_1, r_2, \dots, r_n\}$. In addition to functions $t_i \in T$, every $a_i \in A$ has one partial function $r_i : B \rightarrow [-1, +1]^\perp$ that expresses his liking or dislike of product $b_k \in B$. No person can rate every available product, so functions $r_i \in B$ are necessarily partial.

$$r_i(b_k) = \begin{cases} p, & \text{if } \text{rates}(a_i, b_k) = p \\ \perp, & \text{if no rating for } b_k \text{ from } a_i \end{cases} \quad (2)$$

Intuitively, high positive values for $r_i(b_k)$ denote that a_i highly appreciates b_k , while negative values express dislike, respectively.

- **Taxonomy C over set $D = \{d_1, d_2, \dots, d_l\}$** . 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.

We suppose all information about agents a_i , their trust relationships t_i and ratings r_i stored in machine-readable homepages distributed throughout the Web. Contrarily, taxonomy C , set B of products and descriptor assignment function f must hold *globally* and therefore offer public accessibility. Central maintenance of this information hence becomes inevitable. Later on, we will demonstrate that such sources of information for product categorization already exist for certain application domains.

3.2 Trust-Based Neighborhood Formation

The computation of trust-based neighborhoods constitutes one pivotal pillar of our approach. Clearly, neighborhoods are subjective, reflecting every agent a_i 's very beliefs about the accorded trustworthiness of immediate peers. The incorporation of trust-based social networks basically addresses two of the above-stated issues, namely security and credibility of information provided, and computational complexity:

Maintaining Security and Credibility. Trust renders automatic recommendation generation for a_i secure, making a_i rely upon opinions from peers deemed trustworthy only. Note that in general, collaborative filtering tends to be highly susceptible to manipulation. For instance, malicious agents a_j can accomplish high similarity with a_i simply by copying its rating profile [16].

Dealing with application scenarios subject to central control and monitoring, attacks on recommender systems only pose minor concerns. Contrarily, on the Semantic Web, where, according to Tim Berners-Lee, “anyone can say anything about anything”, attack-resistance and robustness may become an important criterion for recommender systems. Hence, as Marsh [11] already indicated, trust makes agents “less vulnerable to others”.

Computational Complexity Reduction. However, for our scenario, trust also serves another purpose, namely that of detecting similar peers. Neighborhood formation based upon common collaborative filtering techniques requires $O(|A|^2)$ time complexity and thus lacks scalability for decentralized scenarios. Contrarily, trust-driven neighborhood formation schemes actually do scale well. However, substituting common neighborhood formation approaches only makes sense when ensuring that trust also reflects similarity to a certain extent:

Recent studies by Swearingen and Sinha [21] provided empirical evidence that people tend to rely upon recommendations received from trusted fellows, i.e., friends, family members etc., more than upon online recommender systems. Moreover, own research endeavors [22] revealed that trust and interest similarity tend to positively correlate, justifying trust as an appropriate supplement or surrogate for collaborative filtering. Results from social psychology back our findings, since positive association between

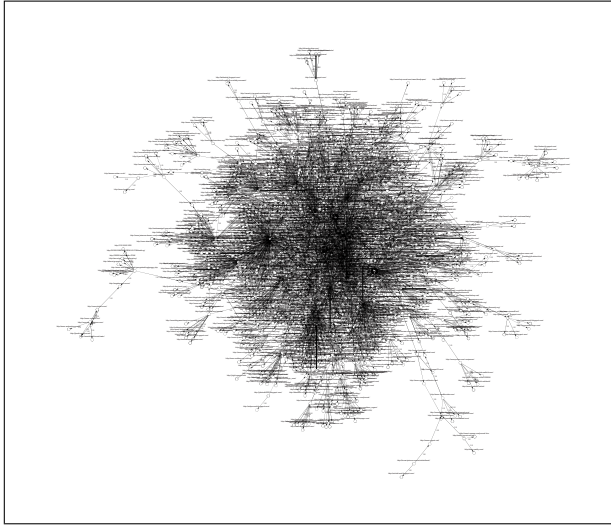


Fig. 1. All Consuming’s largest trust network cluster

attitudinal similarity and interpersonal attraction has effectively become one of this discipline’s most reliable findings [3].

