Albert-Ludwigs-Universität Freiburg
Fakultät für Angewandte Wissenschaften, Institut für Informatik

---

Towards Decentralized Recommender Systems

Dissertation zur Erlangung des Doktorgrades (Dr.-Ing.)

von

Cai-Nicolas Ziegler

geboren am 28.11.1977 in Würzburg

Dekan : Prof. Dr. Jan Korvink

Referenten : **Prof. Dr. Georg Lausen**
Prof. Dr. Joseph A. Konstan

Datum der Promotion: **13. Juni 2005**

Zusammenfassung

Automatisierte Recommender-Systeme berechnen Produktvorschläge, welche genau auf die Interessen und Bedürfnisse ihrer Benutzer zugeschnitten sind und stellen somit exzellente Mittel dar, um der stetig wachsenden Informationsflut Herr zu werden. Allerdings sieht sich deren praktische Einsetzbarkeit bis dato weithin auf Szenarien beschränkt, bei denen man alle für die zur Berechnung von Empfehlungen relevante Information als in einem einzigen Knoten gekapselt annehmen konnte.

Seit einigen Jahren nehmen verteilte Infrastrukturen, wie zum Beispiel Peer-to-Peer und Ad-Hoc Netzwerke, das Semantic Web, der Grid etc., immer deutlichere Konturen an und ersetzen klassische Client/Server-Modelle bereits in vielerlei Hinsicht. Diese Infrastrukturen könnten gleichwohl von den von Recommender-Systemen bereitgestellten Diensten profitieren und somit einen Paradigmenwechsel hin zu *dezentralisierten* Recommender-Systemen einläuten.

Im Rahmen dieser Dissertation untersuchen wir zunächst die neuen Herausforderungen, denen es sich im Hinblick auf die Konzeption dezentraler Recommender-Systeme zu stellen gilt, und schlagen diverse neue Ansätze vor, mit deren Hilfe man speziell jene Probleme zu bewältigen vermag. Das Spektrum der vorgestellten Methoden reicht dabei von der Verwendung von mächtigen Taxonomien zur Klassifikation von Produkten zwecks künstlicher Verdichtung der Daten, bis hin zu Vertrauensmetriken, die entworfen wurden, um Fragen der Skalierbarkeit derartiger Systeme zu lösen. Empirische Untersuchungen bezüglich der Korrelation interpersonellen Vertrauens und Interessengleichheit stellen den Mörtel dar, welcher jene einzelnen Bausteine zusammenfügt und die schlussendliche Realisierung eines exemplarischen Frameworks für dezentrale Recommender-Systeme ermöglicht.

Während die angesprochenen Bausteine, im namentlichen Taxonomie-basiertes Filtern, Topic Diversification und die Appleseed Vertrauensmetrik, notwendige Komponenten für die Konzeption eines auf sozialem Vertrauen basierten, dezentralen Recommender-Systems darstellen, so sind diese gleichermaßen wichtige wissenschaftliche Beiträge per se und auch außerhalb der Fragestellung “Dezentrale Recommender-Systeme” von praktischer Relevanz.

Abstract

Automated recommender systems make product suggestions that are tailored to the human user's individual needs and represent powerful means to combat information glut. However, their practical applicability has been largely confined to scenarios where all information relevant for recommendation making is kept in one single, authoritative node.

Recently, novel distributed infrastructures are emerging, e.g., peer-to-peer and ad-hoc networks, the Semantic Web, the Grid, etc., and supersede classical client/server approaches in many respects. These infrastructures could likewise benefit from recommender system services, leading to a paradigm shift towards *decentralized* recommender systems.

In this thesis, we investigate the challenges that decentralized recommender systems bring up and propose diverse techniques in order to cope with those particular issues. The spectrum of methods proposed ranges from the employment of product classification taxonomies as powerful background knowledge, alleviating the sparsity problem, to trust propagation mechanisms designed to address the scalability issue. Empirical investigations on the correlation of interpersonal trust and interest similarity provide the component glue that melds these results together and renders the eventual creation of a decentralized recommender framework feasible.

While these building bricks, namely taxonomy-driven filtering, topic diversification, and the Appleseed trust metric, are vital for the conception of our trust-based decentralized recommender, they are also valuable contributions in their own right, addressing issues not only confined to the universe of decentralized recommender systems.

Acknowledgements

Above all, I would like to thank Prof. Dr. Georg Lausen, my supervisor at the University of Freiburg. He has supported me in all my endeavors and provided encouragement and guidance throughout this work's lifetime. Besides, being the head of the group of databases and information systems (DBIS) and being the person he is, he has created an environment where it is a pleasure to work and conduct research.

I would also like to thank my second supervisor, Prof. Dr. Joseph A. Konstan, who has provided invaluable input from different, more HCI-centric perspectives during my research stay at the GroupLens labs, University of Minnesota. Thanks as well to Prof. Dr. Dr. Lars Schmidt-Thieme for fruitful discussions on methods to quantify recommender system performance and how to improve the quality of my work, and to Prof. Dr. Günter Müller for his kind willingness to read and evaluate my dissertation.

During my time as a research assistant in the databases group I have learned to appreciate the very warm and pleasant working atmosphere, created by my colleagues Matthias Ihle, Martin Weber, Elisabeth Lott, Norbert Küchlin, and Harald Hiss. Not to forget my former cohorts Lule Ahmedi and Fang Wei. And, of course, my roommate and almost-namesake Kai Simon with whom it has always been a pleasure to talk, discuss, and share a good deal of laughs.

Thanks as well to all those researchers who helped me on my way, particularly Paolo Massa, Zvi Topol, Ernesto Díaz-Avilés, Jen Golbeck, Sean M. McNee, John Riedl, and Dan Cosley. Moreover, I would like to express my gratitude towards Ron Hornbaker and Erik Benson, maintaining the All Consuming and BookCrossing community, respectively, for rendering the online user studies possible. Without their effort and willingness to help, a great deal of my work would not have been viable.

Last, but by no means least, I would like to thank my best friends, Peter, Bille, Tim, Betty, Mel, Jari, Andreas, Christian, Robin, Daniela, Matthias, Virginie, A, Silvi, Stefan, Tanja, and Katja, many of them having accompanied me for more than ten years. I am greatly indebted for their affection, their support and encouragement. Also I cannot thank my beloved parents and Chris, my cute "little" brother, enough for their love and their devotion, helping me in any conceivable respect. Finally, I want to thank Vanessa, my fair lady, for remembering me that there are other things out there than computational models and machine-processable information.

Freiburg im Breisgau, February 17, 2005 Cai-Nicolas Ziegler

To my parents, and Chris, my little brother.

Contents

Chapter 1

Introduction

"Networks straddle the world [...]. But the sheer volume of information dissolves the information. We are unable to take it all in."

– Günther Grass (*1927)

Contents

1.1 Motivation

Total information overload becomes increasingly severe in our modern times of omnipresent mass-media and global communication facilities, exceeding the human perception's ability to dissect relevant information from irrelevant. Consequently, since more than 60 years [van Rijsbergen, 1975], significant research efforts have been striving to conceive automated filtering systems that provide humans with desirable and relevant information only. Search engines count among these filtering systems and have gained wide-spread acceptance, rendering information search feasible even within chaotic and anarchical environments such as the Web.

During the last 10 years, recommender systems [Resnick and Varian, 1997; Konstan, 2004] have been gaining momentum as another efficient means of reducing complexity when searching for relevant information. Recommenders intend to provide people with suggestions of products they will appreciate, based upon their past preferences, history of purchase, or demographic information [Resnick et al., 1994].

1.1.1 Collaborative Filtering Systems

Most successful industrial and academic recommender systems employ so-called collaborative filtering techniques [Goldberg et al., 1992]. Collaborative filtering systems mimic social processes, when asking like-minded friends or family members for their particular opinion on new book releases, in an automated fashion. Their principal mode of operation can be broken down into three major steps:

- **Profiling.** For each user a_i part of the community, an interest profile for the domain at hand, e.g., books, videos, etc., is computed. In general, these profiles are represented as partial *rating functions* $r_i : B \to [-1, +1]^{\perp}$, where $r_i(k) \in [-1, +1]$ gives a_i's rating for product b_k, taken from the current domain of interest. Ratings are expressions of value for products. High values $r_i(k) \to +1$ denote appreciation, while low values $r_i(k) \to -1$ indicate dislike.
- **Neighborhood formation.** Neighborhood formation aims at finding the best-M like-minded neighbors of a_i, based on their profiles of interest. Roughly speaking, the more ratings two users a_i, a_j have in common, and the more their corresponding ratings $r_i(k), r_j(k)$ have similar or identical values, the higher the similarity between a_i and a_j.
- **Rating prediction.** Predictions for products b_k still unknown to a_i depend on mainly two factors. First, the *similarity* of neighbors a_j having rated b_k, and second, the *rating* $r_j(k)$ they have assigned to product b_k. Eventually, top-N recommendation lists for users a_i are compiled based upon these predictions.

Hence, the intuition behind collaborative filtering is that if user a_i has agreed with his neighbors in the past, he will do so in the future as well.

1.1.2 Towards Decentralization

With few exceptions [Foner, 1999; Olsson, 2003; Miller, 2003; Sarwar, 2001], recommender systems have been crafted with *centralized* scenarios in mind, i.e., central computational control and central data storage. On the other hand, *decentralized* infrastructures are becoming increasingly popular on the Internet and the Web, among those the Semantic Web, the Grid, peer-to-peer networks for file-sharing and collaborative tasks, and ubiquitous computing. All these scenarios comprise an abundant wealth of metadata information that could be exploited for personalized recommendation making. For instance, think of the Semantic Web as an enormous network of inter-linked personal homepages published in machine-readable fashion. All these homepages contain certain preference data, such as the respective user's friends and trusted peers, and appreciated products, e.g., CDs, DVDs, and so forth. Through weblogs, best described as online diaries, parts of this vision have already become true and gained wide-spread acceptance.

We could exploit this existing information infrastructure, the user's personal preferences with respect to peers and products, in order to provide personalized recommendations of products he might have an interest in. This personal recommender would thus be an application running on one local node, namely the respective user's personal computer, and collect data from throughout the Semantic Web. Our devised recommender would thus exhibit the following characteristics:

Centralized computation. All computations are performed on one single node. Since we assume these nodes to be people's PCs, limitations are set concerning the computational power, i.e., we cannot suppose large clusters of high-speed servers as is the case for e-commerce stores and larger online communities.

Decentralized data storage. Though computations are localized, data and preference information are not. They are distributed throughout the network, e.g., the Semantic Web, and peers generally maintain partial views of their environment only.

Though having referred to the Semantic Web in the above example, the devised example also translates to other decentralized infrastructures, such as the beforementioned peer-to-peer file-sharing systems.

1.2 Research Issues

Now, when thinking of personal recommender systems exhibiting features as those depicted above, several research issues spring to mind that are either non-existent or less severe when dealing with conventional, centralized approaches:

- **Ontological commitment.** The Semantic Web and other decentralized infrastructures feature 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 [Golbeck et al., 2003], an acronym for "Friend of a Friend", defines an ontology for establishing simple social networks and represents an open standard systems can rely upon.
- **Interaction facilities.** Decentralized recommender systems have primarily been subject to multi-agent research projects [Foner, 1997; Olsson, 1998; Chen and Singh, 2001]. In these 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, opts for an inherently *data*-centric environment model. Messages are exchanged by publishing or updating documents encoded in RDF, OWL, or similar formats. Hence, 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 devoid of central authorities, e.g., 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 [Lam and Riedl, 2004; O'Mahony et al., 2004]. 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 [Goldberg et al., 1992], 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.
- **Sparsity and low profile overlap.** As indicated in Section 1.1.1, interest profiles are generally represented by vectors showing the user's opinion for every product. In order to reduce dimensionality and ensure profile overlap, hence combatting the so-called sparsity issue [Sarwar, 2001], some centralized systems like MovieLens (*http://www.movielens.org*) and Ringo [Shardanand and Maes, 1995] prompt people 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, such as FilmTrust (*http://trust.mindswap.org/FilmTrust/*) and Jester [Goldberg et al., 2001], 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. Instead, "anyone can say anything about anything", as stated by Tim Berners-Lee. Hence, new approaches to ensure profile overlap are needed in order to make profile similarity measures meaningful.

1.3 Proposed Approach

In this thesis, we attack some of the above-mentioned issues and integrate our techniques and results into one coherent framework:

Endeavors to ensure semantical interoperability through ontologies constitute the cornerstone of Semantic Web conception and have been subject to numerous research projects. Consequently, we do not concentrate on this aspect but suppose data compatibility from the outset, relying upon common quasi-standards, e.g., FOAF and

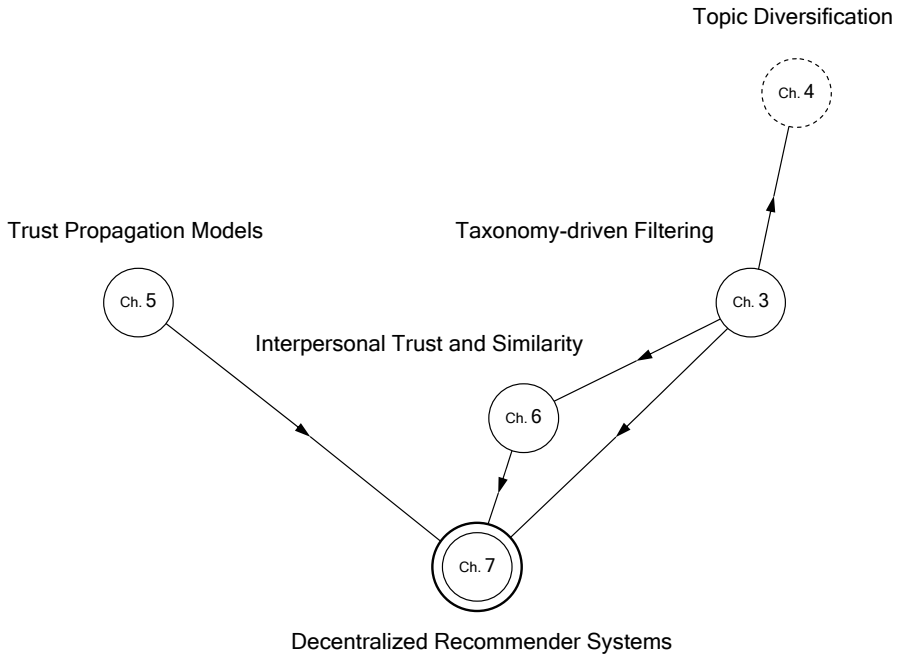

Figure 1.1. Dependency graph modelling the contributions of this thesis

friends. Our interest rather focuses on handling computational complexity, security, data-centric message passing, and sparsity. To this end, we introduce two fundamental approaches, namely taxonomy-driven profiling and filtering and spreading activation-based trust propagation. While taxonomy-driven profiling addresses the sparsity issue mentioned in Section 1.2, trust networks and trust propagation attack the security and computational complexity issues.

Interdependencies between contributions made are depicted in Figure 1.1. Edges point from contributions *exerting* an impact to those they *influence*. Hence, our method for taxonomy-driven filtering, presented in Chapter 4, affects topic diversification and our evaluation framework for analyzing interactions between trust and similarity, likewise. Taxonomy-driven filtering makes use of product classification taxonomies as powerful background knowledge to make user profiles denser and create overlap between two user profiles even when those two users have not rated any product in common. Within the taxonomy-driven filtering framework, we present topic diversification, used therein in order to avoid overfitting when learning preferences.

However, we found that topic diversification also works as an efficient means to increase user satisfaction when applied on top of conventional collaborative filtering algorithms. Its in-depth investigation, outlined in Chapter 4, does not contribute to

the overall problem setting of crafting decentralized recommender systems, though, which is indicated through dashed lines in Figure 1.1.

Chapter 6 presents a framework we conceived in order to analyze correlations between interpersonal trust and attitudinal similarity. To this end, owing to the extreme sparsity of the dataset we conducted our empirical study upon, we applied our taxonomy-driven profiling approach in order to make similarity computations more meaningful.

In Chapter 5 we introduce Appleseed, a local group trust metric based on spreading activation models, designed for computing subjective neighborhoods of most trustworthy peers on the network. Taxonomy-driven filtering, Appleseed, and knowledge about positive impacts of trust relationships on interest similarity constitute necessary prerequisites for the decentralized recommender framework we propose in Chapter 7.

1.3.1 Contributions

The contributions made in this thesis refer to various disciplines, e.g., information filtering and retrieval, network analysis, and social psychology. Note that all these building bricks are *contributions in their own right*, being important not only for the overall decentralized recommender framework, but also for numerous other applications, e.g., reputation systems [Kinateder and Rothermel, 2003; Chen and Singh, 2001], open rating systems [Guha, 2003], and dimensionality reduction in filtering systems [Sarwar et al., 2000b].

- **Taxonomy-driven filtering.** Taxonomy-driven filtering relies upon very large product classification taxonomies as powerful background knowledge to render recommendation computations feasible in environments where sparsity prevails. Our proposed approach features two important contributions:

  Interest profile assembly and similarity measurement. Our method for assembling profiles based on *topic interests* rather than *product ratings* constitutes the nucleus of taxonomy-driven filtering. Hereby, hierarchical relationships between topics provide an essential inference mechanism. The resulting profiles serve as means for computing user-user, user-product, and product-product similarity.

  Recommender framework for sparse data. We embedded the taxonomy-driven profiling and similarity measuring technique into an adaptive framework for computing recommendations. The respective system, devised for centralized data storage and computation, also introduces a new product-user relevance predictor. Empirical evidence, featuring both online and offline evaluations for two different datasets, shows our approach's superior performance over common benchmark systems for sparse product-rating matrices.

- **Topic diversification.** Originally conceived as an auxiliary procedure for the taxonomy-driven recommender, topic diversification appeared as an excellent means to increase user satisfaction when applied to recommendation lists computed by arbitrary recommenders. Ample offline analysis and an online study involving more than 2,100 users underline its suitability as a supplement procedure for item-based collaborative filtering algorithms [Sarwar et al., 2001; Karypis, 2001; Deshpande and Karypis, 2004], mitigating the so-called "portfolio effect" [Ali and van Stam, 2004] these systems suffer from.
- **Trust propagation based on spreading activation models.** Trust metrics have been introduced for modelling the so-called Public Key Infrastructure (PKI) [Zimmermann, 1995] and are nowadays gaining momentum through the emergence of decentralized infrastructures such as the Semantic Web [Richardson et al., 2003; Golbeck et al., 2003] and P2P [Kamvar et al., 2003; Aberer and Despotovic, 2001]. We propose a scalable, attack-resistant [Levien, 2004; Twigg and Dimmock, 2003] trust metric based upon spreading activation models which is able to compute *neighborhoods* of most-trustworthy peers.
- **Interpersonal trust and interest similarity.** While the proverbial saying that "birds of a feather flock together" suggests that bonds of trust are mostly forged among like-minded individuals, no empirical evidence has been given so far. Social psychology has conducted considerable research on interactions between interpersonal *attraction* and attitudinal similarity [Berscheid, 1998], but not with respect to interpersonal *trust*. We therefore present a framework for analyzing interdependencies between trust and similarity and provide empirical, statistically significant evidence from an online book-reading community showing that trust and similarity do correlate within this context.
- **Decentralized recommender framework.** Our final contribution aims at the seamless integration of all previous results and techniques, excepting topic diversification, into one coherent framework for decentralized recommendation making. Empirical results based on offline analysis are given in order to compare its efficiency with two non-decentralized recommenders not making use of trust-based neighborhood formation.

1.3.2 Published Work

Large portions of contributions made in this work have been published in international conferences [Ziegler and Lausen, 2004c,a,b; Ziegler et al., 2004, 2005], refereed workshops [Ziegler, 2004a; Ziegler, Schmidt-Thieme, and Lausen, 2004; Ziegler, 2004b], and journals [Ziegler and Lausen, 2005]. Results of Chapter 3 have been published in [Ziegler, Lausen, and Schmidt-Thieme, 2004; Ziegler, Schmidt-Thieme, and Lausen, 2004], and Chapter 4 largely relates to [Ziegler et al., 2005]. The contents of Chapter 5 have appeared in [Ziegler and Lausen, 2004c] and have been extended in [Ziegler and Lausen, 2005]. Portions of Chapter 6 are documented in [Ziegler and

Lausen, 2004a], while [Ziegler, 2004a; Ziegler and Lausen, 2004b; Ziegler, 2004b] cover Chapter 7 and, in part, Chapter 1.

Chapter 2

On Recommender Systems

"Attitude is a little thing that makes a big difference."

– Winston Churchill (1874-1965)

Contents

2.1 Introduction

Recommender systems [Resnick and Varian, 1997] have gained wide-spread acceptance and attracted increased public interest during the last decade, levelling the ground for new sales opportunities in e-commerce [Schafer et al., 1999; Sarwar et al., 2000a]. For instance, online retailers such as Amazon.com (*http://www.amazon.com*) successfully employ an extensive range of different types of recommender systems.