Massa [12] conducted experiments on top of the well-known Epinions rating community (<http://www.epinions.com>), revealing that similarity computation based upon common collaborative filtering approaches fails when supposing very large product sets, hence making neighborhood formation impossible. Operating on the same data, Massa found that “trust-aware techniques can produce trust scores for very high numbers of peers”. Neighborhood formation thus becomes more facile to achieve.

Figure 1 shows the largest cluster of the All Consuming community trust network, visualized with layouting tools we implemented for that purpose. Note that approximately 95% of all members are part of the latter cluster, isolating comparatively few peers and levelling the ground for effective neighborhood formation.

Trust Propagation Models. Trust-based neighborhood detection for a_i , using those “trust-aware techniques” mentioned by Massa, implies *deriving* trust values for peers a_j not directly trusted by a_i , but one of the persons the latter agent trusts directly or indirectly. Note that functions $t_i(a_j)$ are commonly sparse, providing values for only few a_j compared to A ’s overall community size. Hence, trust metrics exploit the “conditional transitivity” property of trust [1] and allow for rendering trust functions dense.

Numerous scalar metrics [4, 10] have been proposed for computing trust between two given individuals a_i and a_j . We hereby denote computed trust weights by $t_i^c(a_j)$ as opposed to explicit trust $t_i(a_j)$. However, our approach requires metrics that compute *nearest trust neighbors*, and not evaluate trust values for any two given agents. We hence opt for local group trust metrics [23], which have only been attracting marginal interest until now. The most important and most well-known local group trust metric is Levien’s

Advogato metric [10]. However, the metric can only make *boolean* decisions with respect to trustworthiness, classifying agents into trusted and untrusted ones.

Appleseed [23], our own novel proposal for local group trust computation, allows more fine-grained analysis, assigning continuous trust weights for peers within trust computation range. Rankings thus become feasible. Appleseed's principal concepts derive from spreading activation models [18], which have been conceived for modelling human semantic memory. Appleseed operates on partial trust graph information, exploring the social network within predefined ranges only and allowing the neighborhood detection process to retain scalability. Hereby, high ranks are accorded to trustworthy peers, i.e., those agents which are largely trusted by others with high trustworthiness, similar to PageRank [17]. These ranks are used later on for selecting agents deemed suitable for making recommendations.

3.3 Taxonomy-Driven Similarity Metrics

Trust allows selecting peers with overall above-average interest similarity. However, for each active user, some peers having completely opposed interests generally exist. The proposition that interpersonal attraction, and hence trust, implies attitudinal similarity does not always hold true. Supplementary filtering thus becomes indispensable. Two approaches are conceivable:

- **User-User Closeness.** Additional filtering based upon profile similarity $c(a_i, a_j)$ between agents a_i, a_j is applied to the neighborhood of a_i 's trustworthy peers. The resulting set of peers only contains trusted peers which are *guaranteed* to resemble the active user a_i .
- **Product-User Relevance.** Instead of pruning the very neighborhood, one could also consider the set of all products b_k appreciated by at least one of a_i 's neighbors and dispose of those products not fitting the interest profile of a_i . We hereby denote the relevance of b_k for a_i by $c_b(a_i, b_k)$.

For both cases, the filtering task faces the problem of low profile overlap by virtue of information sparseness and potentially large product sets [12]. In order to alleviate the prevailing issue, we propose taxonomy-driven profile computation [22, 24], which allows to derive similarity between users a_i and a_j even though these peers have not rated one product in common. Moreover, our novel filtering method also permits to compute similarity between two products b_k, b_h .