Their principal objective is that of complexity reduction for the human being, sifting through very large sets of information and selecting those pieces that are relevant for the active user[1]. Moreover, recommender systems apply personalization techniques, considering that different users have different preferences and different information needs [Konstan et al., 1997]. For instance, supposing the domain of book recommendations, historians are supposedly more interested in medieval prose, e.g.,

[1]The term "active user" refers to the person for whom recommendations are made.

Geoffrey Chaucer's Canterbury Tales, than literature about self-organization, which might be more relevant for AI researchers.

2.2 Collecting Preference Information

Hence, in order to generate personalized recommendations that are tailored to the active user's specific needs, recommender systems must collect personal preference information, e.g., the user's history of purchase, click-stream data, demographic information, and so forth. Traditionally, expressions of preference of users a_i for products b_k are generally called *ratings* $r_i(b_k)$. Two different types of ratings are distinguished:

Explicit ratings. Users are required to *explicitly* specify their preference for any particular item, usually by indicating their extent of appreciation on 5-point or 7-point likert scales. These scales are then mapped to numeric values, for instance continuous ranges $r_i(b_k) \in [-1, +1]$. Negative values commonly indicate dislike, while positive values express the user's liking.

Implicit ratings. Explicit ratings impose additional efforts on users. Consequently, users often tend to avoid the burden of explicitly stating their preferences and either leave the system or rely upon "free-riding" [Avery and Zeckhauser, 1997]. Alternatively, garnering preference information from mere *observations* of user behavior is much less obtrusive [Nichols, 1998]. Typical examples for implicit ratings are purchase data, reading time of Usenet news [Resnick et al., 1994], and browsing behavior [Gaul and Schmidt-Thieme, 2002; Middleton et al., 2004]. While easier to collect, implicit ratings bear some serious implications. For instance, some purchases are gifts and thus do not reflect the active user's interests. Moreover, the inference that purchasing implies liking does not always hold.

Owing to the difficulty of acquiring explicit ratings, some providers of product recommendation services adopt bilateral approaches. For instance, Amazon.com computes recommendations based on explicit ratings *whenever possible*. In case of unavailability, observed implicit ratings are used instead.

2.3 Recommender System Types and Techniques

Two principal paradigms for computing recommendations have emerged, namely *content-based* and *collaborative* filtering [Goldberg et al., 1992]. Content-based filtering, also called *cognitive filtering* [Malone et al., 1987], computes similarities between the active user a_i's basket of appreciated products, and products from the product universe that are still unknown to a_i. Product-product similarities are based

on features and selected attributes. Whereas collaborative filtering, also called *social filtering* [Resnick et al., 1994], computes similarities between *users*, based upon their rating profile. Most similar users then serve as "advisers" suggesting the most relevant products to the active user.

Advanced recommender systems tend to *combine* collaborative and content-based filtering, trying to mitigate the drawbacks of either approach and exploiting synergetic effects. These systems have been coined "hybrid systems" [Balabanović and Shoham, 1997]. Burke [2002] provides an extensive survey of hybridization methods.

2.3.1 Content-based Techniques

Content-based approaches to recommendation making are deeply rooted in information retrieval [Baudisch, 2001]. Typically, these systems learn Bayesian classifiers through content features [Lang, 1995; Ghani and Fano, 2002; Lam et al., 1996; Sollenborn and Funk, 2002], or perform nearest-neighbor vector-space queries [Pazzani, 1999; Alspector et al., 1998; Mukherjee et al., 2001; Ferman et al., 2002]. Bayesian classifiers use Bayes' theorem of conditional independence:

$$P(R \mid F) = \frac{P(F \mid R) \cdot P(R)}{P(F)} \tag{2.1}$$

Moreover, Bayesian classifiers make the "naïve" assumption that product description features are independent, which is usually not the case. Given the class label, the probability of b_k belonging to class R_i, given its n feature values $F_1, \ldots, F_n$, is defined as follows:

$$P(R_i \mid F_1, \ldots, F_n) = \frac{1}{Z} \cdot P(R_i) \cdot \prod_{j=1}^{n} P(F_j \mid R_i) \tag{2.2}$$

Variable Z represents a scaling factor only dependent on $F_1, \ldots, F_n$. Probabilities $P(R_i)$ and $P(F_j \mid R_i)$ can be estimated from training data.

For vector-space queries, attributes, e.g., plain-text terms or machine-readable metadata, are extracted from product descriptions and used for user profiling and product representation. For instance, Fab [Balabanović and Shoham, 1997] represents documents in terms of the 100 words with the highest TF-IDF weights [Baeza-Yates and Ribeiro-Neto, 1999], i.e., the words that occur more frequently in those documents than they do on average.

2.3.2 Collaborative Filtering

Content-based filtering only works when dealing with domains where feature extraction is feasible and attribute information readily available. Collaborative filtering (CF), on the other hand, uses content-less representations and does not face

that same limitation. For instance, Jester [Goldberg et al., 2001] uses collaborative filtering to recommend jokes to its users. While content-based filtering considers the descriptive *features* of products, collaborative filtering uses the *ratings* that users assign to products. Hence, CF algorithms typically operate on a set of users $A = \{a_1, a_2, \ldots, a_n\}$, a set of products $B = \{b_1, b_2, \ldots, b_m\}$, and partial rating functions $r_i : B \to [-1, +1]^{\perp}$ for each user $a_i \in A$. Negative values $r_i(b_k)$ denote dislike, while positive values express a_i's liking of product b_k. Bottom values $r_i(b_k) = \perp$ indicate that a_i has not rated b_k.

Owing to their high quality output and minimal information requirements, CF systems have become the most prominent representatives of recommender systems. Many commercial vendors, e.g., Amazon.com [Linden et al., 2003] and TiVo [Ali and van Stam, 2004], use variations of CF techniques to suggest products to their customers. Besides simple Bayesian classifiers [Miyahara and Pazzani, 2000; Breese et al., 1998; Lang, 1995; Lam et al., 1996], horting [Aggarwal et al., 1999], and association rule-based techniques [Sarwar et al., 2000a], mainly two approaches have acquired wide-spread acceptance, namely *user-based* and *item-based* collaborative filtering. In fact, the term "collaborative filtering" is commonly used as a synonym for user-based CF, owing to this technique's immense popularity.

The following two sections roughly depict algorithmic implementations of both user-based and item-based CF.

2.3.2.1 User-based Collaborative Filtering

The Ringo [Shardanand and Maes, 1995] and GroupLens [Konstan et al., 1997] projects have been among the first recommender systems to apply techniques known as "user-based collaborative filtering". Representing each user a_i's rating function r_i as a vector, they first compute similarities $c(a_i, a_j)$ between all pairs $(a_i, a_j) \in A \times A$. To this end, common statistical correlation coefficients are used, typically Pearson correlation [Resnick et al., 1994], and the cosine similarity measure, well-known from information retrieval [Baeza-Yates and Ribeiro-Neto, 1999]. As its name already suggests, the cosine similarity measure quantifies the similarity between two vectors $\vec{v_i}, \vec{v_j} \in [-1, +1]^{|B|}$ by the cosine of their angles:

$$\mathrm{sim}(\vec{v_i}, \vec{v_j}) = \frac{\sum_{k=0}^{|B|} v_{i,k} \cdot v_{j,k}}{\left(\sum_{k=0}^{|B|} v_{i,k}^2 \cdot \sum_{k=0}^{|B|} v_{j,k}^2 \right)^{\frac{1}{2}}} \tag{2.3}$$

Pearson correlation, derived from a linear regression model [Herlocker et al., 1999], is similar to cosine similarity, but measures the degree to which a linear relationship exists between two variables. Symbols $\overline{v_i}, \overline{v_j}$ denote the *averages* of vectors $\vec{v_i}, \vec{v_j}$:

$$\text{sim}(\vec{v_i}, \vec{v_j}) = \frac{\sum_{k=0}^{|B|} (v_{i,k} - \overline{v_i}) \cdot (v_{j,k} - \overline{v_j})}{\left(\sum_{k=0}^{|B|} (v_{i,k} - \overline{v_i})^2 \cdot \sum_{k=0}^{|B|} (v_{j,k} - \overline{v_j})^2 \right)^{\frac{1}{2}}} \tag{2.4}$$

Either using the cosine similarity measure or Pearson correlation to compute similarities $c(a_i, a_j)$ between all pairs $(a_i, a_j) \in A \times A$, *neighborhoods* $\text{prox}(a_i)$ of top-M most similar neighbors are built for every peer $a_i \in A$. Next, predictions are computed for all products b_k that a_i's neighbors have rated, but which are yet unknown to a_i, i.e, more formally, predictions $w_i(b_k)$ for $b_k \in \{b \in B \mid \exists a_j \in \text{prox}(a_i) : r_j(b) \neq \perp\}$:

$$w_i(b_k) = \overline{r_i} + \frac{\sum_{a_j \in \text{prox}(a_i)} (r_j(b_k) - \overline{r_j}) \cdot c(a_i, a_j)}{\sum_{a_j \in \text{prox}(a_i)} c(a_i, a_j)} \tag{2.5}$$

Predictions are thus based upon weighted averages of deviations from a_i's neighbors' means. For top-N recommendations, a list $P_{w_i} : \{1, 2, \ldots, N\} \rightarrow B$ is computed, based upon predictions w_i. Note that function P_{w_i} is injective and reflects recommendation ranking in *descending* order, giving highest predictions first.

Performance Tuning

In order to make better predictions, various researchers have proposed several modifications to the core user-based CF algorithm. The following list names the most prominent ones, but is certainly not exhaustive:

- **Inverse user frequency.** In information retrieval applications based on the vector-space model, word frequencies are commonly modified by a factor known as the "inverse document frequency" [Baeza-Yates and Ribeiro-Neto, 1999]. The idea is to reduce the impact of frequently occurring words, and increase the weight for uncommon terms when computing similarities between document vectors. Inverse user frequency, first mentioned by Breese et al. [1998], adopts that notion and rewards co-votes for less common items much more than co-votes for very popular products.
- **Significance weighting.** The computation of user-user correlations $c(a_i, a_j)$ only considers products that *both* users have rated, i.e., $b_k \in (\{b \,|\, r_i(b) \neq \perp\} \cap \{b \,|\, r_j(b) \neq \perp\})$. Hence, even if a_i and a_j have co-rated only one single product b_k, they will have maximum correlation if $r_i(b_k) = r_j(b_k)$ holds. Clearly, such correlations, being based upon few data-points only, are not very reliable. Herlocker et al. [1999] therefore proposed to *penalize* user correlations based on fewer than 50 ratings in common, applying a significance weight of $s/50$,

where s denotes the number of co-rated items. Default voting [Breese et al., 1998] is another approach to address the same issue.

- **Case amplification.** While both preceding modifications refer to the similarity computation process, case amplification [Breese et al., 1998] addresses the *rating prediction* step, formalized in Equation 2.5. Correlation weights $c(a_i, a_j)$ close to one are emphasized, and low correlation weights punished:

$$c'(a_i, a_j) = \begin{cases} c(a_i, a_j)^\rho, & \text{if } c(a_i, a_j) \geq 0 \\ -(-c(a_i, a_j))^\rho, & \text{else} \end{cases} \tag{2.6}$$

  Hence, highly similar users have much more impact on predicted ratings than before. Values ρ around 2.5 are typically assumed.

Filtering Agents

Some researchers [Sarwar et al., 1998; Good et al., 1999] have taken the concept of user-based collaborative filtering somewhat further and added *filterbots* as additional users eligible for selection as neighbors for "real" users. Filterbots are automated programs behaving in certain, pre-defined ways. For instance, in the context of the GroupLens Usenet news recommender [Konstan et al., 1997], some filterbots rated Usenet articles based on the proportion of spelling errors, while others focused on text length, and so forth. Sarwar has shown that filterbots can improve recommendation accuracy when operating in sparsely populated CF systems [Sarwar et al., 1998].

2.3.2.2 Item-based Collaborative Filtering

Item-based CF [Karypis, 2001; Sarwar et al., 2001; Deshpande and Karypis, 2004] has been gaining momentum over the last five years by virtue of favorable computational complexity characteristics and the ability to decouple the model computation process from actual prediction making. Specifically for cases where $|A| \gg |B|$, item-based CF's computational performance has been shown superior to user-based CF [Sarwar et al., 2001]. Its success also extends to many commercial recommender systems, such as Amazon.com's [Linden et al., 2003] and TiVO [Ali and van Stam, 2004].

As with user-based CF, recommendation making is based upon ratings $r_i(b_k)$ that users $a_i \in A$ provide for products $b_k \in B$. However, unlike user-based CF, similarity values c are computed for *items* rather than *users*, hence $c : B \times B \rightarrow [-1, +1]$. Roughly speaking, two items b_k, b_e are similar, i.e., have large $c(b_k, b_e)$, if users who rate one of them tend to rate the other, and if users tend to assign identical or similar ratings to them. Effectively, item-based similarity computation equates to the user-based case when turning the product-user matrix 90°. Next, neighborhoods $\text{prox}(b_k) \subseteq B$ of top-M most similar items are defined for each b_k. Predictions $w_i(b_k)$

are computed as follows:

$$w_i(b_k) = \frac{\sum_{b_e \in B'_k} (c(b_k, b_e) \cdot r_i(b_e))}{\sum_{b_e \in B'_k} |c(b_k, b_e)|}, \tag{2.7}$$

where

$$B'_k := \{b_e \in B \,|\, b_e \in \text{prox}(b_k) \wedge r_i(b_e) \neq \perp\}$$

Intuitively, the approach tries to mimic real user behavior, having user a_i judge the value of an unknown product b_k by comparing the latter to known, similar items b_e and considering how much a_i appreciated these b_e.

The eventual computation of a top-N recommendation list P_{w_i} follows the user-based CF's process, arranging recommendations according to w_i in descending order.

2.3.3 Hybrid Recommender Systems

Hybrid approaches are geared towards unifying collaborative and content-based filtering under one single framework, leveraging synergetic effects and mitigating inherent deficiencies of either paradigm. Consequently, hybrid recommenders operate on both product rating information *and* descriptive features. In fact, numerous ways for combining collaborative and content-based aspects are conceivable, Burke [2002] lists an entire plethora of hybridization methods. Most widely adopted among these, however, is the so-called "collaboration via content" paradigm [Pazzani, 1999], where content-based profiles are built to detect similarities among users.

Sample Approaches

One of the earliest hybrid recommender systems is Fab [Balabanović and Shoham, 1997], which suggests Web pages to its users. Melville et al. [2002] and Hayes and Cunningham [2004] use content information for *boosting* the collaborative filtering process. Torres et al. [2004] and McNee et al. [2002] propose various hybrid systems for recommending citations of research papers. Huang et al. [2002, 2004] use content-based features in order to construct correlation graphs to explore *transitive* associations between users. Model-driven hybrid approaches have been suggested by Basilico and Hofmann [2004], proposing perceptron learning and kernel functions, and by Schein et al. [2002], using more traditional Bayesian classifiers.

2.4 Evaluating Recommender Systems

Evaluations of recommender systems are indispensable in order to quantify how *useful* recommendations made by system S_x are compared to S_y over the complete

set of users A. *Online* evaluations, i.e., directly asking users for their opinions, are, in most cases, not an option. Reasons are manifold:

Deployment. In order to perform online evaluations, an intact virtual community able to run recommender system services is needed. On the other hand, successfully deploying an online community and making it become self-sustaining is cumbersome and may exceed the temporal scope of most research projects.

Obtrusiveness. Even if an online community is readily available, evaluations cannot simply be performed at will. Many users may regard questionnaires as an additional burden, providing no immediate reward for themselves, and perhaps even decide to leave the system.

Hence, research has primarily relied upon *offline* evaluation methods, which are applicable to datasets containing past product ratings, such as, for instance, the well-known MovieLens and EachMovie datasets, both publicly available.[2] Machine learning cross-validation techniques are applied to these datasets, e.g., hold-out, K-folding, or leave-one-out testing, and evaluation metrics run upon. The following sections give an outline of popular metrics used for offline evaluations. An extensive and more complete survey is provided by Herlocker et al. [2004].

2.4.1 Accuracy Metrics

Accuracy metrics have been defined first and foremost for two major tasks: first, to judge the *accuracy* of single predictions, i.e., how much predictions $w_i(b_k)$ for products b_k deviate from a_i's actual ratings $r_i(b_k)$. These metrics are particularly suited for tasks where predictions are displayed along with the product, e.g., annotation in context [Herlocker et al., 2004]. Second, *decision-support* metrics evaluate the effectiveness of helping users to select high-quality items from the set of all products, generally supposing *binary preferences*.

2.4.1.1 Predictive Accuracy Metrics

Predictive accuracy metrics measure how close predicted ratings come to true user ratings. Most prominent and widely used [Shardanand and Maes, 1995; Herlocker et al., 1999; Breese et al., 1998; Good et al., 1999], mean absolute error (MAE) represents an efficient means to measure the statistical accuracy of predictions $w_i(b_k)$ for sets B_i of products:

$$|\overline{E}| = \frac{\sum_{b_k \in B_i} |r_i(b_k) - w_i(b_k)|}{|B_i|} \tag{2.8}$$

[2] See *http://www.grouplens.org* for EachMovie, MovieLens, and other datasets.

Related to MAE, mean squared error (MSE) *squares* the error before summing. Hence, large errors become much more pronounced than small ones. Very easy to implement, predictive accuracy metrics are inapt for evaluating the quality of top-N recommendation lists: users only care about errors for high-rank products. On the other hand, prediction errors for low-rank products are unimportant, knowing that the user has no interest in them anyway. However, MAE and MSE account for both types of errors in exactly the same fashion.

2.4.1.2 Decision-Support Metrics

Precision and recall, both well-known from information retrieval, do not consider predictions and their deviations from actual ratings. They rather judge how *relevant* a set of ranked recommendations is for the active user.

Typically, before using these metrics, K-folding is applied, dividing every user a_i's rated products $b_k \in R_i = \{b \in B \mid r_i(b) \neq \perp\}$ into K disjoint slices of preferably equal size. Folding parameters $K \in \{4, 5, \ldots, 10\}$ are commonly assumed. Next, $K-1$ randomly chosen slices are used to form a_i's *training set* R_i^x. 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, constitutes the *test set*, i.e., those products the recommendation algorithms intend to predict.

Precision, Recall, and F1

Sarwar et al. [2000b] present an 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 $|T_i^x|$:

$$\text{Recall} = 100 \cdot \frac{|T_i^x \cap \Im P_i^x|}{|T_i^x|} \tag{2.9}$$

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

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:

$$\text{Precision} = 100 \cdot \frac{|T_i^x \cap \Im P_i^x|}{|\Im P_i^x|} \tag{2.10}$$

Another popular metric used extensively in information retrieval and recommender systems research [Sarwar et al., 2000b; Huang et al., 2004; Montaner, 2003] is the standard F1 metric. F1 combines precision and recall in one single metric, giving equal weight to both of them:

$$\mathrm{F1} = \frac{2 \cdot \mathrm{Recall} \cdot \mathrm{Precision}}{\mathrm{Recall} + \mathrm{Precision}} \tag{2.11}$$

Breese Score

Breese et al. [1998] introduce an interesting extension to recall, known as weighted recall or Breese score. The underlying idea refers to the intuition 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 decay. The expected utility of a ranked list P_i^x of products is as follows:

$$H(P_i^x, T_i^x) = \sum_{b \in (T_i^x \cap \Im P_i^x)} \frac{1}{2^{(P_i^{x\,-1}(b)-1)/(\alpha-1)}} \tag{2.12}$$

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 follows:

$$\mathrm{BScore}(P_i^x, T_i^x) = 100 \cdot \frac{H(P_i^x, T_i^x)}{\sum_{k=1}^{|T_i^x|} \frac{1}{2^{(k-1)/(\alpha-1)}}} \tag{2.13}$$

Interestingly, when assuming $\alpha = \infty$, Breese score is identical to unweighted recall.

Other popular decision-support metrics include ROC [Schein et al., 2002; Melville et al., 2002; Good et al., 1999], the so-called *receiver operating characteristic*. ROC measures the extent to which an information filtering system is able to successfully distinguish between signal and noise. Less frequently used, NDPM [Balabanović and Shoham, 1997] compares two different, weakly ordered rankings.

2.4.2 Beyond Accuracy

Though accuracy metrics are an important facet of usefulness, there are traits of user satisfaction they are unable to capture. Still, non-accuracy metrics have largely been denied major research interest so far and have only been treated as marginally important supplements for accuracy metrics.

2.4.2.1 Coverage

Among all non-accuracy evaluation metrics, coverage has been the most frequently used [Herlocker et al., 1999; Middleton et al., 2004; Good et al., 1999]. Coverage measures the percentage of elements part of the problem domain for which predictions can be made.

For instance, supposing the user-based collaborative filtering approach presented in Section 2.3.2.1, coverage for the entire set of users is computed as follows:

$$\text{Coverage} = 100 \cdot \frac{\sum_{a_i \in A} \left| \{ b \in B \mid \exists a_j \in \text{prox}(a_i) : r_j(b) \neq \perp \} \right|}{|B| \cdot |A|} \tag{2.14}$$

2.4.2.2 Novelty and Serendipity

Some recommenders produce highly accurate results that are still useless in practice, e.g., suggesting bananas to customers in a grocery store: almost everyone appreciates bananas, so their recommending implies high accuracy. On the other hand, owing to their high popularity, most people *intuitively* purchase bananas upon entering a grocery store. They do not require an additional recommendation since they "already know" [Terveen and Hill, 2001].

Novelty and serendipity metrics thus measure the *non-obviousness* of recommendations made, penalizing "cherry-picking" [Herlocker et al., 2004].

Chapter 3

Taxonomy-driven Filtering

"We can love nothing but what agrees with us, and we can only follow our taste or our pleasure when we prefer our friends to ourselves."

– François de la Rochefoucauld (1694-1778)

Contents

3.1 Introduction

One of the primary issues that recommender systems are facing is rating sparsity, particularly pronounced for *decentralized scenarios* (see Section 1.2). Hence, high-quality product suggestions are only feasible when information density is high, i.e., large numbers of users voting for small numbers of items and issuing large numbers of

explicit ratings each. Small-sized, decentralized and open Web communities, where ratings are mainly derived *implicitly* from user behavior and interaction patterns, poorly qualify for blessings provided by recommender systems.

In this chapter, we explore an approach that intends to alleviate the information sparsity issue by exploiting *product classification taxonomies* as powerful background knowledge. Semantic product classification corpora for diverse fields are becoming increasingly popular, facilitating smooth interaction across company boundaries and fostering meaningful information exchange. For instance, the United Nations Standard Products and Services Classification (UNSPSC) contains over 11,000 codes [Obrst et al., 2003]. The taxonomies that Amazon.com (*http://www.amazon.com*) provides feature even more abundant, hierarchically arranged background knowledge: the book classification taxonomy alone comprises 13,500 topics, its pendant for categorizing movies and DVDs has about 16,400 concepts. Moreover, all products available on Amazon.com bear several descriptive terms referring to these taxonomies, thus making product descriptions machine-readable.

Our novel taxonomy-based similarity metric, making inferences from hierarchical relationships between classification topics, represents the core of our hybrid filtering framework to compensate for sparsity. Quality recommendations become feasible in communities suffering from low information density, too.

We collected and crawled data from the very sparse All Consuming book readers' community (*http://www.allconsuming.net*) and conducted various experiments indicating that the taxonomy-driven method significantly outperforms benchmark systems. Repeating the offline evaluations for the dense MovieLens dataset, our approach's performance gains shrink considerably, but still exceed benchmark scores.

3.2 Related Work

Many attempts have been made to overcome the sparsity issue. Sarwar et al. [2000b] propose singular value decomposition (SVD) as an efficient means to reduce the dimensionality of product-user matrices in collaborative filtering. Results reported have been mixed. Personal filtering agents [Good et al., 1999], surrogates for human users, represent another approach and have been shown to slightly improve results when deployed into sparse, human-only communities. Srikumar and Bhasker [2004] combine association rule mining and user-based CF to cope with sparsity.

The idea of using taxonomies for information filtering has been explored before, the most prominent example being directory-based browsing of information mines, e.g., Yahoo (*http://www.yahoo.com*), Google Directory (*http://www.google.com*), and ACM Computing Reviews (*http://www.reviews.com*). Moreover, Sollenborn and Funk [2002] propose *category*-based filtering, similar to the approach pursued by [Baudisch, 2001]. Pretschner and Gauch [1999] personalize Web search by using ontologies that represent user interests for profiling.

However, these taxonomy-based approaches do not exploit semantic "is-a" relationships between topics for profiling. Middleton et al. [2001, 2002] recommend research papers, using ontologies to inductively learn topics that users are particularly interested in. Knowing a user's most liked topics then allows efficient product set filtering, weeding out those research papers that do not fall into these favorite topics. In contrast to our own technique proposed, Middleton uses clustering techniques for categorization and does not make use of human-created, large-scale product classification taxonomies.

3.3 Approach Outline

Following the "collaboration via content" paradigm [Pazzani, 1999], our approach computes content-based user profiles which are then used to discover like-minded peers. Once the active 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 b depends on two factors. First, the similarity weight of neighbors voting for b, and, second, b's content description with respect to the active user's interest profile. Hence the hybrid nature of our approach.

3.3.1 Information Model

Before delving into algorithmic details, we introduce the formal information model, which can be tied easily to arbitrary application domains. Note that the model at hand also serves as foundation for subsequent chapters.

- **Agents** $A = \{a_1, a_2, \ldots, a_n\}$**.** All community members are elements of A. Possible identifiers are globally unique names, URIs, etc.
- **Product set** $B = \{b_1, b_2, \ldots, b_m\}$**.** All domain-relevant products are stored in set B. Unique identifiers either refer to proprietary product codes from an online store, such as Amazon.com's ASINs, or represent globally accepted standard codes, e.g., ISBNs.
- **User ratings** $R_1, R_2, \ldots, R_n$**.** Every agent a_i is assigned a set $R_i \subseteq B$ which contains his *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 [Nichols, 1998].
- **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 $d_e, d_h \in D, e \neq h$, and hence impose tree-like structuring, similar to single-inheritance class hierarchies known from object-oriented languages. Leaf topics d_e are topics with zero outdegree, formally $C(d_e) = \bot$, 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 may be too imprecise.

3.3.2 Taxonomy-driven Profile Generation

Collaborative filtering techniques represent user profiles by vectors $\vec{v_i} \in [-1, +1]^{|B|}$, 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 or cosine similarity to their respective profile vectors (see Section 2.3.2). Clearly, for very large $|B|$ and comparatively small $|A|$, this representation fails, owing to insufficient overlap of rating vectors.

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: 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:

$$\forall a_i \in A : \sum_{k=1}^{|D|} v_{i,k} = s \tag{3.1}$$

By virtue of agent-wise normalization for a_i's profile, the 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, the 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 $d_{k_e} \in f(b_k)$. Then 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 $p_0 = \top$:

$$\sum_{m=0}^{q} \mathrm{sco}(p_m) = \mathrm{sc}(d_{k_e}) \tag{3.2}$$

In addition, based on results obtained from research on semantic distance in taxonomies (e.g., see [Budanitsky and Hirst, 2000] and [Resnik, 1999]), 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 fewer siblings p_{m+1} possesses, the more interest score is accorded to its super-topic node p_m:

$$\mathrm{sco}(p_m) = \kappa \cdot \frac{\mathrm{sco}(p_{m+1})}{\mathrm{sib}(p_{m+1}) + 1} \tag{3.3}$$

We 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 [Resnik, 1995, 1999].

Propagation factor κ permits to fine-tune 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.

Equations 3.2 and 3.3 describe conditions which have to hold for the computation of leaf node p_q's profile score $\mathrm{sco}(p_q)$ and the computation of scores for its taxonomy ancestors p_k, where $k \in \{0, 1, \ldots, q-1\}$. We hence derive the following recursive definition for $\mathrm{sco}(p_q)$:

$$\mathrm{sco}(p_q) := \kappa \cdot \frac{\mathrm{sc}(d_{k_e})}{g_q}, \tag{3.4}$$

where

$$g_0 := 1, \;\; g_1 := 1 + \frac{1}{\mathrm{sib}(p_q) + 1},$$

and $\forall n \in \{2, \ldots, q\}$

$$g_n := g_{n-1} + (g_{n-1} - g_{n-2}) \cdot \frac{1}{\mathrm{sib}(p_{q-n+1}) + 1}$$

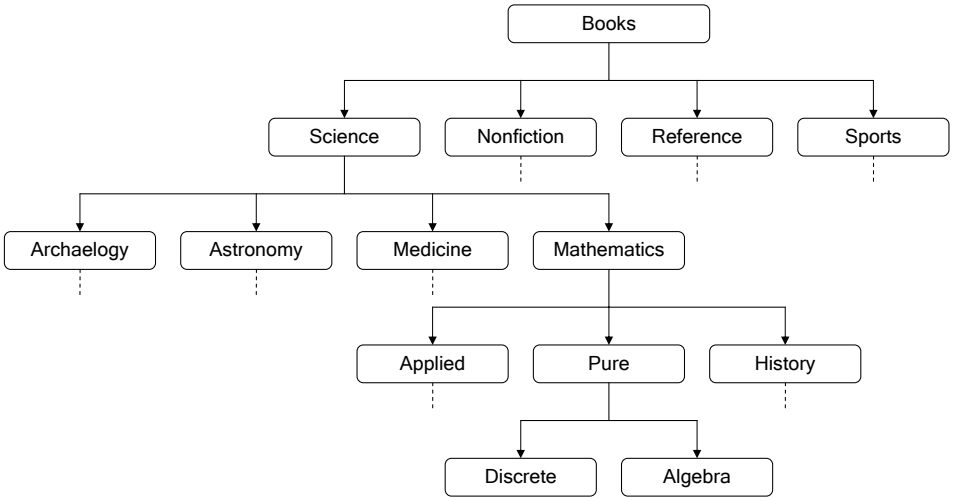

Figure 3.1. Fragment from the Amazon.com book taxonomy

Computed scores $\text{sco}(p_m)$ are used to build a profile vector $\vec{v_i}$ for 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 computation) Suppose taxonomy C as depicted in Figure 3.1, 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. Therefore, score 50 must be distributed among these topics according to Equation 3.2 and 3.3. The application of Equation 3.4 yields score 29.091 for topic ALGEBRA. Likewise, applying Equation 3.3, we get 14.545 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 $\vec{v_i}$ for a_i.

3.3.3 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.

3.3.3.1 Measuring Proximity

Pearson's correlation coefficient and cosine similarity count among the most prominent correlation measures for CF (see Section 2.3.2.1). For our taxonomy-driven method, we opted for Pearson correlation, which Herlocker et al. [2002] have found to perform better on collaborative filtering than cosine similarity.

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. Clearly, 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[1] 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 mined from All Consuming, we had one user reading books mainly from the genres of Science Fiction, Fantasy, and Artificial Intelligence. The person in question was negatively correlated to another one reading books about American History, Politics, and Conspiracy Theories.

3.3.3.2 Selecting Neighbors

Having computed proximity weights $c(a_i, a_j)$ for the active user a_i and agents $a_j \in A \setminus \{a_i\}$, neighborhood formation takes place. Agent a_i's neighborhood, denoted by $\text{prox}(a_i)$, contains a_i's most similar peers for use in computing recommendation lists.

Herlocker et al. [1999] name 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 $\text{prox}(a_i)$, whereas best-M-neighbors picks the M best correlates for a_i's neighborhood.

[1] Leaf nodes of HISTORY are not shown in Figure 3.1.

We opted for best-M-neighbors, since correlation-thresholding implies diverse unwanted effects when sparsity is high [Herlocker et al., 1999].

3.3.4 Recommendation Generation

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

$$B_i = \bigcup \{R_j \mid a_j \in \text{prox}(a_i)\} \setminus R_i \tag{3.5}$$

Candidates $b_k \in B_i$ are then weighted according to their *relevance* for a_i. 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 (see Section 1.1.1).
- **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 likewise important. The purely content-based metric 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 $c_b(a_i, b_k)$ derives from the user similarity computation scheme. For this purpose, we create a *dummy user* a_θ with $R_\theta = \{b_k\}$ and define $c_b(a_i, b_k) := c(a_i, a_\theta)$.

The relevance $w_i(b_k)$ of product b_k for the active user a_i is then defined by the following formula:

$$w_i(b_k) = \frac{q \cdot c_b(a_i, b_k) \cdot \sum_{a_j \in A_i(b_k)} c(a_i, a_j)}{|A_i(b_k)| + \Upsilon_R}, \tag{3.6}$$

where

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

and

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

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 $a_j \in \text{prox}(a_i)$. 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* [Sollenborn and Funk, 2002]. For experimental analysis, we tried values between 0 and 2.5.

Parameter Γ_T rewards products b_k that carry *many* content descriptors, i.e., have 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. Reward through Γ_T is assigned linearly by virtue of $(|f(b_k)| \cdot \Gamma_T)$. Consider that 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.

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 according to w_i in *descending* order. For top-N recommendations, all entries $P_{w_i}(k), k > N$ are discarded.

3.3.5 Topic Diversification

A technique we call *topic diversification* constitutes another cornerstone contribution of this chapter. The latter method represents an *optional* procedure to supplement recommendation generation and to enhance the computed list's utility for agent a_i.

The idea underlying topic diversification 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 using content-based and hybrid filtering techniques. 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.

3.3.5.1 Recommendation Dependency

In order to implement topic diversification, we assume that recommended products $P_{w_i}(o)$ and $P_{w_i}(p)$, 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.

To our best knowledge, Brafman et al. [2003] are the only researchers assuming dependencies 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 3.1 depicts the entire procedure, a brief textual sketch is given in the next few paragraphs.

3.3.5.2 Topic Diversification Algorithm

Function P_{w_i*} denotes the new recommendation list, resulting from the application of 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 the mentioned similarity measure, denoted $c^*(b)$, by applying our scheme for taxonomy-driven profile generation and proximity measuring presented in Section 3.3.2 and 3.3.3.1.

Sorting all products b according to $c^*(b)$ in reverse order, we obtain the *dissimilarity rank* $P_{c^*}^{\text{rev}}$. 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 $P_{c^*}^{\text{rev}}$ 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 diversification factors $\Theta_F \in [0, 0.9]$.

Note that the ordered input lists P_{w_i} must be *considerably larger* than the eventual top-N list. Algorithm 3.1 uses constant x for that purpose. In our later experiments, we assumed $x = 4$, hence using top-80 input lists for final top-20 recommendations.

procedure diversify ($P_{w_i} : \{1, \ldots, |B_i|\} \rightarrow B$, $\Theta_F \in [0, 1]$) {
  $B_i \leftarrow \{P_{w_i}(k) \mid k \in [1, x \cdot N]\}$; $P_{w_i*}(1) \leftarrow P_{w_i}(1)$;
  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**
      $P_{c^*}^{\text{rev}^{-1}}(b) \leftarrow |B_i'| - P_{c^*}^{-1}(b)$;
      $w_i^*(b) \leftarrow P_{w_i}^{-1}(b) \cdot (1 - \Theta_F) + P_{c^*}^{\text{rev}^{-1}}(b) \cdot \Theta_F$;
    end do
    $P_{w_i*}(z) \leftarrow \min\{w_i^*(b) \mid b \in B_i'\}$;
  end do
  return P_{w_i*};
}

Algorithm 3.1. Sequential topic diversification

3.3.5.3 Osmotic Pressure Analogy

The effect of dissimilarity bears traits similar to that of osmotic pressure and selective permeability known from molecular biology (e.g., see Tombs [1997]): steady insertion of products b_o, taken from one specific area of interest d_o, into the recommendation list equates to the passing of molecules from one specific substance through the cell membrane into cytoplasm. With increasing concentration of d_o, owing to the membrane's selective permeability, the pressure for molecules b from other substances d rises. When pressure gets sufficiently high for one given topic d_p, its best products b_p may "diffuse" into the recommendation list, even though the original rank $P_{w_i}^{-1}(b_p)$ might be inferior to the rank of candidates from the prevailing domain d_o. Consequently, pressure for d_p decreases, paving the way for another domain for which pressure peaks.

Topic diversification hence resembles the membrane's selective permeability, which allows cells to maintain their internal composition of substances at required levels.

3.4 Offline Experiments and Evaluation

The following sections present empirical results that were obtained from evaluating our approach. Core engine parts of our system, along with most other software tools

for data extraction and screen scraping, were implemented in Java, small portions in Perl. Remote access via Web interfaces was rendered feasible through PHP frontends.

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

3.4.1 Data Acquisition

Experimentation, parameterization, and fine-tuning were conducted on "real-world" data, obtained from All Consuming (*http://www.allconsuming.net*), 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. Crawling was started on January 12 and finished on January 16, 2004.

The entire dataset comprises 2,783 users, representing either "real", registered members of All Consuming, or personal weblogs collected by the community's spiders, 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 the respective books. On average, users provided 5.24 book ratings.

After the application of various data cleansing procedures and duplicate removal, Amazon.com's tree-structured book classification taxonomy contained 13,525 distinct concepts. Our crawling tools collected 27,202 topic descriptors from Amazon.com, relating 8,641 books to the latter concept lattice. Consequently, for 596 of those 9,237 books mentioned by All Consuming's users, no content information could be obtained from Amazon.com, signifying only 6.45% rejects. 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[2] 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.

3.4.2 Evaluation Framework

Since our taxonomy-driven recommender system operates on *binary* preference input, i.e., implicit rather than explicit ratings, predictive accuracy metrics (see Section 2.4.1.1) are not suitable for evaluation. We hence opted for decision-support accuracy metrics (see Section 2.4.1.2), namely precision, recall, and Breese score.

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

3.4.2.1 Benchmark Systems

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

Bottom Line Definition

For any given user a_i, the naïve system randomly selects an item $b \in B \setminus R_i$ for a_i'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, representing the base case that "non-naïve" algorithms are supposed to surpass.

Collaborative Filtering Algorithm

The common user-based CF algorithm, featuring extensions proposed by Herlocker et al. [2002], traditionally serves as benchmark when evaluating recommender systems operating on *explicit* preferences. Sarwar et al. [2000b] propose an adaptation specifically geared towards implicit ratings, known as "most frequent items". We used that latter system as CF benchmark, computing relevance weights $w_i(b_k)$ for books b_k from a_i's candidates set B_i according to the following scheme:

$$w_i(b_k) = \sum_{a_j \in A_i(b_k)} c(a_i, a_j) \tag{3.7}$$

Set $A_i(b_k) \subseteq \text{prox}(a_i)$ contains all neighbors of a_i who have implicitly rated b_k.

We measure user similarity $c(a_i, a_j)$ according to Pearson correlation (see Section 2.3.2.1). 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 $\vec{v_i}, \vec{v_j} \in \{0, 1\}^{|B|}$.

Hybrid Recommender Approach

The third system exploits both collaborative and content-based filtering techniques, representing user profiles $\vec{v_i}$ through collections of descriptive terms, along with their frequency of occurrence.

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

$$\forall d \in D : v_{i,d} = |\{b_k \in R_i \,|\, d \in f(b_k)\}| \tag{3.8}$$

Neighborhoods are formed by computing Pearson correlations between all pairs of content-driven profile vectors and selecting best-M neighbors. Relevance is then defined as below:

$$w_i(b_k) = \frac{c_b(a_i, b_k) \cdot \sum_{a_j \in A_i(b_k)} c(a_i, a_j)}{|A_i(b_k)|} \tag{3.9}$$

Mind that Equation 3.9 presents a *special case* of Equation 3.6, assuming $\Gamma_T = 0$ and $\Upsilon_R = 0$. Essentially, the depicted hybrid approach constitutes a simplistic adaptation of our taxonomy-driven system. Differences largely refer to the underlying algorithm's lack of super-topic score inference, one major cornerstone of our novel method, and the lack of parameterization.

3.4.2.2 Experiment Setup

The evaluation framework intends to compare the *utility* of recommendation lists generated by all four recommender systems, applying precision, recall, and Breese score (see Section 2.4.1.2). In order to obtain *global* metrics, we averaged the respective metric values for all evaluated users.

First, we selected all users a_i with more than five ratings and discarded those having fewer ratings, owing to the fact that reasonable recommendations are beyond feasibility for these cases.

For cross-validation, we applied 5-folding, effectively performing 80/20 splits of every user a_i's implicit ratings R_i into five pairs of *training sets* R_i^x and *test sets* T_i^x, where $T_i^x = R_i \setminus R_i^x$. Consequently, we computed five complete recommendation lists for every a_i, i.e., one list for each $R_i^x, x \in \{1, \ldots, 5\}$.

3.4.2.3 Parameterization

We defined $|\operatorname{prox}(a_i)| = 20$, i.e., requiring neighborhoods to contain exactly 20 peers, and we provided top-20 recommendations for each active user a_i's training set R_i^x. Similarities between profiles, based upon R_i^x and the original ratings R_j of all other agents a_j, were computed anew for each training set R_i^x of a_i.

For performance analysis, 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 adopted 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_k bearing *detailed* content descriptions, i.e., having large $|f(b_k)|$, by assuming $\Gamma_T = 0.1$. Hence, a 10%

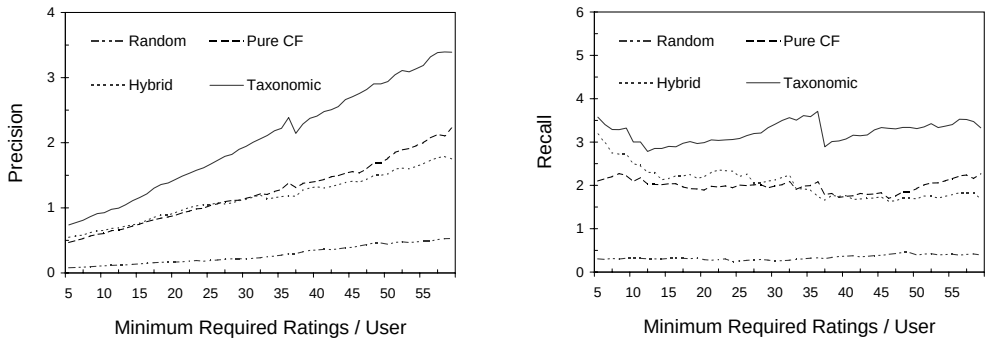

Figure 3.2. Unweighted precision and recall metrics

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.