Profile Generation. Our approach to taxonomy-driven generation of interest profiles [22, 24] extends basic ideas derived from Middleton's ontology-enhanced content-based filtering method [13]. In contrast to generic feature-based filtering, product categories still play an important role, but we have them arranged in a taxonomy and not separate from each other. Products b_k bear topic descriptors $d_{k_e} \in f(b_k)$ that relate these b_k to taxonomic nodes. Several classifications per product are possible, hence $|f(b_k)| \geq 1$. Each product the user likes infers some interest score for those $d_{k_e} \in f(b_k)$. Since these categories d_{k_e} are arranged in taxonomy C , we can also infer a fractional interest for all *super-topics* of d_{k_e} . Hereby, remote super-topics are accorded less interest score than super-topics close to d_{k_e} . Assume that (p_0, p_1, \dots, p_q) gives the path from top element

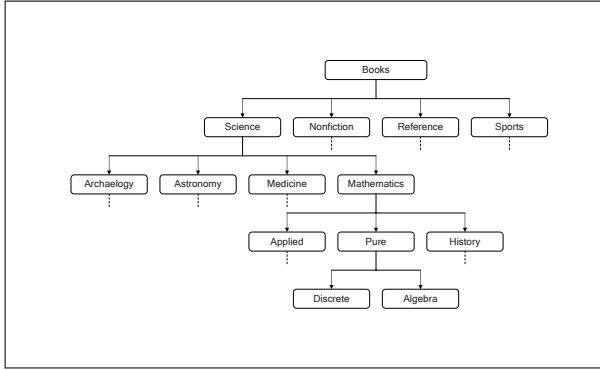


Fig. 2. Small fragment from the Amazon.com book taxonomy

$p_0 = \top$ to node $p_q = d_{\kappa_e}$. Function $\text{sib}(p)$ returns the number of p 's siblings, while $\text{sco}(p)$ returns p 's score:

$$\forall m \in \{0, 1, \dots, q - 1\} : \text{sco}(p_m) = \kappa \cdot \frac{\text{sco}(p_{m+1})}{\text{sib}(p_{m+1}) + 1} \tag{3}$$

Scores are normalized, i.e., all topic score that a_i 's profile assigns to nodes from taxonomy C amounts to some fixed value s . Hence, high product ratings from agents with short product rating histories have higher impact on profile generation than product ratings from persons issuing rife ratings. Score s is divided evenly among all products that contribute to a_i 's profile makeup. Factor κ permits fine-tuning the extent of super-topic score inference, depending on the underlying taxonomy's depth and granularity.

Example 1 (Topic Score Assignment). Suppose the taxonomy given in Figure 2 which represents a tiny fragment from the Amazon.com book taxonomy, and propagation factor $\kappa = 1$. Let user a_i have mentioned 4 books, namely Matrix Analysis, Fermat's Enigma, Snow Crash, and Neuromancer. For Matrix Analysis, 5 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 divided among these topics according to Equation 3. Score 29.087 becomes accorded to topic **Algebra**. Likewise, 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 the profile vector of user a_i .

Success or failure of our approach largely depends upon taxonomy C used for classification. The more thoroughly crafted and fine-grained the latter taxonomy, the more meaningful our profile information becomes. Clearly, topic descriptors $f(b_k)$ for products b_k must be chosen skillfully, too. By virtue of inference of fractional interest for super-topics, one may establish high user similarity for users which have not even rated one single product in common, as has been indicated before. According to our scheme, the more score two profiles have accumulated in same branches, the higher their computed similarity.

Similarity Computation. Taxonomy-driven interest profiles form the grounding for our novel filtering paradigm. Similarity computation between agents a_i, a_j , and between agents a_i and products b_k ¹, respectively, requires some distance metric.

For our approach, we apply common nearest-neighbor techniques, namely Pearson correlation [6, 20] and cosine distance known from information retrieval. Hereby, profile vectors map *category* score vectors from C instead of plain product-rating vectors. High similarity evolves from interest in many identical or related branches, whereas negative correlation indicates diverging interests.

For instance, suppose a_i reads literature about **Applied Mathematics** only, and a_j about **Algebra**, then their computed similarity will be high, considering significant branch overlap from node **Mathematics** onward.