3.4.2.4 Result Analysis

We measured performance by computing precision, recall, and Breese score, assuming half-life $\alpha = 5$ and $\alpha = 10$, for all four recommenders and all combinations of training and test sets. Results are displayed in Figure 3.2 and 3.3.

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. Note that larger x-coordinates hence imply that *fewer* 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.

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 the taxonomy-driven recommendation generation, these agents have high ranks with respect to all four metrics applied. Removal thus temporarily lowers the curves.

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

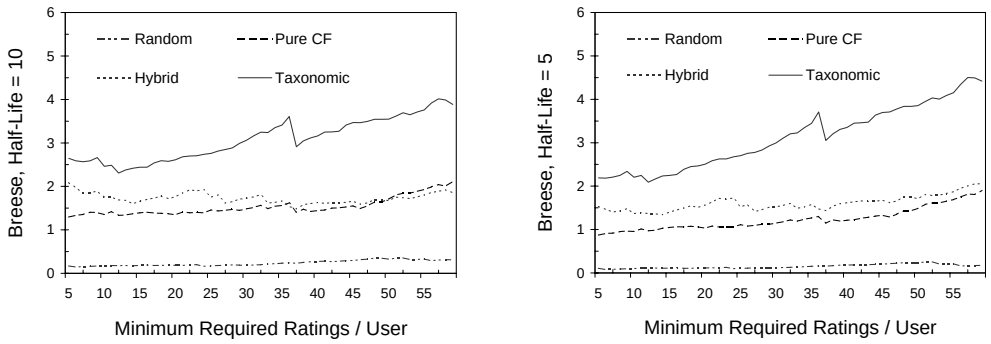

Figure 3.3. Weighted recall, using half-life $\alpha = 10$ and $\alpha = 5$

Precision

Surprisingly, precision increases even for the random recommender when ignoring users with few 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 *no possibility* of achieving 100% precision.

Analysis of unweighted precision, given on the left-hand side of Figure 3.2, shows that the gap between our taxonomy-driven approach and its collaborative and hybrid pendants becomes even larger when users are required to have rated many books. Agents with small numbers of ratings tend to perturb 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 steadily outperforms the former when making recommendations for agents with numerous ratings.

Unweighted and Weighted Recall

Unweighted recall, shown on the right-hand side of Figure 3.2, presents a slightly different scenario: even though the performance gap between the taxonomy-driven recommender and both other, non-naïve methods still persists, it 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 [1999].

Figure 3.3 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 $\alpha = \infty$ (see Section 2.4.1.2). While pure collaborative filtering shows largely

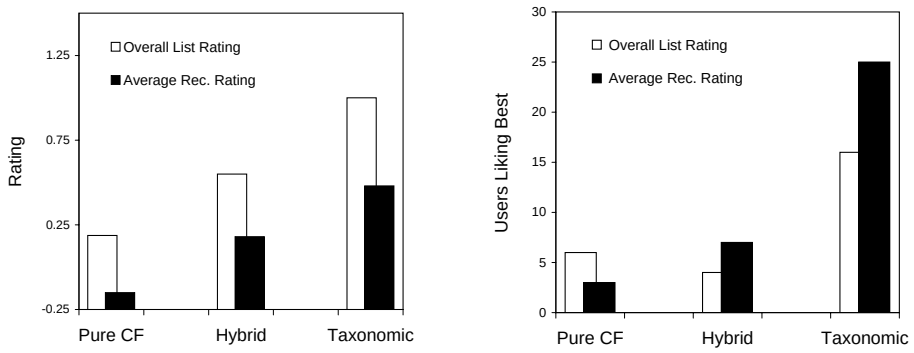

Figure 3.4. Results obtained from the online study

insensitive to decreasing α, hybrid and taxonomy-driven recommenders do not. Assuming $\alpha = 10$, the first derivative of the latter two systems improves over their corresponding recall curves for increasing x-coordinates. This notable development becomes even more pronounced when further decreasing half-life to $\alpha = 5$.

Consequently, in case of content-exploiting methods, 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, for collaborative filtering, relevant products seem to be more uniformly distributed among top-20 ranks.

3.5 Deployment and Online Study

On February 9, 2004, we deployed our taxonomy-driven recommender system into the All Consuming community[3], providing 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.

Besides our taxonomy-driven approach, we also implanted both other non-naïve approaches documented before into All Consuming. Registered users could hence access *three distinct lists* of top-20 recommendations, customized according to their personal rating profile. We utilized the depicted system setup to conduct online performance comparisons, going beyond offline statistical measures.

[3]The computed recommendation lists can be reached through All Consuming's *News*-section, see *http://cgi.allconsuming.net/news.html*.

3.5.1 Online Study Setup

For the 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.

3.5.2 Result Analysis

54 All Consuming members, not affiliated with our department and university, volunteered to participate in our online study by December 3, 2004. They provided 2,164 ratings about recommendations they were offered, and 131 additional, overall list quality statements. 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 50 users rated one or more recommendations computed according to the purely collaborative method, named A, 46 did so for the taxonomy-driven approach, labelled B, and 42 for the simplistic hybrid algorithm. In a first experiment, depicted on the left side of Figure 3.4, we compared the overall recommendation list statements 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, the comparison was based upon the overall statements and average recommendation ratings, likewise. In order to guarantee fairness, we discarded users not having rated all three systems for each metric. The right-hand chart of Figure 3.4 shows that the taxonomy-driven method outperforms both other recommendation techniques.

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

3.6 Movie Data Analysis

The dataset we obtained from crawling the All Consuming community exhibits two properties we believe pivotal for the superiority of our technique over common benchmark methods:

- **Rating information sparseness.** Compared to the number of ratings, the number of unique ISBNs is relatively large. Moreover, most users have issued few ratings, these being implicit only. Hence, the probability of having product rating profiles with numerous overlapping products is low, implying bad performance scores for standard collaborative filtering approaches. On the other hand, taxonomy-driven profiling has been conceived to directly address these issues and to render sparse profile vectors dense.
- **Fine-grained domain classification.** Books address most aspects of our everyday life, e.g., education, business, science, entertainment, and so forth. Therefore, the construction of highly nested and detailed classification taxonomies becomes feasible. Moreover, owing to comparatively high *costs of consumption*[4], people *deliberately* consume products matching their specific interests only. Inspection of purchase and book reading histories clearly reveals these diverse interests and makes profile compositions easily discernable, which is essential for finding appropriate neighbors.

However, we would like to test our approach on domains where the two aforementioned assumptions do not hold anymore. We hence opted for the popular MovieLens dataset [Sarwar et al., 2001, 2000b], which contains *explicit* ratings about movies and has a very high density.

Movies bear intrinsic features that make them largely different from books. For instance, their cost of consumption tends to be much lower. Consequently, people are more inclined to experience products that may not perfectly match their profile of interest. We conjecture that such exploratory behavior makes interest profiles, inferred from implicit or explicit ratings, less concise and less accurate.

Moreover, movies are basically geared towards the entertainment sector only, not spanning other areas of life, e.g., science, business, and so forth. We believe both aspects disadvantageous for taxonomy-driven profiling.

3.6.1 Dataset Composition

The small MovieLens dataset contains 943 users who have issued 100,000 explicit ratings on a 5-point likert scale, referring to 1,682 movies. The average number of ratings per user hence amounts to 106.04, meaning that the average user has rated 6.31% of all ratable products. These numbers highly contrast All Consuming's figures, where the average user has rated 5.24 books and thus only 0.04% of the entire product set.

In order to make taxonomy-driven recommendations feasible, we crawled Amazon.com's movie taxonomy, extracting 16,481 hierarchically arranged topics. This number clearly exceeds the book taxonomy's 13,525 concepts. In addition, both

[4]Note that reading books takes much more time than watching DVDs or listening to CDs.

lattices exhibit subtly different characteristics with respect to structure: the movie taxonomy's average distance from root to leafs amounts to 4.25, opposed to 5.05 for books. However, the average number of inner node siblings is higher for movies than for books, contrasting 18.53 with 16.65. Hence, we may conclude that the book taxonomy is deeper, though more condensed, than its movie pendant.

We were able to obtain taxonomic descriptions for 1519 of all 1682 movies on MovieLens from Amazon.com, collecting 9281 descriptors in total. On average, 5.52 topic descriptors were found for those movies for which content information could be provided. The remaining 163 movies were removed from the dataset, along with all 8668 ratings referring to them.

3.6.2 Offline Experiment Framework

We opted for roughly the same analysis setup as presented for the All Consuming offline evaluations. Since MovieLens features explicit ratings, we denote user a_i's ratings by function $r_i : B \to \{1, 2, \ldots, 5\}^{\perp}$ rather than $R_i \subseteq B$. We tailored our evaluation metrics and benchmark recommenders in order to account for explicit ratings.

3.6.2.1 Benchmark Systems

The parameters for the taxonomy-driven approach were slightly modified in order to optimize results. For topic diversification, we supposed $\Theta_F = 0.25$. Super-topic score inference was promoted by assuming $\kappa = 1.0$. Moreover, only movies b_k that had been assigned an excellent rating of 4 or 5, i.e., $b_k \in \{b \in B \mid r_i(b) \geq 4\}$, were considered for the generation of a_i's profile.

The random-based recommender was kept in order to mark the absolute bottom line.

Collaborative Filtering Algorithm

Instead of "most frequent items" [Sarwar et al., 2000b], we used the original GroupLens collaborative recommender [Konstan et al., 1997; Resnick et al., 1994], which had been specifically designed with *explicit* ratings in mind (see Section 2.3.2.1). We extended the system by implementing modifications proposed by Herlocker et al. [1999], i.e., significance weighting, deviation from mean, and best-M neighborhood formation, in order to render the recommender as competitive as possible. We found that the application of significance weighting, i.e., penalizing high correlation values based upon less than 50 products in common, increased the system's performance *substantially*.

Most Popular Products Recommender

Breese et al. [1998] compare benchmarks against an efficient, though non-personalized recommender. The algorithm simply proposes overall top-N most rated products to the active user a_i. However, these products are required not to occur in a_i's training set R_i^x, i.e., $P_i^x \cap R_i^x = \emptyset$, implying that recommendation lists P_i^x, P_j^x for two different users a_i, a_j are not completely identical.

3.6.2.2 Setup and Metrics

Again, we applied 5-folding cross-validation and assumed neighborhoods of dimension $|\operatorname{prox}(a_i)| = 20$ for all users a_i. In contrast to the All Consuming experimental setup, we provided top-10 recommendations instead of top-20.[5]

Moreover, the input test sets we provided to precision and recall slightly differed from the input provided in preceding experiments: in order to account for the fact that all ratings were *explicit*, i.e., that we actually *knew* if user a_i had liked product b_k experienced earlier, only test set products $b_k \in \{b \in T_i^x \mid r_i(b) \geq 4\}$ were counted as hits, i.e., those products that a_i had assigned *positive* ratings:

$$\text{Recall} = 100 \cdot \frac{|\Im P_i^x \cap \{b \in T_i^x \mid r_i(b) \geq 4\}|}{|\{b \in T_i^x \mid r_i(b) \geq 4\}|} \tag{3.10}$$

Accordingly, the computation of precision with rating-constrained test set input looks as follows:

$$\text{Precision} = 100 \cdot \frac{|\Im P_i^x \cap \{b \in T_i^x \mid r_i(b) \geq 4\}|}{|\Im P_i^x|} \tag{3.11}$$

We also computed F1 scores (see Section 2.4.1.2), based upon the above-given versions of precision and recall.

3.6.2.3 Result Analysis

Precision and recall values considering the complete 943 users dataset were computed for all four recommender systems. The respective scores are given by Figure 3.5. One can see that the obtained metric values tend to be *higher* than their equivalents for the All Consuming community data. Apart from the random recommender[6], all algorithms achieved more than 10% recall and 12% precision. The reasons for these comparatively high scores lie primarily in the much larger density of MovieLens as opposed to All Consuming, indicated before in Section 3.6.1.

[5] We found little variation in precision/recall scores when decreasing the recommendation list size from 20 to 10.

[6] The random recommender maintained precision/recall values far below 1% and is not displayed in Figure 3.5.

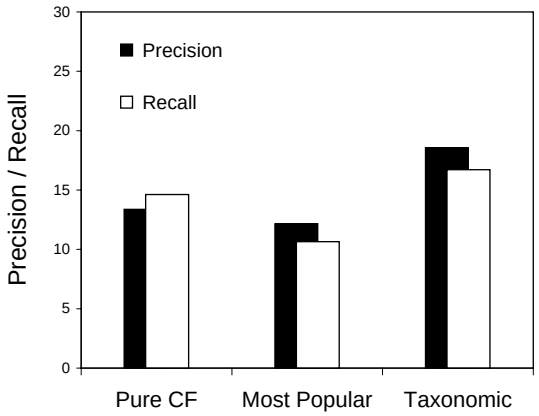

Figure 3.5. Performance analysis for the complete MovieLens dataset

The taxonomy-driven approach, having an F1 metric score of 17.59%, outperforms the purely collaborative system as well as the non-personalized recommender for top-N most popular products. The latter method also shows inferior to the collaborative filter, made explicit by an F1 score of 11.34% versus 13.98%.

However, the relative performance gap between our taxonomy-driven approach and its benchmark recommenders is definitely more pronounced for the All Consuming book rating data than for MovieLens. Conjectures about possible reasons have already been mentioned in Section 3.6, counting domain-dependence and rating sparsity among the major driving forces.

Dataset Size Sensitivity

In a second experiment, we tested the *sensitivity* of all presented non-naïve recommenders with respect to the numbers of users eligible for neighborhood formation. Neither the product set size nor the number of ratings per user were modified. We created 8 subsets of the MovieLens user base, selecting the first $x \cdot 50$ users from the complete set, $x \in \{1, 2, \ldots, 8\}$. Results are shown in Figure 3.6.

For the smallest set, i.e., $|A| = 50$, the non-personalized recommender for overall most appreciated products shows to be the best predictor, while the purely collaborative filtering system performs worst among the three non-random recommenders. However, for 100 users already, the two personalized approaches overtake the non-personalized system and exhibit steadily increasing F1 scores for increasing numbers of users x. Interestingly, the gradient for the taxonomy-driven method's curve is still slightly superior to the collaborative filtering recommender's.

We regard this observation as an indication that neighborhood formation relying

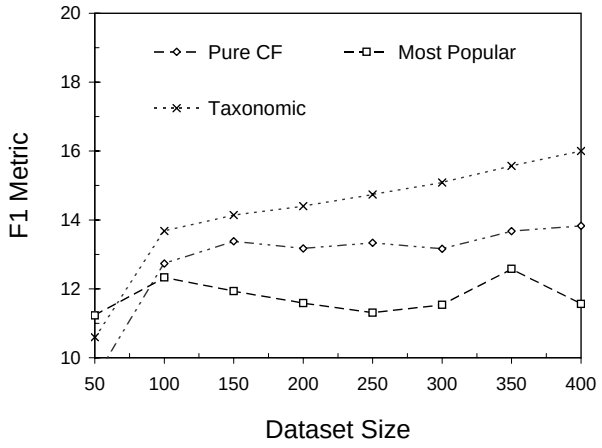

Figure 3.6. MovieLens dataset size sensitivity

upon taxonomy-based user profiles makes sense for denser rating data, too. The accuracy still does not degrade below the purely collaborative system's benchmark, even though the gap appears much smaller than for sparser rating information scenarios.

3.7 Conclusion

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

Offline 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 conducted online studies, asking All Consuming community members to rate and compare diverse recommender systems. In addition to sparse book rating information, we tested our approach's performance when dealing with substantially different data, running benchmark comparisons on the well-known MovieLens dataset. Results suggested that taxonomy-driven recommending still performs better on denser data than competing systems. However, the performance gap becomes comparatively small and does no longer justify additional efforts for acquiring costly domain knowledge, which taxonomy-driven filtering substantially depends on.

Chapter 4

Topic Diversification Revisited

"All life is an experiment. The more experiments you make the better."

– Ralph Waldo Emerson (1803–1883)

Contents

4.1 Introduction

Chapter 3 has introduced topic diversification as an efficient means to avoid *topic overfitting* in our taxonomy-driven filtering approach. However, the topic diversification method can be applied to *any* recommender system that generates ordered top-N lists of recommendations, as long as taxonomic domain knowledge is available for the recommendation domain in question.

Winners Take All

The main reason for the a posteriori application of topic diversification to conventional recommender systems lies in the fact that most recommender algorithms are highly susceptible to *winners-take-all* behavior: soon as the user's profile bucket contains one subset of similar products that appears larger than any other similarity-based subset, the chances that *all* computed recommendations will derive from that cluster are high. The observation can be made for algorithms using content-based similarity measures, and techniques based on collaborative similarity metrics, e.g.,

item-based CF (see Section 2.3.2.2), likewise. For instance, many people complain that Amazon.com's (*http://www.amazon.com*) recommendations, computed according to the item-based CF scheme [Linden et al., 2003], appear too "similar" with respect to content. Hence, customers that have purchased many books written by Herrmann Hesse may happen to obtain recommendation lists where *all* top-5 entries contain books from that respective author only. When considering pure accuracy, all these recommendations seem excellent, since the active user clearly appreciates Hesse's novels. On the other hand, assuming that the active user has several interests other than Herrmann Hesse, e.g., historical novels in general and books about world travel, the recommended set of items appears poor, owing to its lack of diversity.

Some researchers, e.g., Ali and van Stam [2004], have noticed the depicted issue, commonly known as the "portfolio effect", for other recommender systems before. However, to our best knowledge, no solutions have been proposed so far.

Reaching Beyond Accuracy

Topic diversification can solve the portfolio effect issue, balancing and diversifying personalized recommendation lists to reflect the user's *entire spectrum* of interests. However, when running offline evaluations based upon accuracy metrics such as precision, recall, and MAE (see Section 2.4.1), we may expect the performance of topic diversification-enhanced filters to show *inferior* to that of their respective non-diversified pendants. Hence, while believed beneficial for actual user satisfaction, we conjecture that topic diversification will prove detrimental to accuracy metrics.

For evaluation, we therefore pursue a twofold approach, conducting one large-scale online study that involves more than 2,100 human subjects, and offline analysis runs based on 361,349 ratings. Both evaluation methods feature the application of diverse degrees of diversification to the two most popular recommendation techniques, i.e., user-based and item-based CF (see Section 2.3.2). The bilateral evaluation approach renders the following types of result analysis possible:

- **Accuracy and diversity.** The application of precision and recall metrics to user-based and item-based CF with varying degrees of diversification, $\Theta_F \in [0.1, 0.9]$, exposes the *negative* impacts that topic diversification exerts on accuracy. Proposing the offline *intra-list similarity* measure, we are able to capture and quantify the diversity of top-N recommendation lists, with respect to one given similarity metric. Contrasting the measured accuracy and diversity, their overall *negative* correlation becomes revealed.
- **Topic diversification benefits and limitations.** The online study shows that users tend to appreciate diversified lists. For diversification factors $\Theta_F \in [0.3, 0.4]$ (see Section 3.3.5.2), satisfaction significantly exceeds the respective non-diversified cases. However, online results also reveal that too much diversification, $\Theta_F \in [0.6, 0.9]$, appears harmful and detrimental to user satisfaction.

- **Accuracy versus satisfaction.** Several researchers have argued that "accuracy does not tell the whole story" [Cosley et al., 2002; Herlocker et al., 2004]. Nevertheless, no evidence has been given to show that some aspects of actual user satisfaction reach beyond accuracy. We close this gap by contrasting our online and offline results, showing that for $\Theta_F \rightarrow 0.4$, accuracy deteriorates while satisfaction improves.

4.2 Related Work

Few efforts have addressed the problem of making top-N lists more diverse. Considering literature on collaborative filtering and recommender systems in general only, none have been presented before, to our best knowledge.

However, some work related to our topic diversification approach can be found in information retrieval, specifically meta-search engines. A critical aspect of meta-search engine design is the merging of several top-N lists into one single top-N list. Intuitively, this merged top-N list should reflect the highest quality ranking possible, also known as the "rank aggregation problem" [Dwork et al., 2001]. Most approaches use variations of the "linear combination of score" model (LC), described by Vogt and Cottrell [1999]. The LC model effectively resembles our scheme for merging the original, accuracy-based ranking with the current dissimilarity ranking, but is more general and does not address the diversity issue. Fagin et al. [2003] propose metrics for measuring the distance between top-N lists, i.e., inter-list similarity metrics, in order to evaluate the quality of merged ranks. Oztekin et al. [2002] extend the linear combination approach by proposing rank combination models that also incorporate content-based features in order to identify the most relevant topics.

More related to our idea of creating lists that represent the whole plethora of the user's topic interests, Kummamuru et al. [2004] present their clustering scheme that groups search results into clusters of related topics. The user can then conveniently browse topic folders relevant for his search interest. The commercially available search engine Northern Light (*http://www.northernlight.com*) incorporates similar functionalities. Google (*http://www.google.com*) uses several mechanisms to suppress top-N list items that are too similar in content, showing them only upon the user's explicit request. Unfortunately, no publications on that matter are available.

4.3 Empirical Analysis

We conducted offline evaluations to understand the ramifications of topic diversification on accuracy metrics, and online analysis to investigate how our method affects actual user satisfaction. We applied topic diversification with factors $\Theta_F \in \{0, 0.1, 0.2, \ldots 0.9\}$ to lists generated by both user-based CF and item-based CF,

observing effects that occur when steadily increasing Θ_F and analyzing how both approaches respond to diversification.

For online as well as offline evaluations, we used data gathered from BookCrossing (*http://www.bookcrossing.com*). This community caters for book lovers exchanging books around the world and sharing their experiences with other readers.

Data Collection

In a 4-week crawl, we collected data about 278,858 members of BookCrossing and 1,157,112 ratings, both implicit and explicit, referring to 271,379 distinct ISBNs. Invalid ISBNs were excluded from the outset.

The complete BookCrossing dataset, anonymized for privacy reasons, is available via the author's homepage (*http://www.informatik.uni-freiburg.de/~cziegler/BX/*).

Next, we mined Amazon.com's book taxonomy, comprising 13,525 distinct topics. In order to be able to apply topic diversification, we mined supplementary content information, focusing on taxonomic descriptions that relate books to taxonomy nodes from Amazon.com (*http://www.amazon.com*). Since many books on BookCrossing refer to rare, non-English books, or outdated titles not in print anymore, we were able to garner background knowledge for only 175,721 books. In total, 466,573 topic descriptors were found, giving an average of 2.66 topics per book.

Condensation Steps

Owing to the BookCrossing dataset's extreme sparsity, we decided to *condense* the set in order to obtain more meaningful results from CF algorithms when computing recommendations. Hence, we discarded all books missing taxonomic descriptions, along with all ratings referring to them. Next, we also removed book titles with fewer than 20 overall mentions. Only community members with at least 5 ratings each were kept.

The resulting dataset's dimensions were considerably more moderate, comprising 10,339 users, 6,708 books, and 361,349 book ratings.

4.3.1 Offline Experiments

We performed offline experiments comparing precision, recall, and intra-list similarity scores for 20 different recommendation list setups. Half these lists were based upon user-based CF with different degrees of diversification, the others on item-based CF. Note that we did not compute MAE metric values since we are dealing with implicit rather than explicit ratings.

The before-mentioned *intra-list similarity metric* intends to capture the *diversity* of a list. Diversity may refer to all kinds of features, e.g., genre, author, and other discerning characteristics. Based upon an arbitrary function $c_{ILS} : B \times B \rightarrow$

$[-1, +1]$ measuring the similarity $c_{ILS}(b_k, b_e)$ between products b_k, b_e according to some custom-defined criterion, we define intra-list similarity for an agent a_i's list P_{w_i} as follows:

$$\mathrm{ILS}(P_{w_i}) = \frac{\sum_{b_k \in \Im P_{w_i}} \sum_{b_e \in \Im P_{w_i}, b_k \neq b_e} c_{ILS}(b_k, b_e)}{2} \tag{4.1}$$

Higher metric scores express lower diversity. An interesting mathematical feature of $\mathrm{ILS}(P_{w_i})$ we are referring to in later sections is *permutation-insensitivity*, i.e., let S_N be the symmetric group of all permutations on $N = |P_{w_i}|$ symbols:

$$\forall \sigma_i, \sigma_j \in S_N : \mathrm{ILS}(P_{w_i} \circ \sigma_i) = \mathrm{ILS}(P_{w_i} \circ \sigma_j) \tag{4.2}$$

Hence, simply rearranging positions of recommendations in a top-N list P_{w_i} does not affect P_{w_i}'s intra-list similarity.

4.3.1.1 Experiment Setup

For cross-validation of precision and recall metrics of all 10,339 users, we adopted 4-folding. Hence, rating profiles R_i were effectively split into training sets R_i^x and test sets $T_i^x, x \in \{1, \ldots, 4\}$, at a ratio of 3 : 1. For each of the 41,356 different training sets, we computed 20 top-10 recommendation lists.

To generate the diversified lists, we computed top-50 lists based upon pure, i.e., non-diversified, item-based CF and pure user-based CF. The high-performance SUGGEST recommender engine[1] was used to compute these base case lists. Next, we applied the diversification algorithm to both base cases, applying Θ_F factors ranging from 10% up to 90%. For eventual evaluations, all lists were truncated to contain 10 books only.

4.3.1.2 Result Analysis

We were interested in seeing how accuracy, captured by precision and recall, behaves when increasing Θ_F from 0.1 up to 0.9. Since topic diversification may make books with high predicted accuracy trickle down the list, we hypothesized that accuracy will *deteriorate* for $\Theta_F \rightarrow 0.9$. Moreover, in order to find out if our novel algorithm has any significant, positive effects on the diversity of items featured, we also applied our intra-list similarity metric. An overlap analysis for diversified lists, $\Theta_F \geq 0.1$, versus their respective non-diversified pendants indicates how many items stayed the same for increasing diversification factors.

[1] Visit *http://www-users.cs.umn.edu/∼karypis/suggest/* for further details.

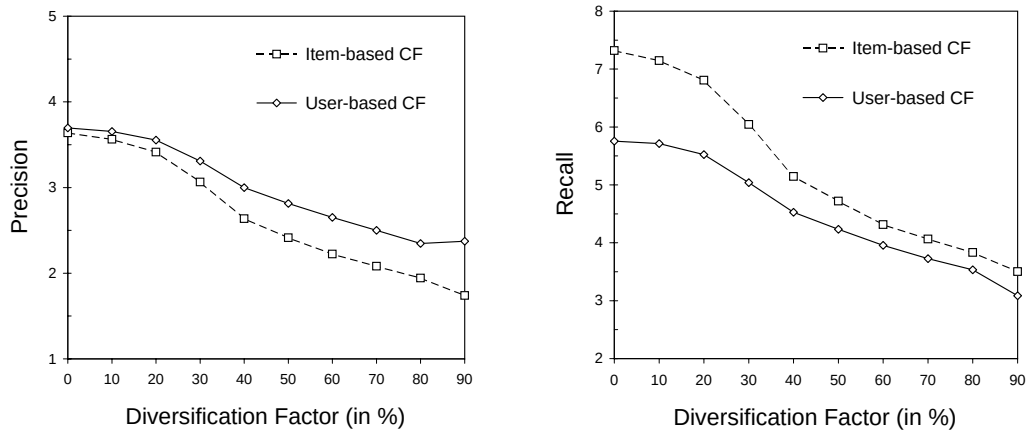

Figure 4.1. Precision and recall metrics for increasing Θ_F

Precision and Recall

First, we analyzed precision and recall for both non-diversified base cases, i.e., when $\Theta_F = 0$. Table 4.1 states that user-based and item-based CF exhibit almost identical accuracy, indicated by precision values. Their recall values differ considerably, hinting at deviating behavior with respect to the types of users they are scoring for.

	Item-based CF	**User-based CF**
Precision	3.64	3.69
Recall	7.32	5.76

Table 4.1. Precision and recall for non-diversified CF

Next, we analyzed the behavior of user-based and item-based CF when steadily increasing Θ_F by increments of 10%, depicted by Figure 4.1. The two charts reveal that diversification has detrimental effects on *both* metrics and on *both* CF algorithms. Interestingly, corresponding precision and recall curves have almost identical shape.

The loss in accuracy is more pronounced for item-based than for user-based CF. Furthermore, for either metric and either CF algorithm, the drop is most distinctive for $\Theta_F \in [0.2, 0.4]$. For lower Θ_F, negative impacts on accuracy are marginal. We believe this last observation due to the fact that precision and recall are permutation-

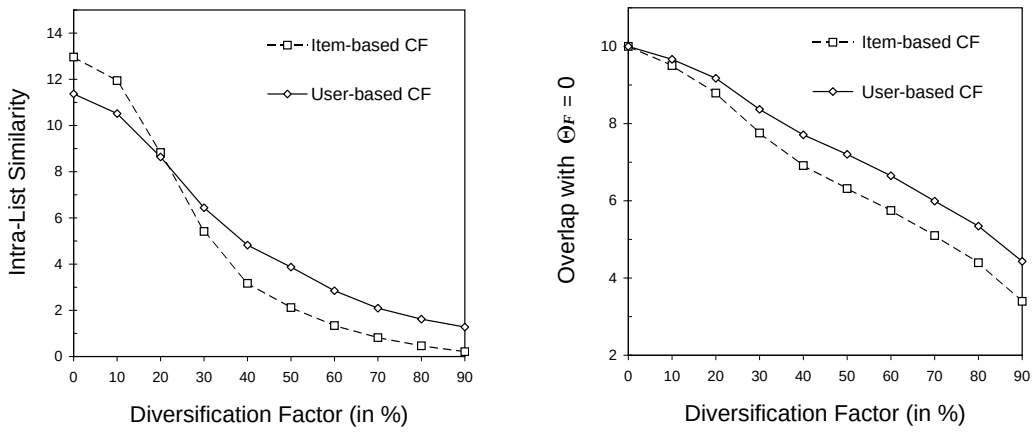

Figure 4.2. Intra-list similarity and original list overlap for increasing Θ_F

insensitive, i.e., the mere order of recommendations within a top-N list does not influence the metric value, as opposed to Breese score (see Section 2.4.1.2). However, for low Θ_F, the pressure that the dissimilarity rank exerts on the top-N list's makeup is still too weak to make many new items diffuse into the top-N list. Hence, we conjecture that rather the *positions* of current top-N items change, which does not affect either precision or recall.

Intra-List Similarity

Knowing that our diversification technique bears a significant, *negative* impact on accuracy metrics, we wanted to know how our approach affected the intra-list similarity measure. Similar to the precision and recall experiments, we computed metric values for user-based and item-based CF with $\Theta_F \in [0, 0.9]$ each. We instantiated the metric's embedded similarity function c_{ILS} with our taxonomy-driven metric c^*, defined in Section 3.3.5.2. Results obtained are provided by Figure 4.2.

The topic diversification method considerably lowers the pairwise similarity between list items, thus making top-N recommendation lists more diverse. Diversification appears to affect item-based CF stronger than its user-based counterpart, in line with our findings about precision and recall. For lower Θ_F, curves are less steep than for $\Theta_F \in [0.2, 0.4]$, which also well aligns with our precision and recall analysis. Again, the latter phenomenon can be explained by one of the metric's inherent features: like precision and recall, intra-list similarity is permutation-insensitive.

Original List Overlap

The right-hand side of Figure 4.2 depicts the number of recommended items staying the same when increasing Θ_F with respect to the original list's content. Both curves exhibit roughly linear shapes, being less steep for low Θ_F, though. Interestingly, for factors $\Theta_F \leq 0.4$, at most 3 recommendations change on average.

Conclusion

We found that diversification appears largely detrimental to both user-based and item-based CF along precision and recall metrics. In fact, this outcome aligns with our expectations, considering the nature of those two accuracy metrics and the way that the topic diversification method works. Moreover, we found that item-based CF seems more susceptible to topic diversification than user-based CF, backed by results from precision, recall and intra-list similarity metric analysis.

4.3.2 User Survey

Offline experiments helped us in understanding the implications of topic diversification on both CF algorithms. We could also observe that the effects of our approach are different on different algorithms. However, knowing about the deficiencies of accuracy metrics, we wanted to assess *real* user satisfaction for various degrees of diversification, thus necessitating an online survey.

For the online study, we computed each recommendation list type anew for users in the denser BookCrossing dataset, though without K-folding. In cooperation with BookCrossing, we mailed all eligible users via the community mailing system, asking them to participate in our online study. Each mail contained a personal link that would direct the user to our online survey pages. In order to make sure that only the users themselves would complete their survey, links contained unique, encrypted access codes.

During the 3-week survey phase, 2,125 users participated and completed the study.

4.3.2.1 Survey Outline and Setup

The survey consisted of several screens that would tell the prospective participant about this study's nature and his task, show all his ratings used for making recommendations, and would finally present a top-10 recommendation list, asking several questions thereafter.

For each book, users could state their interest on a 5-point rating scale. Scales ranged from "not much" to "very much", mapped to values 1 to 4, and offered the user to indicate that he had "already read the book", mapped to value 5. In order to successfully complete the study, users were *not* required to rate all their top-10 recommendations. Neutral values were assumed for non-votes instead. However, we

required users to answer all further questions, concerning the list as a whole rather than its single recommendations, before submitting their results. We embedded those questions we were actually keen about knowing into ones of lesser importance, in order to conceal our intentions and not bias users.

The one top-10 recommendation list for each user was chosen among 12 candidate lists, either user-based or item-based featuring no diversification, i.e., $\Theta_F = 0$, medium levels, $\Theta_F \in \{0.3, 0.4, 0.5\}$, and high diversification, $\Theta_F \in \{0.7, 0.9\}$. We opted for those 12 instead of all 20 list types in order to acquire enough users completing the survey for each slot. The assignment of a specific list to the current user was done dynamically, at the time of the participant entering the survey, and in a round-robin fashion. Thus, we could guarantee that the number of users per list type was roughly identical.

4.3.2.2 Result Analysis

For the analysis of our inter-subject survey, we were mostly interested in the following three aspects. First, the *average rating* users gave to their 10 single recommendations. We expected results to roughly align with scores obtained from precision and recall, owing to the very nature of these metrics. Second, we wanted to know if users perceived their list as well-diversified, asking them to tell whether the lists reflected rather a broad or narrow *range of their reading interests*. Referring to the intra-list similarity metric, we expected the users' perceived range of topics, i.e., the list's diversity, to increase with increasing Θ_F. Third, we were curious about the *overall satisfaction* of users with their recommendation lists in their entirety, the measure to compare performance.

Both last-mentioned questions were answered by each user on a 5-point likert scale, higher scores denoting better performance. Moreover, we averaged the eventual results by the number of users. Statistical significance of all mean values was measured by parametric one-factor ANOVA (see, e.g., [Armitage and Berry, 2001]), where $p < 0.05$ is assumed if not indicated otherwise.

Single-Vote Averages

Users perceived recommendations made by user-based CF systems on average as more accurate than those made by item-based CF systems, as depicted in Figure 4.3(a). At each featured diversification level Θ_F, the differences between the two CF types are statistically significant, $p \ll 0.01$.

Moreover, for each algorithm, higher diversification factors obviously entail lower single-vote average scores, which confirms our hypothesis stated before. The item-based CF's cusp at $\Theta_F \in [0.3, 0.5]$ appears as a notable outlier, opposed to the trend, but differences between the 3 means at $\Theta_F \in [0.3, 0.5]$ are not statistically significant, $p > 0.15$. However, the differences between all factors Θ_F are significant

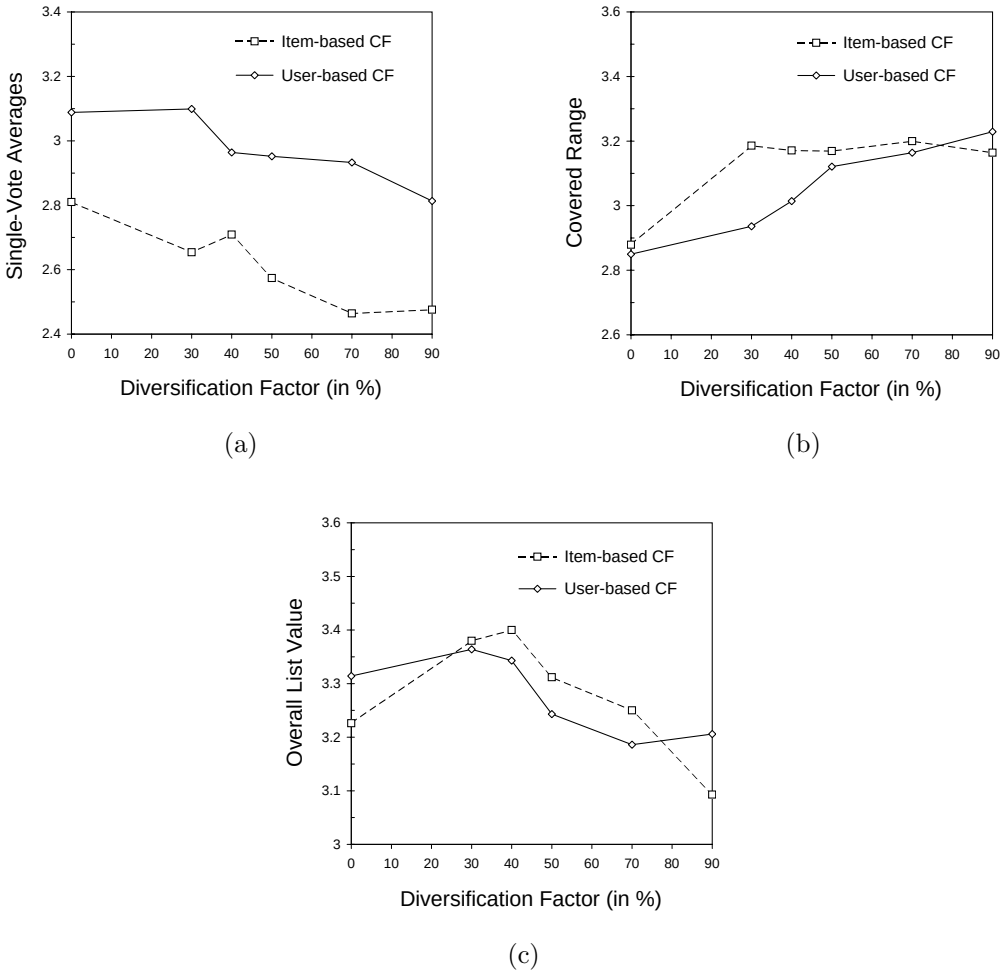

Figure 4.3. Single-vote averages (a), covered range (b), and overall value (c)

for item-based CF, $p \ll 0.01$, and for user-based CF, $p < 0.1$.

Hence, topic diversification *negatively* correlates with pure accuracy. Besides, users perceived the performance of user-based CF as significantly better than item-based CF for all corresponding levels Θ_F.

Covered Range

Next, we analyzed whether the users actually *perceived* the variety-augmenting effects caused by topic diversification, illustrated before through measurement of intra-list similarity. Users' reactions to steadily incrementing Θ_F are displayed in Figure 4.3(b). First, between both algorithms on corresponding Θ_F levels, only the differ-

ence of means at $\Theta_F = 0.3$ shows statistical significance.

Studying the trend of user-based CF for increasing Θ_F, we notice that the perceived range of reading interests covered by users' recommendation lists also increases. Hereby, the curve's first derivative maintains an approximately constant level, exhibiting slight peaks between $\Theta_F \in [0.4, 0.5]$. Statistical significance holds for user-based CF between means at $\Theta_F = 0$ and $\Theta_F > 0.5$, and between $\Theta_F = 0.3$ and $\Theta_F = 0.9$.

Contrary to that observation, the item-based curve exhibits a drastically different behavior. While soaring at $\Theta_F = 0.3$ to 3.186, reaching a score almost identical to the user-based CF's peak at $\Theta_F = 0.9$, the curve barely rises for $\Theta_F \in [0.4, 0.9]$, remaining rather stable and showing a slight, though insignificant, upward trend. Statistical significance was shown for $\Theta_F = 0$ with respect to all other samples from $\Theta_F \in [0.3, 0.9]$. Hence, our online results do not perfectly align with findings obtained from offline analysis. While the intra-list similarity chart in Figure 4.2 indicates that diversity increases when increasing Θ_F, the item-based CF chart defies this trend, first soaring then flattening. We conjecture that the following three factors account for these peculiarities:

Diversification factor impact. Offline results for the intra-list similarity metric already indicated that the impact of topic diversification on item-based CF is much stronger than on user-based CF. Consequently, the item-based CF's user-perceived interest coverage is significantly higher at $\Theta_F = 0.3$ than the user-based CF's.

Human perception. We believe that human perception can capture the level of diversity inherent to a list only to some extent. Beyond that point, increasing diversity remains unnoticed. For the application scenario at hand, Figure 4.3 suggests this point around score value 3.2, reached by user-based CF only at $\Theta_F = 0.9$, and approximated by item-based CF already at $\Theta_F = 0.3$.

Interaction with accuracy. Analyzing results obtained, we have to bear in mind that covered range scores are *not* fully independent from single-vote averages. When accuracy is poor, i.e., the user feels unable to identify recommendations that are interesting to him, chances are high his discontentment will also negatively affect his diversity rating. For $\Theta_F \in [0.5, 0.9]$, single-vote averages are remarkably low, which might explain why perceived coverage scores do not improve for increasing Θ_F.

However, we may conclude that users *do* perceive the application of topic diversification as an overly positive effect on reading interest coverage.