3.4 Recommendation Generation

We already indicated that two alternative designs are viable for post-filtering trust neighborhoods. We opt for product-user relevance, i.e., deferring supplementary neighborhood filtering into the recommendation process. We hence consider *all* peers part of the trust neighborhood, but weed out products not matching the active user a_i 's profile.

The relevance of some product b_k one of a_i 's trusted peers recommends then depends on various factors, the two most important aspects being the following ones:

- **Accorded trust $t_i^c(a_j)$ of peers a_j mentioning b_k .** Trust-based neighborhood formation substitutes finding nearest neighbors based upon interest similarity. Likewise, similarity ranks $c(a_i, a_j)$ become replaced by trust weights $t_i^c(a_j)$ for computing the predicted relevance of b_k for a_i .
- **Content-based relevance $c_b(a_i, b_k)$ of product b_k for user a_i .** Besides mere trustworthiness of peers a_j rating product b_k , the content-based relevance of b_k for the active user a_i is likewise important, e.g., one may consider the situation where even close friends recommend products not fitting our interest profile at all.

Since both functions, i.e., $t_i^c(a_j)$ and $c_b(a_i, b_k)$, operate on completely different scales, the conversion of these absolute weights into ranks seems appropriate. For instance, the trusted agent a_j with highest trust weight $t_i^c(a_j)$ obtains rank 1, and so forth. Likewise, the product b_k with highest similarity $c_b(a_i, b_k)$ to the active user a_i 's interest profile becomes top-ranked.

Besides merging trustworthiness and content-based relevance ranks, other factors involved comprise product rating frequency and product description richness. Product rating frequency intends to reward products b_k recommended by numerous trusted peers, while description richness penalizes products bearing overly general taxonomic descriptions. The final ranks of products b_k eventually constitute the foundation for assembling top- N recommendation lists for the active user a_i .

¹ Supposing implicit product ratings only, the generation of taxonomy-driven profiles for products b_k equates profile generation for pseudo-user a_θ having implicitly rated b_k only.

3.5 Supplementary Fine-Tuning

Taxonomy-driven profile generation renders another mechanism, dubbed “topic diversification” [24], feasible. Hereby, our novel approach, optionally applicable on top of recommendation generation, allows rearrangement of the active user a_i ’s recommendation list in order to better reflect a_i ’s full range of interests, considering the impact of specific topics a_i implicitly declares interest in. For instance, suppose that novels classifying under Modern German Poetry have twice the share of Social Psychology in a_i ’s reading list. Then post-processing of a_i ’s recommendation list by means of topic diversification procedures allows to fully account for that fact.

To our best knowledge, no similar approaches exist or have been documented in literature affiliated with recommender systems. Moreover, topic diversification becomes even more valuable when making recommendations *across* diverse domains of interest, e.g., books, DVDs, apparel, etc.

Other enhancements include considering *explicit* product ratings for recommendation generation whenever available. However, note that most scenarios only allow for collecting implicit ratings, e.g., purchase data, product mentions, etc., rather than explicit ones.

4 Real-World Deployment

Section 3.1 exposed our envisioned information infrastructure. We will show that such an architecture may actually come into life and become an integral part of the Semantic Web:

- **Social Networks.** FOAF defines machine-readable homepages based upon RDF and allows weaving acquaintance networks. Golbeck [5] proposed some modifications making FOAF support “real” trust relationships instead of mere acquaintanceship.
- **Product Rating Information.** Moreover, FOAF networks seamlessly integrate with so-called “weblogs”, which are steadily gaining momentum. These personalized “online diaries” are especially valuable with respect to product rating information. For instance, some crawlers extract certain hyperlinks from weblogs and analyze their makeup and content. Hereby, those referring to product pages from large catalogs like Amazon.com (<http://www.amazon.com>) count as implicit votes for these goods. Mappings between hyperlinks and some sort of unique identifier are required for diverse catalogs, though. Unique identifiers exist for *some* product groups like books, which are given “International Standard Book Numbers”, i.e., ISBNs. Efforts to enhance weblogs with explicit, machine-readable rating information have also been proposed and are becoming increasingly popular. For instance, BLAM! (<http://www.pmbrowser.info/hublog/>) allows creating book ratings and helps embedding these into machine-readable weblogs.
- **Product Classification Taxonomies.** Besides user-centric information, i.e., agent a_i ’s trust relationships t_i and product ratings r_i , taxonomies for product classification play an important role within our approach. Luckily, these taxonomies exist for