Overall List Value

The third feature variable we were evaluating, the overall value users assigned to their personal recommendation list, effectively represents the "target value" of our

studies, measuring actual user satisfaction. Owing to our conjecture that user satisfaction is a mere composite of accuracy and other influential factors, such as the list's diversity, we hypothesized that the application of topic diversification would *increase* satisfaction. At the same time, considering the downward trend of precision and recall for increasing Θ_F, in accordance with declining single-vote averages, we expected user satisfaction to drop off for large Θ_F. Hence, we supposed an arc-shaped curve for both algorithms.

Results for the overall list value are provided by Figure 4.3(c). Analyzing user-based CF, we observe that the curve does *not* follow our hypothesis. Slightly improving at $\Theta_F = 0.3$ over the non-diversified case, scores drop for $\Theta_F \in [0.4, 0.7]$, eventually culminating in a slight but visible upturn at $\Theta_F = 0.9$. While lacking reasonable explanations and being opposed to our hypothesis, the curve's data-points actually bear no statistical significance for $p < 0.1$. Hence, we conclude that topic diversification has a marginal, largely negligible impact on overall user satisfaction, initial positive effects eventually being offset by declining accuracy.

On the contrary, for item-based CF, results obtained look very different. In compliance with our previous hypothesis, the curve's shape roughly follows an arc, peaking at $\Theta_F = 0.4$. Taking the three data-points defining the arc, we obtain statistical significance for $p < 0.1$. The endpoint's score at $\Theta_F = 0.9$ being inferior to the non-diversified case's, we observe that too much diversification appears detrimental, most likely owing to substantial interactions with accuracy.

Eventually, for overall list value analysis, we come to conclude that topic diversification has no measurable effects on user-based CF, but significantly improves item-based CF performance for diversification factors Θ_F around 40%.

4.3.2.3 Multiple Linear Regression

Results obtained from analyzing user feedback along various feature axes already indicated that users' overall satisfaction with recommendation lists not only depends on accuracy, but also on the range of reading interests covered. In order to more rigidly assess that indication by means of statistical methods, we applied *multiple linear regression* to our survey results, choosing the overall list value as dependent variable. As independent input variables, we provided single-vote averages and covered range, both appearing as first-order and second-order polynomials, i.e., SVA and CR, and SVA^2 and CR^2, respectively. We also tried several other, more complex models, without achieving significantly better model fitting.

Analyzing multiple linear regression results, shown in Table 4.2, confidence values $P(> |t|)$ clearly indicate that statistically significant correlations for accuracy and covered range with user satisfaction exist. Since statistical significance also holds for their respective second-order polynomials, i.e., CR^2 and SVA^2, we conclude that these relationships are non-linear and more complex, though.

As a matter of fact, linear regression delivers a strong indication that the intrinsic

utility of a list of recommended items is more than just the average value of accuracy votes for all single items, but also depends on the perceived diversity.

	Estimate	**Error**	***t*-Value**	$P(>\|t\|)$
(const)	3.27	0.023	139.56	$< 2e-16$
SVA	12.42	0.973	12.78	$< 2e-16$
$\mathbf{SVA}^2$	-6.11	0.976	-6.26	$4.76e-10$
CR	19.19	0.982	19.54	$< 2e-16$
$\mathbf{CR}^2$	-3.27	0.966	-3.39	0.000727
	Multiple R^2: 0.305, adjusted R^2: 0.303			

Table 4.2. Multiple linear regression results

4.3.3 Limitations

There are some limitations to the study, notably referring to the way topic diversification was implemented. Though the Amazon.com taxonomies were human-created, there might still be some mismatch between what the topic diversification algorithm perceives as "diversified" and what humans do. The issue is effectively inherent to the taxonomy's structure, which has been designed with *browsing tasks* and ease of searching rather than with interest profile generation in mind. For instance, the taxonomy features topic nodes labelled with letters for alphabetical ordering of authors from the same genre, e.g., Books → Fiction → ... → Authors, A-Z → G. Hence, two Sci-Fi books from two different authors with the same initial of their last name would be classified under the same node, while another Sci-Fi book from an author with a *different* last-name initial would *not*. Though the problem's impact is largely marginal, owing to the relatively deep level of nesting where such branchings occur, the procedure appears far from intuitive.

An alternative approach to further investigate the accuracy of taxonomy-driven similarity measurement, and its limitations, would be to have *humans* do the clustering, e.g., by doing card sorts or by estimating the similarity of any two books contained in the book database. The results could then be matched against the topic diversification method's output.

4.4 Summary

This chapter provided empirical analyses in order to evaluate the application of our topic diversification method to common collaborative filtering algorithms, and introduced the novel *intra-list similarity* metric.

Contrasting precision and recall metrics, computed both for user-based and item-based CF and featuring different levels of diversification, with results obtained from a large-scale user survey, we showed that the user's overall liking of recommendation lists goes beyond accuracy and involves other factors, e.g., the users' perceived list diversity. We were thus able to provide empirical evidence that lists are *more* than mere aggregations of single recommendations, but bear an intrinsic, added value.

Though effects of diversification were largely marginal on user-based CF, item-based CF performance improved significantly, an indication that there are some behavioral differences between both CF classes. Moreover, while pure item-based CF appeared slightly inferior to pure user-based CF in overall satisfaction, diversifying item-based CF with factors $\Theta_F \in [0.3, 0.4]$ made item-based CF outperform user-based CF. Interestingly, for $\Theta_F \leq 0.4$, no more than three items changed with respect to the original list, shown in Figure 4.2. Small changes thus have high impact.

We believe our findings especially valuable for practical application scenarios, knowing that many commercial recommender systems, eg., Amazon.com [Linden et al., 2003] and TiVo [Ali and van Stam, 2004], are item-based, owing to the algorithm's computational efficiency. For these commercial systems, topic diversification could be an interesting supplement, increasing user satisfaction and thus the customers' incentive to purchase recommended goods.

Chapter 5

Trust Propagation Models

"Perhaps there is no single variable which so thoroughly influences interpersonal and group behavior as does trust."

– Golembiewski and McConkie, 1975

Contents

5.1 Introduction

While previous chapters have primarily presented methods to overcome several specific recommender systems issues, the current chapter moves into another direction and focuses on *trust metrics*, i.e., network-based tools for predicting the extent of interpersonal trust shared between two human subjects. Though not directly related to recommender systems research, the contributions made therein are of utter relevance for their later integration into the decentralized recommender framework. Our main contributions are the following:

Trust metric classification scheme. We analyze existing trust metrics and classify them according to three non-orthogonal features axes. Advantages and draw-

backs with respect to decentralized scenarios are discussed and we formulate an advocacy for local group trust metrics.

Appleseed trust metric. Compelling in its simplicity, our Appleseed local group trust metric borrows many ideas from spreading activation models [Quillian, 1968], taken from cognitive psychology, and relates their concepts to trust evaluation in an intuitive fashion. Moreover, extensions are provided that make our trust metric handle *distrust* statements, likewise.

On Trust and Trust Propagation

In our world of information overload and global connectivity leveraged through the Web and other types of media, social trust [McKnight and Chervany, 1996] between individuals becomes an invaluable and precious good. Trust exerts an enormous impact on decisions whether to believe or disbelieve information asserted by other peers. Belief should only be accorded to statements from people we deem trustworthy. Hence, trust assumes the role of an instrument for "complexity reduction" [Luhmann, 1979]. However, when supposing huge networks such as the Semantic Web, trust judgements based on personal experience and acquaintanceship become unfeasible. In general, we accord trust, defined by Mui et al. [2002] as the "subjective expectation an agent has about another's future behavior based on the history of their encounters", to only small numbers of people. These people, again, trust another limited set of people, and so forth. The network structure emanating from our person (see Figure 5.1), composed of trust statements linking individuals, constitutes the basis for trusting people we do not know personally. Playing an important role for the conception of decentralized infrastructures, e.g., the Semantic Web, the latter structure has been dubbed the "Web of Trust" [Golbeck et al., 2003].

Its effectiveness has been underpinned through empirical evidence from social psychology and sociology, indicating that *transitivity* is an important characteristic of social networks [Holland and Leinhardt, 1972; Rapoport, 1963]. To the extent that communication between individuals becomes motivated through positive affect, the drive towards transitivity can also be explained in terms of Heider's famous "balance theory" [Heider, 1958], i.e., individuals are more prone to interact with friends of friends than unknown peers.

Adopting the most simple policy of trust propagation, all those people who are trusted by persons we trust are considered likewise trustworthy. Trust would thus propagate through the network and become accorded whenever two individuals can reach each other via at least one trust path. However, owing to certain implications of interpersonal trust, e.g., attack-resistance, trust decay, etc., more complex metrics are needed to sensibly evaluate social trust. Subtle social and psychological aspects must be taken into account and specific criteria of computability and scalability satisfied.

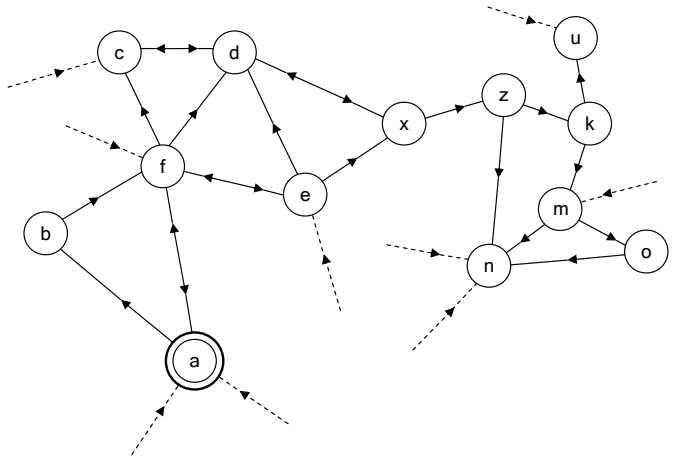

Figure 5.1. Sample web of trust for agent a

In this chapter, we aim at designing one such complex trust metric[1], particularly tailored to social filtering tasks (see Section 2.3) by virtue of its ability to infer continuous trust values through fixpoint iteration, rendering ordered trust-rank lists feasible. Before developing our trust metric model, we analyze existing approaches and arrange them into a new classification scheme.

5.2 Computational Trust in Social Networks

Trust represents an invaluable and precious good one should award deliberately. Trust metrics compute quantitative *estimates* of how much trust an agent a_i should accord to his peer a_j, taking into account trust ratings from other persons on the network. These metrics should also act "deliberately", not overly awarding trust to persons or agents whose trustworthiness is questionable.

5.2.1 Classification of Trust Metrics

Applications for trust metrics and trust management [Blaze et al., 1996] are rife. First proposals for metrics date back to the early nineties, where trust metrics have

[1]Note that trust concepts commonly adopted for webs of trust, and similar trust network applications, are largely general and do not cover specifics such as "situational trust" [Marsh, 1994a], as has been pointed out before [Golbeck et al., 2003]. For instance, agent a_i may blindly trust a_j with respect to books, but not trust a_j with respect to trusting others, for a_j has been found to accord trust to other people too easily. For our trust propagation scheme at hand, we also suppose this largely uni-dimensional concept of trust.

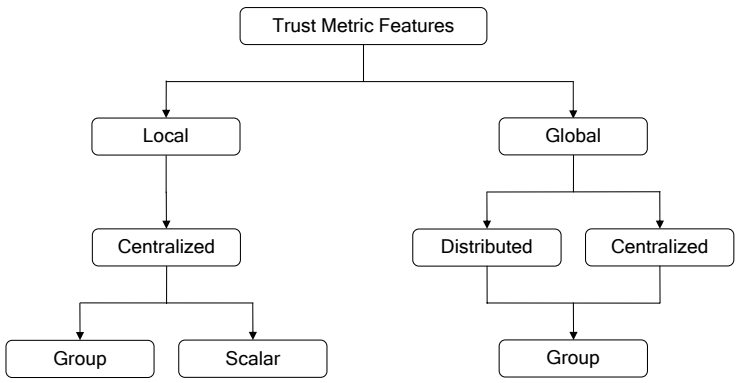

Figure 5.2. Trust metric classification

been deployed in various projects to support the "Public Key Infrastructure" (PKI) [Zimmermann, 1995]. The metrics proposed by Levien and Aiken [1998], Reiter and Stubblebine [1997b], Maurer [1996], and Beth et al. [1994] count among the most popular ones for public key authentication. New areas and research fields apart from PKI have come to make trust metrics gain momentum. Peer-to-peer networks, ubiquitous and mobile computing, and rating systems for online communities, where maintenance of explicit certification authorities is not feasible anymore, have raised the research interest in trust. The whole plethora of available metrics can hereby be defined and characterized along various classification axes. We identify three principal dimensions, namely *network perspective*, *computation locus*, and *link evaluation*. These axes are *not* orthogonal, though, for various features impose restrictions on the feature range of other dimensions (see Figure 5.2).

Network Perspective

The first dimension impacts the *semantics* assigned to the values computed. Trust metrics may basically be subdivided into those with *global*, and those with *local* scope. Global trust metrics take into account *all* peers and trust links connecting them. Global trust ranks are assigned to an individual based upon complete trust graph information. Many global trust metrics, such as those presented by Kamvar et al. [2003], Guha [2003], and Richardson et al. [2003], borrow their ideas from the renowned PageRank algorithm [Page et al., 1998] to compute Web page reputation, and to some lesser extent from HITS [Kleinberg, 1999]. The basic intuition behind these approaches is that nodes should be ranked higher the better the rank of nodes pointing to them. Obviously, the latter notion likewise works for trust and page reputation.

Trust metrics with local scope, on the other hand, take into account personal bias. Interestingly, some researchers claim that only local trust metrics are "true" trust metrics, since global ones compute overall reputation rather than personalized trust[2] [Mui et al., 2002]. Local trust metrics take the agent for whom to compute trust as an additional input parameter and are able to operate on *partial* trust graph information. The rationale behind local trust metrics is that persons an agent a_i trusts may be completely different from the range of individuals that agent a_j deems trustworthy. Local trust metrics exploit structural information defined by personalized webs of trust. Hereby, the personal web of trust for individual a_i is given through the set of trust relationships emanating from a_i and passing through nodes he trusts either directly or indirectly, as well as the set of nodes reachable through these relationships. Merging all webs of trust engenders the global trust graph. Local trust metrics comprise Levien's Advogato trust metric [Levien and Aiken, 2000], metrics for modelling the PKI [Beth et al., 1994; Maurer, 1996; Reiter and Stubblebine, 1997b] and the Semantic Web trust infrastructure [Golbeck and Hendler, 2004], and Sun Microsystems's Poblano [Chen and Yeager, 2003]. The latter work borrows from Abdul-Rahman and Hailes [1997].

Computation Locus

The second axis refers to the place where trust relationships between individuals are evaluated and quantified. Local[3] or *centralized* approaches perform all computations in one single machine and hence need to be granted full access to all relevant trust information. The trust data itself may be distributed over the network. Most of the before-mentioned metrics count among the class of centralized approaches.

Distributed metrics for the computation of trust and reputation, such as those described by Richardson et al. [2003], Kamvar et al. [2003], and Sankaralingam et al. [2003], equally deploy the load of computation on every trust node in the network. Upon receiving trust information from his predecessor nodes in the trust graph, an agent a_i merges the data with his own trust assertions and propagates synthesized values to his successor nodes. The entire process of trust computation is necessarily asynchronous and its convergence depends on the eagerness or laziness of nodes to propagate information. Another characteristic feature of distributed trust metrics refers to the fact that they are inherently global. Though the individual computation load is lower with respect to centralized computation approaches, nodes need to store trust information about *any other* node in the system.

[2]Recall the definition of trust given before, expressing that trust is a "subjective expectation".

[3]Note that in this context, the term "local" refers to the *place of computation* and not the network perspective.

Link Evaluation

The third dimension distinguishes scalar and group trust metrics. According to Levien [2004], *scalar* metrics analyze trust assertions independently, while *group trust* metrics evaluate groups of assertions "in tandem". PageRank [Page et al., 1998] and related approaches count among global group trust metrics, for the reputation of one page depends on the ranks of referring pages, thus implying the parallel evaluation of relevant nodes, thanks to mutual dependencies. Advogato [Levien and Aiken, 2000] represents an example for local group trust metrics. Most other trust metrics are scalar ones, tracking trust paths from sources to targets and not performing parallel evaluations of groups of trust assertions. Hence, another basic difference between scalar and group trust metrics refers to their functional design. In general, scalar metrics compute trust between two given individuals a_i and a_j, taken from set A of all agents.

On the other hand, group trust metrics generally compute trust ranks for *sets* of individuals in A. Hereby, global group trust metrics assign trust ranks for every $a_i \in A$, while local ones may also return *ranked subsets* of A. Note that complete trust graph information is only important for *global* group trust metrics, but not for *local* ones. Informally, local group trust metrics may be defined as metrics to compute *neighborhoods* of trusted peers for an individual a_i. As input parameters, these trust metrics take an individual $a_i \in A$ for which to compute the set of peers he should trust, as well as an amount of trust the latter wants to share among his most trustworthy agents. For instance, in [Levien and Aiken, 2000], the amount of trust is said to correspond to the number of agents that a_i wants to trust. The output is hence given by a *trusted subset* of A.

Note that scalar trust metrics are inherently local, while group trust metrics do not impose any restrictions on features for other axes.

5.2.2 Trust and Decentralization

Section 1.1.2 has mentioned the Semantic Web as sample scenario for our decentralized recommender framework. Hence, for the conception of our trust metric, we will also assume the Semantic Web as working environment and representative for large-scale decentralized infrastructures. Note that all considerations presented are also of utter relevance for large, decentralized networks other than the Semantic Web, e.g., very large peer-to-peer networks, the Grid, etc.

Before discussing specific requirements and fitness properties of trust metrics along those axes proposed before, we need to define one common trust *model* on which to rely upon. Some steps towards one such standardized model have already been taken and incorporated into the FOAF [Dumbill, 2002] project. FOAF is an abbreviation for "Friend of a Friend" and aims at enriching personal homepages with machine-readable content encoded in RDF statements. Besides various other information,

these publicly accessible pages allow their owners to nominate all individuals part of the FOAF universe they know, thus weaving a "web of acquaintances" [Golbeck et al., 2003]. Golbeck et al. [2003] have extended the FOAF schema to also contain *trust* assertions with values ranging from 1 to 9, where 1 denotes complete distrust and 9 absolute trust towards the individual for whom the assertion has been issued. Their assumption that trust and distrust represent *symmetrically opposed* concepts is in line with Abdul-Rahman and Hailes [2000].

The model that we adopt is quite similar to FOAF and its extensions, but only captures the notion of trust and lack of trust, instead of trust and distrust. Note that zero trust and distrust are *not* the same [Marsh, 1994b] and may hence not be intermingled. Explicit modelling of distrust has some serious implications for trust metrics and will hence be discussed separately in Section 5.4. Mind that only few research endeavors have investigated the implementation of distrust so far, e.g., Jøsang et al. [2003], Guha [2003], and Guha et al. [2004].

5.2.2.1 Trust Model

As is the case for FOAF, we assume that all trust information is publicly accessible for any agent in the system through machine-readable personal homepages distributed over the network. Agents $a_i \in A = \{a_1, a_2, \ldots, a_n\}$ are associated with a partial trust function $W_i \in T = \{W_1, W_2, \ldots, W_n\}$ each, where $W_i : A \rightarrow [0,1]^{\perp}$ holds, which corresponds to the set of trust assertions that a_i has stated.

In most cases, functions $W_i(a_j)$ will be very sparse as the number of individuals an agent is able to assign explicit trust ratings for is much smaller than the total number n of agents. Moreover, the higher the value of $W_i(a_j)$, the more trustworthy a_i deems a_j. Conversely, $W_i(a_j) = 0$ means that a_i considers a_j to be *not trustworthy*. The assignment of trust through continuous values between 0 and 1, and their adopted semantics, is in perfect accordance with [Marsh, 1994a], where possible stratifications of trust values are proposed. Our trust model defines one directed trust graph with nodes being represented by agents $a_i \in A$, and directed edges from nodes a_i to nodes a_j representing trust statements $W_i(a_j)$.

For convenience, we introduce the partial function $W : A \times A \rightarrow [0,1]^{\perp}$, which we define as the union of all partial functions $W_i \in T$.

5.2.2.2 Trust Metrics for Decentralized Networks

Trust and reputation ranking metrics have primarily been used for the PKI [Reiter and Stubblebine, 1997a,b; Levien and Aiken, 1998; Maurer, 1996; Beth et al., 1994], rating and reputation systems part of online communities [Guha, 2003; Levien and Aiken, 2000; Levien, 2004], peer-to-peer networks [Kamvar et al., 2003; Sankaralingam et al., 2003; Kinateder and Rothermel, 2003; Kinateder and Pearson, 2003; Aberer and Despotovic, 2001], and also mobile computing [Eschenauer et al., 2002].

Each of these scenarios favors different trust metrics. For instance, reputation systems for online communities tend to make use of *centralized trust servers* that compute global trust values for all users on the system [Guha, 2003]. On the other hand, peer-to-peer networks of moderate size rely upon distributed approaches that are in most cases based upon PageRank [Kamvar et al., 2003; Sankaralingam et al., 2003].

The Semantic Web, however, as an example for a large-scale decentralized environment, is expected to be made up of millions of nodes, i.e., millions of agents. The fitness of *distributed* approaches to trust metric computation, such as depicted by Richardson et al. [2003] and Kamvar et al. [2003], hence becomes limited for various reasons:

Trust data storage. Every agent a_i needs to store trust rating information about any other agent a_j on the Semantic Web. Agent a_i uses this information in order to merge it with own trust beliefs and propagates the synthesized information to his trusted agents [Levien, 2004]. Even though one might expect the size of the Semantic Web to be several orders of magnitude smaller than the traditional Web, the number of agents for whom to keep trust information will still exceed the storage capacities of most nodes.

Convergence. The structure of the Semantic Web is diffuse and not subject to some higher ordering principle or hierarchy. Furthermore, the process of trust propagation is *necessarily asynchronous* (see Section 1.2). As the Semantic Web is huge in size with possibly numerous antagonist or idle agents, convergence of trust values might take a very long time.

The huge advantage of distributed approaches, on the other hand, is the *immediate availability* of computed trust information about any other agent a_j in the system. Moreover, agents have to disclose their trust assertions only to peers they actually *trust* [Richardson et al., 2003]. For instance, suppose that a_i declares his trust in a_j by $W_i(a_j) = 0.1$, which is very low. Hence, a_i might want a_j not to know about that fact. As distributed metrics only propagate *synthesized* trust values from nodes to successor nodes in the trust graph, a_i would not have to openly disclose his trust statements to a_j.

As it comes to centralized, i.e., *locally computed*, metrics, *full* trust information access is required for agents inferring trust. Hence, online communities based on trust require their users to disclose all trust information to the community server, but not necessarily to other peers [Guha, 2003]. Privacy thus remains preserved. On the Semantic Web and in the area of ubiquitous and mobile computing, however, there is no such central authority that computes trust. *Any* agent might want to do so. Our own trust model, as well as trust models proposed by Golbeck et al. [2003], Eschenauer et al. [2002], and Abdul-Rahman and Hailes [1997], are hence based upon the assumption of *publicly available trust information*. Though privacy concerns may persist, this assumption is vital, owing to the afore-mentioned deficiencies of distributed computation models. Moreover, centralized *global* metrics, such

as depicted by Guha [2003] and Page et al. [1998], also fail to fit our requirements: because of the huge number of agents issuing trust statements, only dedicated server clusters could be able to manage the whole bulk of trust relationships. For small agents and applications roaming the Semantic Web, global trust computation is not feasible.

Scalar metrics, e.g., PKI proposals [Reiter and Stubblebine, 1997a,b; Levien and Aiken, 1998; Maurer, 1996; Beth et al., 1994] and those metrics described by Golbeck et al. [2003], have poor scalability properties, owing to exponential time complexity [Reiter and Stubblebine, 1997a].

Consequently, we advocate *local group* trust metrics for the Semantic Web and other large-scale decentralized networks. These metrics bear several welcome properties with respect to computability and complexity, which may be summarized as follows:

Partial trust graph exploration. Global metrics require a priori full knowledge of the entire trust network. Distributed metrics store trust values for all agents in the system, thus implying massive data storage demands. On the other hand, when computing trusted *neighborhoods*, the trust network only needs to be explored partially: originating from the trust source, one only follows those trust edges that seem promising, i.e., bearing high trust weights, and which are not too far away from the trust source. Inspection of personal, machine-readable homepages is thus performed in a just-in-time fashion. Hence, prefetching bulk trust information is not required.

Computational scalability. Tightly intertwined with partial trust graph exploration is computational complexity. Local group trust metrics scale well to any social network size, as only tiny subsets of relatively constant size[4] are visited. This is not the case for global trust metrics.

5.3 Local Group Trust Metrics

Local group trust metrics, in their function as means to compute trust neighborhoods, have not been subject to mainstream research so far. Significant research has effectively been limited to the work done by Levien [2004] who has conceived and deployed the Advogato group trust metric. This section provides an overview of Advogato and introduces our own Appleseed trust metric, eventually comparing both approaches.

[4]Supposing identical parameterizations for the metrics in use, as well as similar network structures.

5.3.1 Outline of Advogato Maxflow

The Advogato maximum flow trust metric has been proposed by Levien and Aiken [2000] in order to discover which users are trusted by members of an online community and which are not. Trust is computed through one centralized community server and considered relative to a seed of users enjoying supreme trust. However, the metric is not only applicable to community servers, but also to *arbitrary* agents which may compute *personalized lists* of trusted peers, not only one single global ranking for the whole community they belong to. In this case, the active agent himself constitutes the singleton trust seed. The following paragraphs briefly introduce Advogato's basic concepts. For more detailed information, refer to [Levien and Aiken, 2000], [Levien and Aiken, 1998], and [Levien, 2004].

5.3.1.1 Trust Computation Steps

Local group trust metrics compute sets of agents trusted by those being part of the trust seed. In case of Advogato, its input is given by an integer number n, which is supposed to be equal to the number of members to trust [Levien and Aiken, 2000], as well as the trust seed s, which is a subset of the entire set of users A. The output is a *characteristic function* that maps each member to a boolean value indicating his trustworthiness:

$$\text{Trust}_M : 2^A \times \mathbb{N}_0^+ \rightarrow (A \rightarrow \{\text{true}, \text{false}\}) \tag{5.1}$$

The trust model underlying Advogato does *not* provide support for weighted trust relationships in its original version.[5] Hence, trust edges extending from individual x to y express *blind*, i.e., *full*, trust of x in y. The metrics for PKI maintenance suppose similar models. Maximum integer network flow computation [Ford and Fulkerson, 1962] has been investigated by Reiter and Stubblebine [1997b,a] in order to make trust metrics more reliable. Levien adopted and extended this approach for group trust in his Advogato metric:

Capacities $C_A : A \rightarrow \mathbb{N}$ are assigned to every community member $x \in A$ based upon the shortest-path distance from the seed to x. Hereby, the capacity of the seed itself is given by the input parameter n mentioned before, whereas the capacity of each successive distance level is equal to the capacity of the previous level l divided by the average outdegree of trust edges $e \in E$ extending from l. The trust graph we obtain hence contains one single source, which is the set of seed nodes considered as one single "virtual" node, and multiple sinks, i.e., all nodes other than those defining the seed. Capacities $C_A(x)$ constrain nodes. In order to apply Ford-Fulkerson maximum integer network flow [Ford and Fulkerson, 1962], the underlying problem has to be formulated as single-source/single-sink, having capacities C_E :

[5] Though various levels of peer certification exist, their interpretation does not perfectly align with weighted trust relationships.

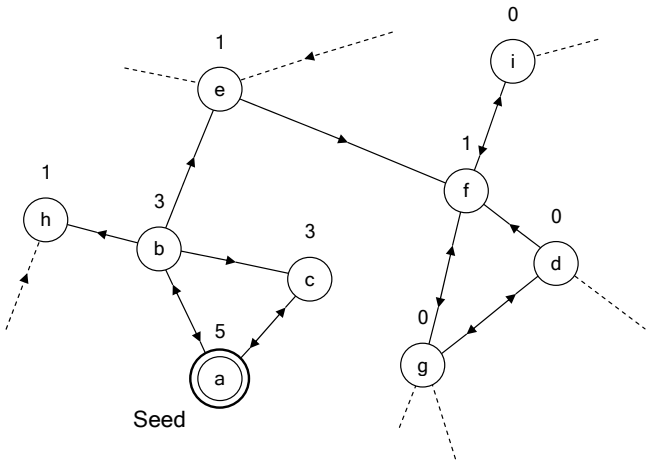

Figure 5.3. Trust graph *before* conversion for Advogato

$E \to \mathbb{N}$ constrain *edges* instead of *nodes*. Hence, Algorithm 5.1 is applied to the old directed graph $G = (A, E, C_A)$, resulting in a new graph structure $G' = (A', E', C_{E'})$.

Figure 5.4 depicts the outcome of converting node-constrained single-source/multiple-sink graphs (see Figure 5.3) into single-source/single-sink ones with capacities constraining edges.

Conversion is followed by simple integer maximum network flow computation from the trust seed to the super-sink. Eventually, the trusted agents x are exactly those peers for whom there is flow from "negative" nodes x^- to the super-sink. An additional constraint needs to be introduced, requiring flow from x^- to the super-sink whenever there is flow from x^- to x^+. The latter constraint assures that node x does not only serve as an intermediate for the flow to pass through, but is *actually added* to the list of trusted agents when reached by network flow. However, the standard implementation of Ford-Fulkerson traces shortest paths to the sink first [Ford and Fulkerson, 1962]. The above constraint is thus satisfied implicitly already.

Example 4 (Advogato trust computation) Suppose the trust graph depicted in Figure 5.3. The only seed node is a with initial capacity $C_A(a) = 5$. Hence, taking into account the outdegree of a, nodes at unit distance from the seed, i.e., nodes b and c, are assigned capacities $C_A(b) = 3$ and $C_A(c) = 3$, respectively. The average outdegree of both nodes is 2.5 so that second level nodes e and h obtain unit capacity. When computing maximum integer network flow, agent a will accept himself, b, c, e, and h as trustworthy peers.

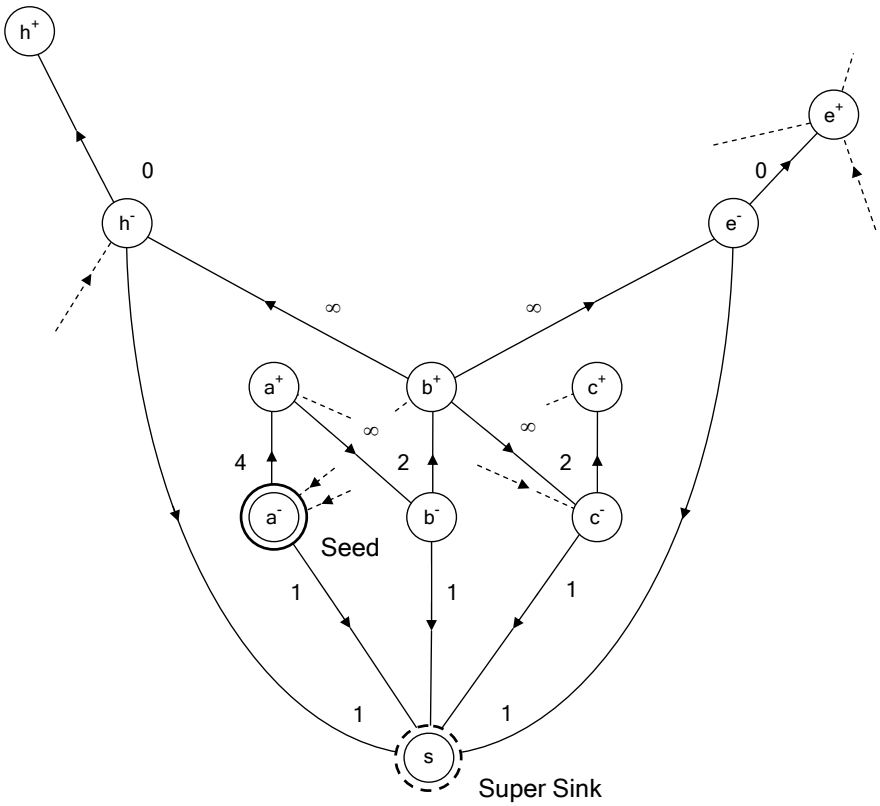

Figure 5.4. Trust graph *after* conversion for Advogato

5.3.1.2 Attack-Resistance Properties

Advogato has been designed with resistance against massive attacks from malicious agents outside of the community in mind. Therefore, an upper bound for the number of "bad" peers chosen by the metric is provided in [Levien and Aiken, 2000], along with an informal security proof to underpin its fitness. Resistance against malevolent users trying to break into the community can already be observed in the example depicted by Figure 5.1, supposing node n to be "bad": though agent n is trusted by numerous persons, he is deemed less trustworthy than, for instance, x. While there are fewer agents trusting x, these agents enjoy higher trust reputation[6] than the numerous persons trusting n. Hence, it is not just the *number* of agents trusting an individual i, but also the trust *reputation* of these agents that exerts an impact on the trust assigned to i. PageRank [Page et al., 1998] works in a similar fashion and has been claimed to possess properties of attack-resistance similar to those of

[6]With respect to seed node a.

```
function transform (G = (A, E, C_A)) {
  set E' ← ∅, A' ← ∅;
  for all x ∈ A do
    add node x⁺ to A';
    add node x⁻ to A';
    if C_A(x) ≥ 1 then
      add edge (x⁻, x⁺) to E';
      set C_E'(x⁻, x⁺) ← C_A(x) − 1;
      for all (x, y) ∈ E do
        add edge (x⁺, y⁻) to E';
        set C_E'(x⁺, y⁻) ← ∞;
      end do
      add edge (x⁻, supersink) to E';
      set C_E'(x⁻, supersink) ← 1;
    end if
  end do
  return G' = (A', E', C_E');
}
```

Algorithm 5.1. Trust graph conversion

the Advogato trust metric [Levien, 2004]. In order to make the concept of attack-resistance more tangible, Levien proposes the "bottleneck property" as a common feature of attack-resistant trust metrics. Informally, this property states that the "trust quantity accorded to an edge $s \rightarrow t$ is not significantly affected by changes to the successors of t" [Levien, 2004].

Attack-resistance features of various trust metrics are discussed in detail in [Levien and Aiken, 1998] and [Twigg and Dimmock, 2003].

5.3.2 The Appleseed Trust Metric

The Appleseed trust metric constitutes the main contribution of this chapter and is our novel proposal for local group trust metrics. In contrast to Advogato, being inspired by maximum network flow computation, the basic intuition of Appleseed is motivated by *spreading activation models*. Spreading activation models have first been proposed by Quillian [1968] in order to simulate human comprehension through semantic memory, and are commonly described as "models of retrieval from long-term memory in which activation subdivides among paths emanating from an activated mental representation" [Smith et al., 2003]. By the time of this writing, the

```
procedure energize (e ∈ ℝ₀⁺, s ∈ A) {
  energy(s) ← energy(s) + e;
  e' ← e / Σ_{(s,n)∈E} W(s,n);
  if e > T then
    ∀(s,n) ∈ E : energize (e' · W(s,n), n);
  end if
}
```

Algorithm 5.2. Recursive energy propagation

seminal work of Quillian has been ported to a whole plethora of other disciplines, such as latent semantic indexing [Ceglowski et al., 2003] and text illustration [Hartmann and Strothotte, 2002]. As an example, we will briefly introduce the spreading activation approach adopted by Ceglowski et al. [2003], used for semantic search in contextual network graphs, in order to then relate Appleseed to that work.

5.3.2.1 Searches in Contextual Network Graphs

The graph model underlying contextual network search graphs is almost identical in structure to the one presented in Section 5.2.2.1, i.e., edges $(x, y) \in E \subseteq A \times A$ connecting nodes $x, y \in A$. Edges are assigned continuous weights through $W : E \rightarrow [0, 1]$. Source node s, the node from which we start searching, is activated through an injection of energy e, which is then propagated to other nodes along edges according to some set of simple rules: all energy is *fully divided* among successor nodes with respect to their normalized local edge weight, i.e., the higher the weight of an edge $(x, y) \in E$, the higher the portion of energy that flows along that edge. Furthermore, supposing average outdegrees greater than one, the closer node x to the injection source s, and the more paths lead from s to x, the higher the amount of energy flowing into x. To eliminate endless, marginal and negligible flow, energy streaming into node x must exceed threshold T in order not to run dry. The described approach is captured formally by Algorithm 5.2, which propagates energy recursively.

5.3.2.2 Trust Propagation

Algorithm 5.2 shows the basic intuition behind spreading activation models. In order to tailor these models to trust computation, later to become the Appleseed trust metric, serious adaptations are necessary. For instance, procedure energize(e, s) registers *all* energy e that has passed through node x, stored in energy(x). Hence, energy(x) represents the *relevance rank* of x. Higher values indicate higher node rank. However, at the same time, all energy contributing to the rank of x is passed

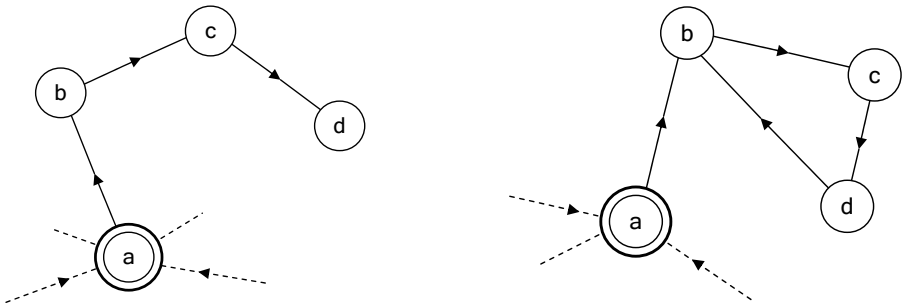

Figure 5.5. Node chains and rank sinks

without loss to successor nodes. Interpreting energy ranks as trust ranks thus implies numerous issues of semantic consistency as well as computability. Consider the graph depicted on the left-hand side of Figure 5.5. Applying spreading activation according to Ceglowski et al. [2003], trust ranks of nodes b and d will be identical. However, intuitively, d should be accorded *less* trust than b, since d's shortest-path distance to the trust seed is higher. Trust decay is commonly agreed upon [Guha, 2003; Jøsang et al., 2003], for people tend to trust individuals trusted by immediate friends more than individuals trusted only by friends of friends. The right-hand side of Figure 5.5 depicts even more serious issues: all energy, or trust[7], respectively, distributed along edge (a, b) becomes *trapped in a cycle* and will never be accorded to any other nodes but those being part of that cycle, i.e., b, c, and d. These nodes will eventually acquire infinite trust rank. Obviously, the *bottleneck property* [Levien, 2004] does not hold. Similar issues occur with simplified versions of PageRank [Page et al., 1998], where cycles accumulating infinite rank have been dubbed "rank sinks".

5.3.2.3 Spreading Factor

We handle both issues, i.e., trust decay in node chains and elimination of rank sinks, by tailoring the algorithm to rely upon our global *spreading factor* d. Hereby, let $\text{in}(x)$ denote the energy influx into node x. Parameter d then denotes the portion of energy $d{\cdot}\text{in}(x)$ that node x distributes among successors, while retaining $(1-d){\cdot}\text{in}(x)$. For instance, suppose $d = 0.85$ and energy quantity $\text{in}(x) = 5.0$ flowing into node x. Then, the total energy distributed to successor nodes amounts to 4.25, while the energy rank $\text{energy}(x)$ of x increases by 0.75. Special treatment is necessary for nodes with zero outdegree. For simplicity, we assume all nodes to have an outdegree of at least one, which makes perfect sense, as will be shown later.

[7]The terms "energy" and "trust" are used interchangeably in this context.

The spreading factor concept is very intuitive and, in fact, very close to real models of energy spreading through networks. Observe that the overall amount of energy in the network, after initial activation in^0, does not change over time. More formally, suppose that $\text{energy}(x) = 0$ for all $x \in A$ before injection in^0 into source s. Then the following equation holds in every computation step of our modified spreading algorithm, incorporating the concept of spreading factor d:

$$\sum_{x \in A} \text{energy}(x) = \text{in}^0 \tag{5.2}$$

Spreading factor d may also be seen as the *ratio* between *direct* trust in x and trust in the ability of x to *recommend* others as trustworthy peers. For instance, Beth et al. [1994] and Maurer [1996] explicitly differentiate between *direct* trust edges and *recommendation* edges.

We commonly assume $d = 0.85$, though other values may also seem reasonable. For instance, having $d \leq 0.5$ allows agents to keep most of the trust they are granted for themselves and only pass small portions of trust to their peers. Observe that low values for d favor trust proximity to the source of trust injection, while high values allow trust to also reach more distant nodes. Furthermore, the introduction of spreading factor d is crucial for making Appleseed retain Levien's bottleneck property, as will be shown in later sections.

5.3.2.4 Rank Normalization

Algorithm 5.2 makes use of edge weight normalization, i.e., the quantity $e_{x \to y}$ of energy distributed along (x, y) from x to successor node y depends on the *relative* weight of $x \to y$, i.e., $W(x, y)$ compared to the sum of weights of all outgoing edges of x:

$$e_{x \to y} = d \cdot \text{in}(x) \cdot \frac{W(x, y)}{\sum_{(x,s) \in E} W(x, s)} \tag{5.3}$$

Normalization is common practice in many trust metrics, among those PageRank [Page et al., 1998], EigenTrust [Kamvar et al., 2003], and AORank [Guha, 2003]. However, while normalized reputation or trust seem reasonable for models with plain, non-weighted edges, serious interferences occur when edges are *weighted*, as is the case for our trust model adopted in Section 5.2.2.1.