certain domains. Amazon.com defines an extensive, fine-grained and deeply-nested taxonomy for books, containing thousands of topics. More important, Amazon.com provides books with subject descriptors referring to the latter taxonomy. Similar taxonomies exist for DVDs, CDs, and videos. Standardization efforts for product classification are channelled through the “United Nations Standard Products and Services Code” project (<http://www.unspsc.org/>). However, the UNSPSC’s taxonomy provides much less information and nesting than, for instance, Amazon.com’s taxonomy for books.

5 Experimental Setting

We created an experimental environment simulating the infrastructure proposed above. Hereby, by means of crawlers and screen scrapers, we gathered information from various trust-aware online communities like All Consuming (<http://www.allconsuming.net>), and Advogato (<http://www.advogato.org>), extracting information about approximately 10,000 users a_i , their trust relationships t_i and implicit product ratings r_i . Ratings were obtained from All Consuming only.

Moreover, we captured Amazon.com’s huge book taxonomy, made up of 13,525 hierarchically arranged topics d_e , and categorization data about 11,031 books b_k that All Consuming community members mentioned. Tailored crawlers search the Web for weblogs and ensure data freshness. All our experiments and empirical evaluations were based upon this “real-world” data.

6 Related Work

Recommender systems have begun attracting major research interest during the early nineties [6]. Nowadays, commercial and industrial systems are rife and wide-spread, detailed comparisons concerning features and approaches are given in [19]. Recommender systems differ from each other mainly through their filtering method. Hereby, distinctions between three types of filtering systems are made [6], namely collaborative, content-based and economic. Collaborative filtering systems [20] generate recommendations obtained from persons having similar interests. Content-based filtering only takes into account descriptions of products, based upon metadata and extracted features. Economic filtering has seen little practical application until now and exerts marginal impact only.

Modern recommender systems are hybrid, combining both content-based and collaborative filtering facilities in one single framework. Fab [2] counts among the first popular hybrid systems, more recent approaches have been depicted in [7] and [13]. Our filtering approach, comprising taxonomy-based profile generation and similarity computation, also exploits both content-based and collaborative filtering facilities. Trust networks add another supplementary level of filtering.

Initial attempts have been taken towards transplanting recommender systems into decentralized scenarios. Olsson [15] gives an extensive overview of existing approaches. Kautz et al. [9] extract implicit social network structures from the Web, using them

as foundations for recommender system services operating on top of these networks. Jensen et al. [8] propose an approach called “explicit peer-based systems”, which makes recommendations based upon friends’ opinions.

7 Future Directions

Our past efforts have mainly focused on designing suitable trust metrics for computing trust neighborhoods [23], and conceiving metrics for making collaborative filtering applicable to decentralized architectures [22]. Moreover, we have shaped and synthesized an extensive infrastructure based upon “real-world” data from various communities and online stores.

Until now, our analysis has been largely confined to the book domain only. Future research will also include movies and other specific product groups and investigate the intrinsic differences between these groups. For instance, Amazon.com’s taxonomy for DVD classification contains more topics than its book counterpart, though being less deep. We would like to better understand the impact that taxonomy structure may have upon profile generation and similarity computation.

Moreover, owing to the fact that our novel taxonomy-driven filtering approach yields excellent results compared to generic benchmark systems when dealing with information sparseness [24], we are planning to investigate our filtering paradigm’s performance when applied to dense product rating datasets, likewise.

References

1. Alvarez Abdul-Rahman and Stephen Hailes. A distributed trust model. In *New Security Paradigms Workshop*, pages 48–60, Cumbria, UK, September 1997.
2. Marko Balabanović and Yoav Shoham. Fab – Content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.
3. Ellen Berscheid. Interpersonal attraction. In Daniel Gilbert, Susan Fiske, and Gardner Lindzey, editors, *The Handbook of Social Psychology*, volume II. McGraw-Hill, New York, NY, USA, 4th edition, 1998.
4. Thomas Beth, Malte Borchertding, and Birgit Klein. Valuation of trust in open networks. In *Proceedings of the 1994 European Symposium on Research in Computer Security*, pages 3–18, 1994.
5. Jennifer Golbeck, Bijan Parsia, and James Hendler. Trust networks on the Semantic Web. In *Proceedings of Cooperative Intelligent Agents*, Helsinki, Finland, August 2003.
6. David Goldberg, David Nichols, Brian Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.
7. Zan Huang, Wingyan Chung, Thian-Huat Ong, and Hsinchun Chen. A graph-based recommender system for digital library. In *Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 65–73, Portland, OR, USA, 2002. ACM Press.
8. Carlos Jensen, John Davis, and Shelly Farnham. Finding others online: Reputation systems for social online spaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 447–454, Minneapolis, MN, USA, 2002. ACM Press.
9. Henry Kautz, Bart Selman, and Mehul Shah. Referral Web: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, March 1997.

10. Raph Levien and Alexander Aiken. Attack-resistant trust metrics for public key certification. In *Proceedings of the 7th USENIX Security Symposium*, San Antonio, TX, USA, January 1998.
11. Stephen Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, Department of Mathematics and Computer Science, University of Stirling, Stirling, UK, 1994.
12. Paolo Massa and Bobby Bhattacharjee. Using trust in recommender systems: an experimental analysis. In Christian Jensen, Stefan Poslad, and Theodosios Dimitrakos, editors, *Proceedings of the 2nd International Conference on Trust Management*, volume 2995 of *LNCS*, Oxford, UK, March 2004. Springer-Verlag.
13. Stuart Middleton, Harith Alani, Nigel Shadbolt, and David De Roure. Exploiting synergy between ontologies and recommender systems. In *Proceedings of the WWW2002 International Workshop on the Semantic Web*, volume 55 of *CEUR Workshop Proceedings*, Maui, HI, USA, May 2002.
14. Bradley Miller, Istvan Albert, Shyong Lam, Joseph Konstan, and John Riedl. MovieLens unplugged: Experiences with an occasionally connected recommender system. In *Proceedings of the ACM 2003 Conference on Intelligent User Interfaces (Accepted Poster)*, Chapel Hill, NC, USA, 2003. ACM.
15. Tomas Olsson. *Bootstrapping and Decentralizing Recommender Systems*. PhD thesis, Uppsala University, Uppsala, Sweden, 2003.
16. Michael O'Mahony, Neil Hurley, Nicolas Kushmerick, and Gu enol e Silvestre. Collaborative recommendation: A robustness analysis. *ACM Transactions on Internet Technology*, 4(3), August 2004.
17. Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.
18. Ross Quillian. Semantic memory. In Marvin Minsky, editor, *Semantic Information Processing*, pages 227–270. MIT Press, Boston, MA, USA, 1968.
19. Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, pages 158–166, Denver, CO, USA, 1999. ACM Press.
20. Upendra Shardanand and Patti Maes. Social information filtering: Algorithms for automating “word of mouth”. In *Proceedings of the ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1, pages 210–217, 1995.
21. Rashmi Sinha and Kirsten Swearingen. 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.
22. Cai-Nicolas Ziegler and Georg Lausen. Analyzing correlation between trust and user similarity in online communities. In Christian Jensen, Stefan Poslad, and Theodosios Dimitrakos, editors, *Proceedings of the 2nd International Conference on Trust Management*, volume 2995 of *LNCS*, pages 251–265, Oxford, UK, March 2004. Springer-Verlag.
23. Cai-Nicolas Ziegler and Georg Lausen. Spreading activation models for trust propagation. In *Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service*, Taipei, Taiwan, March 2004. IEEE Computer Society Press.
24. Cai-Nicolas Ziegler, Georg Lausen, and Lars Schmidt-Thieme. Taxonomy-driven computation of product recommendations, 2004. Submitted for publication.