For instance, refer to the left-hand side of Figure 5.6 for unwanted effects: the amounts of energy that node a accords to successors b and d, i.e., $e_{a \to b}$ and $e_{a \to d}$, respectively, are identical in value. Note that b has issued only *one* trust statement $W(b, c) = 0.25$, stating that b's trust in c is rather weak. On the other hand, d assigns *full* trust to individuals e, f, and g. Nevertheless, the overall trust rank for d will be much higher than for any successor of d, for c is accorded $e_{a \to b} \cdot d$, while

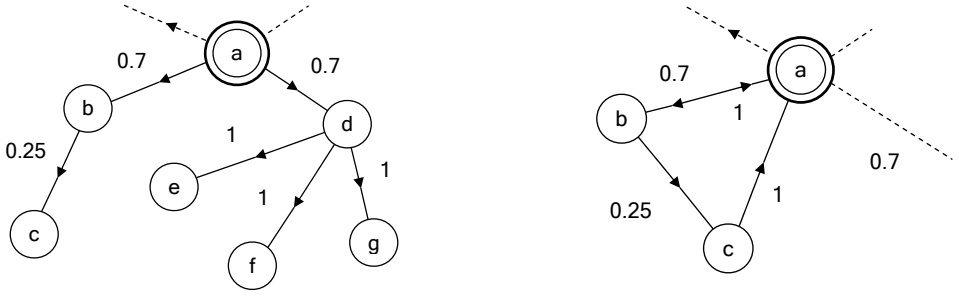

Figure 5.6. Issues with trust normalization

e, f, and g only obtain $e_{a\to d} \cdot d \cdot 1/3$ each. Hence, c will be trusted *three times* as much as e, f, and g, which is not reasonable at all.

5.3.2.5 Backward Trust Propagation

The above issue has already been discussed by Kamvar et al. [2003], but no solution has been proposed therein, arguing that "substantially good results" have been achieved despite the drawbacks. We propose to alleviate the problem by making use of *backward propagation* of trust to the source: when metric computation takes place, additional "virtual" edges (x, s) from every node $x \in A \setminus \{s\}$ to the trust source s are created. These edges are assigned full trust $W(x, s) = 1$. Existing backward links (x, s), along with their weights, are "overwritten". Intuitively, every node is supposed to *blindly trust the trust source* s, see Figure 5.6. The impacts of adding backward propagation links are threefold:

Mitigating relative trust. Again, we refer to the left-hand graph in Figure 5.6. Trust distribution in the underlying case becomes much fairer through backward propagation links, for c now only obtains $e_{a\to b} \cdot d \cdot (0.25/(1+0.25))$ from source s, while e, f, and g are accorded $e_{a\to d} \cdot d \cdot (1/4)$ each. Hence, trust ranks of both e, f, and g amount to 1.25 times the trust assigned to c.

Avoidance of dead ends. Dead ends, i.e., nodes x with zero outdegree, require special treatment in our computation scheme. Two distinct approaches may be adopted. First, the portion of incoming trust $d \cdot \mathrm{in}(x)$ supposed to be passed to successor nodes is completely discarded, which contradicts our intuition of no energy leaving the system. Second, instead of retaining $(1 - d) \cdot \mathrm{in}(x)$ of incoming trust, x keeps *all* trust. The latter approach is also not sensible as it encourages users to not issue trust statements for their peers. Luckily, with backward propagation of trust, all nodes are *implicitly linked* to the trust

source s, so that there are no more dead ends to consider.

Favoring trust proximity. Backward links to the trust source s are favorable for nodes close to the source, as their eventual trust rank will increase. On the other hand, nodes further away from s are penalized.

5.3.2.6 Nonlinear Trust Normalization

In addition to backward propagation, we propose supplementary measures to decrease the negative impact of trust spreading based on relative weights. Situations where nodes y with poor ratings from x are awarded high overall trust ranks, thanks to the low outdegree of x, have to be avoided. Taking the squares of local trust weights provides an appropriate solution:

$$e_{x \to y} = d \cdot \text{in}(x) \cdot \frac{W(x,y)^2}{\sum_{(x,s) \in E} W(x,s)^2} \tag{5.4}$$

As an example, refer to node b in Figure 5.6. With squared normalization, the total amount of energy flowing backward to source a increases, while the amount of energy flowing to the poorly trusted node c decreases significantly. Accorded trust quantities $e_{b \to a}$ and $e_{b \to c}$ amount to $d \cdot \text{in}(b) \cdot (1/1.0625)$ and $d \cdot \text{in}(b) \cdot (0.0625/1.0625)$, respectively. A more severe penalization of poor trust ratings can be achieved by selecting powers above two.

5.3.2.7 Algorithm Outline

Having identified modifications to apply to spreading activation models in order to tailor them for local group trust metrics, we are now able to formulate the core algorithm of Appleseed. Input and output are characterized as follows:

$$\text{Trust}_\alpha : A \times \mathbb{R}_0^+ \times [0,1] \times \mathbb{R}^+ \to (\text{trust} : A \to \mathbb{R}_0^+) \tag{5.5}$$

The first input parameter specifies trust seed s, the second trust injection e, parameter three identifies spreading factor $d \in [0,1]$, and the fourth argument binds accuracy threshold T_c, which serves as convergence criterion. Similar to Advogato, the output is an assignment function of trust with domain A. However, Appleseed allows *rankings* of agents with respect to trust accorded. Advogato, on the other hand, only assigns boolean values indicating presence or absence of trust.

Appleseed works with *partial* trust graph information. Nodes are accessed only when needed, i.e., when reached by energy flow. Trust ranks $\text{trust}(x)$, which correspond to $\text{energy}(x)$ in Algorithm 5.2, are initialized to 0. Any unknown node u hence obtains $\text{trust}(u) = 0$. Likewise, virtual trust edges for backward propagation from node x to the source are added *at the moment that x is discovered*. In every iteration, for those nodes x reached by flow, the amount of incoming trust is computed

as follows:

$$\text{in}(x) = d \cdot \sum_{(p,x) \in E} \Big(\text{in}(p) \cdot \frac{W(p,x)}{\sum_{(p,s) \in E} W(p,s)} \Big) \tag{5.6}$$

Incoming flow for x is hence determined by all flow that predecessors p distribute along edges (p, x). Note that the above equation makes use of *linear normalization* of relative trust weights. The replacement of linear by nonlinear normalization according to Section 5.3.2.6 is straight-forward, though. The trust rank of x is updated as follows:

$$\text{trust}(x) \leftarrow \text{trust}(x) + (1 - d) \cdot \text{in}(x) \tag{5.7}$$

Trust networks generally contain cycles and thus allow no topological sorting of nodes. Hence, the computation of $\text{in}(x)$ for reachable $x \in A$ becomes *inherently recursive*. Several iterations for all nodes are required in order to make the computed information converge towards the least fixpoint. The following criterion has to be satisfied for convergence, relying upon accuracy threshold T_c briefly introduced before.

Definition 1 (Termination) Suppose that $A_i \subseteq A$ represents the set of nodes that were discovered until step i, and $\text{trust}_i(x)$ the current trust ranks for all $x \in A$. Then the algorithm terminates when the following condition is satisfied after step i:

$$\forall x \in A_i : \text{trust}_i(x) - \text{trust}_{i-1}(x) \leq T_c \tag{5.8}$$

Informally, Appleseed terminates when changes of trust ranks with respect to the preceding iteration $i - 1$ are not greater than accuracy threshold T_c.

Moreover, when supposing spreading factor $d > 0$, accuracy threshold $T_c > 0$, and trust source s part of some connected component $G' \subseteq G$ containing at least two nodes, convergence, and thus termination, is guaranteed. The following paragraph gives an informal proof:

Proof 1 (Convergence of Appleseed) Assume that f_i denotes step i's quantity of energy flowing through the network, i.e., all the trust that has not been captured by some node x through function $\text{trust}_i(x)$. From Equation 5.2 follows that in^0 constitutes the *upper boundary* of trust energy floating through the network, and f_i can be computed as follows:

$$f_i = \text{in}^0 - \sum_{x \in A} \text{trust}_i(x) \tag{5.9}$$

Since $d > 0$ and $\exists (s, x) \in E, x \neq s$, the sum of the current trust ranks $\text{trust}_i(x)$ of all $x \in A$ is *strictly increasing* for increasing i. Consequently, $\lim_{i \to \infty} f_i = 0$ holds.

```
function Trust_α (s ∈ A, in^0 ∈ ℝ_0^+, d ∈ [0,1], T_c ∈ ℝ^+) {
  set in_0(s) ← in^0, trust_0(s) ← 0, i ← 0;
  set A_0 ← {s};
  repeat
    set i ← i + 1;
    set A_i ← A_{i−1};
    ∀x ∈ A_{i−1} : set in_i(x) ← 0;
    for all x ∈ A_{i−1} do
      set trust_i(x) ← trust_{i−1}(x) + (1 − d) · in_{i−1}(x);
      for all (x, u) ∈ E do
        if u ∉ A_i then
          set A_i ← A_i ∪ {u};
          set trust_i(u) ← 0, in_i(u) ← 0;
          add edge (u, s), set W(u, s) ← 1;
        end if
        set w ← W(x, u) / Σ_{(x,u') ∈ E} W(x, u');
        set in_i(u) ← in_i(u) + d · in_{i−1}(x) · w;
      end do
    end do
    set m = max_{y ∈ A_i}{trust_i(y) − trust_{i−1}(y)};
  until (m ≤ T_c)
  return (trust : {(x, trust_i(x)) | x ∈ A_i});
}
```

Algorithm 5.3. Outline of the Appleseed trust metric

Moreover, since termination is defined by some fixed accuracy threshold $T_c > 0$, there exists some step k such that $\lim_{i \to k} f_i \leq T_c$. □

5.3.2.8 Parameterization and Experiments

Appleseed allows numerous parameterizations of input variables, some of which are subject to discussion in the section at hand. Moreover, we provide experimental results exposing the observed effects of parameter tuning. Note that all experiments have been conducted on data obtained from "real" social networks: we have written several Web crawling tools to mine the Advogato community Web site and extract trust assertions stated by its more than 8,000 members. Hereafter, we converted all trust data to our trust model proposed in Section 5.2.2.1. The Advogato community server supports four different levels of peer certification, namely Observer, Ap-

PRENTICE, JOURNEYER, and MASTER. We mapped these *qualitative* certification levels to quantitative ones, assigning $W(x, y) = 0.25$ for x certifying y as OBSERVER, $W(x, y) = 0.5$ for an APPRENTICE, and so forth. The Advogato community undergoes rapid growth and our crawler extracted $3,224,101$ trust assertions. Preprocessing and data cleansing were thus inevitable, eliminating reflexive trust statements $W(x, x)$ and shrinking trust certificates to reasonable sizes. Note that some eager Advogato members have issued *more than two thousand* trust statements, yielding an overall average outdegree of 397.69 assertions per node.Clearly, this figure is beyond dispute. Hence, applying our set of extraction tools, we tailored the test data obtained from Advogato to our needs and extracted trust networks with specific average outdegrees for the experimental analysis.

Trust Injection

Trust values $\text{trust}(x)$ computed by the Appleseed metric for source s and node x may differ greatly from explicitly assigned trust weights $W(s, x)$. We already mentioned before that computed trust ranks may *not* be interpreted as absolute values, but rather in comparison with ranks assigned to all other peers. In order to make assigned rank values more tangible, though, one might expect that tuning the trust injection in^0 to satisfy the following proposition will align computed ranks and explicit trust statements:

$$\forall (s, x) \in E : \ \text{trust}(x) \in [W(s, x) - \epsilon, W(s, x) + \epsilon] \tag{5.10}$$

However, when assuming reasonably small ϵ, the approach does not succeed. Recall that *computed* trust values of successor nodes x of s do not only depend on assertions made by s, but also on trust ratings asserted by other peers. Hence, a perfect alignment of explicit trust ratings with computed ones cannot be accomplished. However, we propose a heuristic alignment method, incorporated into Algorithm 5.4, which has proven to work remarkably well in diverse test scenarios. The basic idea is to add another node i and edge (s, i) with $W(s, i) = 1$ to the trust graph $G = (A, E, W)$, treating (s, i) as an indicator to test whether trust injection in^0 is "good" or not. Consequently, parameter in^0 has to be adapted in order to make $\text{trust}(i)$ converge towards $W(s, i)$. The trust metric computation is hence repeated with different values for in^0 until convergence of the explicit and the computed trust value of i is achieved. Eventually, edge (s, i) and node i are removed and the computation is performed one more time. Experiments have shown that our imperfect alignment method yields computed ranks $\text{trust}(x)$ for direct successors x of trust source s which come close to previously specified trust statements $W(s, x)$.

```
function Trust_heu (s ∈ A, d ∈ [0,1], T_c ∈ ℝ⁺) {
  add node i, edge (s,i), set W(s,i) ← 1;
  set in⁰ ← 20, ε ← 0.1;
  repeat
    set trust ← Trust_α (s, in⁰, d, T_c);
    in⁰ ← adapt (W(s,i), trust(i), in⁰);
  until trust(i) ∈ [W(s,i) − ε, W(s,i) + ε]
  remove node i, remove edge (s, i);
  return Trust_α (s, in⁰, d, T_c);
}
```

Algorithm 5.4. Heuristic weight alignment method

Spreading Factor

Small values for d tend to overly reward nodes close to the trust source and penalize remote ones. Recall that *low* d allows nodes to retain most of the incoming trust quantity for themselves, while *large* d stresses the recommendation of trusted individuals and makes nodes distribute most of the assigned trust to their successor nodes.

Experiment 1 (Spreading factor impact) We compare distributions of computed rank values for three diverse instantiations of d, namely $d_1 = 0.1$, $d_2 = 0.5$, and $d_3 = 0.85$. Our setup is based upon a social network with an average outdegree of 6 trust assignments, and features 384 nodes reached by trust energy spreading from our designated trust source. We furthermore suppose $\text{in}^0 = 200$, $T_c = 0.01$, and *linear* weight normalization. Computed ranks are classified into 11 histogram cells with nonlinear cell width. Obtained output results are displayed in Figure 5.7. Mind that we have chosen *logarithmic* scales for the vertical axis in order to render the diagram more legible. For d_1, we observe that the largest number of nodes x with ranks $\text{trust}(x) \geq 25$ is generated. On the other hand, virtually no ranks ranging from 0.2 to 1 are assigned, while the number of nodes with ranks smaller than 0.05 is again much higher for d_1 than for both d_2 and d_3. Instantiation $d_3 = 0.85$ exhibits behavior opposed to that of d_1. No ranks with $\text{trust}(x) \geq 25$ are accorded, while interim ranks between 0.1 and 10 are much more likely for d_3 than for both other instantiations of spreading factor d. Consequently, the number of ranks below 0.05 is lowest for d_3.

The experiment demonstrates that high values for parameter d tend to distribute trust more evenly, neither overly rewarding nodes close to the source, nor penalizing

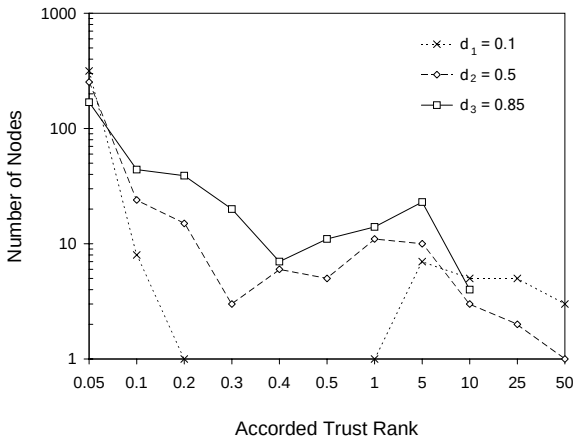

Figure 5.7. Spreading factor impact

remote ones too rigidly. On the other hand, low d assigns high trust ranks to very few nodes, namely those which are closest to the source, while the majority of nodes obtains very low trust rank. We propose to set $d = 0.85$ for general use.

Convergence

We already mentioned before that the Appleseed algorithm is *inherently recursive*. Parameter T_c represents the ultimate criterion for termination. We demonstrate through an experiment that convergence is reached very fast, no matter how large the number of nodes trust is flowing through, and no matter how large the initial trust injection.

Experiment 2 (Convergence rate) The trust network we consider has an average outdegree of 5 trust statements per node. The number of nodes for which trust ranks are assigned amounts to 572. We suppose $d = 0.85$, $T_c = 0.01$, and *linear* weight normalization. Two separate runs were computed, one with trust activation $\text{in}_1 = 200$, the other with initial energy $\text{in}_2 = 800$. Figure 5.8 demonstrates the rapid convergence of both runs. Though the trust injection for the second run is 4 times as high as for the first, convergence is reached in only few more iterations: run one takes 38 iterations, run two terminates after 45 steps.

For both runs, we assumed accuracy threshold $T_c = 0.01$, which is extremely small and accurate beyond necessity already. However, experience taught us that convergence takes place rapidly even for very large networks and high amounts of trust injected, so that assuming the latter value for T_c poses no scalability issues.

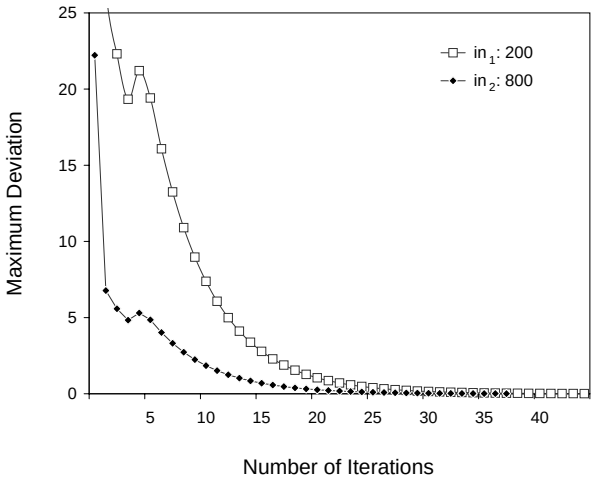

Figure 5.8. Convergence of Appleseed

In fact, the amount of nodes taken into account for trust rank assignment in the above example well exceeds practical usage scenarios: mind that the case at hand demands 572 files to be fetched from the Web, complaisantly supposing that these pages are cached after their first access. Hence, we claim that the actual bottleneck of group trust computation is *not* the Appleseed metric itself, but downloads of trust resources from the network. This bottleneck might also be the reason for selecting thresholds T_c greater than 0.01, in order to make the algorithm terminate after fewer node accesses.

5.3.2.9 Implementation and Extensions

We implemented Appleseed in Java, based upon Algorithm 5.3. We applied moderate fine-tuning and supplemented our metric with an architectural cushion in order to access "real" machine-readable RDF homepages. Other notable modifications to the core algorithm are discussed briefly:

Maximum number of unfolded nodes. We supplemented the set of input parameters by yet another argument M, which specifies the maximum number of nodes to unfold. This extension hinders trust energy from inordinately covering major parts of the entire network. Note that accessing the personal, machine-readable homepages, which contain trust information required for metric computation, represents the actual computation bottleneck. Hence, expanding as few nodes as possible is highly desirable. When choosing reasonably large M, for instance, three times the number of agents assumed trustworthy, we may expect to not

miss any relevant nodes: mind that Appleseed proceeds breadth-first and thus considers close nodes first, which are more eligible for trust than distant ones.

Upper-bounded trust path lengths. Another approach to sensibly restrict the number of nodes unfolded relies upon upper-bounded path lengths. The idea of constraining path lengths for trust computation has been adopted before by Reiter and Stubblebine [1997a] and within the X.509 protocol [Housely et al., 1999]. Depending on the overall trust network connectivity, we opt for maximum path lengths around three, aware of Milgram's "six degrees of separation" paradigm [Milgram, 1992]. In fact, trust decay is inherent to Appleseed, thanks to spreading factor d and backward propagation. Stripping nodes at large distances from the seed therefore only marginally affects the trust metric computation results while simultaneously providing major speed-ups.

Zero trust retention for the source. Third, we modified Appleseed to hinder trust source s from accumulating trust energy, essentially introducing one novel spreading factor $d_s = 1.0$ for the seed only. Consequently, all trust is divided among peers of s and none retained, which is reasonable. Convergence may be reached faster, since $\text{trust}_{i+1}(x) - \text{trust}_i(x)$ tends to be maximal for seed node s, thanks to backward propagation of trust (see Section 5.3.2.5). Furthermore, supposing the same trust quantity in^0 injected, assigned trust ranks become greater in value, also enlarging gaps between neighbors in trust rank.

Testbed Conception

Trust metrics and models for trust propagation have to be *intuitive*, i.e., humans must eventually comprehend *why* agent a_i has been accorded a higher trust rank than a_j and come to similar results when asked for personal judgement. Consequently, we implemented our own testbed, which graphically displays social networks. We made use of the yFiles [Wiese et al., 2001] library to perform complex graph drawing and layouting tasks[8]. The testbed allows for parameterizing Appleseed through dialogs. Detailed output is provided, both graphical and textual. Graphical results comprise the highlighting of nodes with trust ranks above certain thresholds, while textual results return quantitative trust ranks of all accessed nodes, the number of iterations, and so forth. We also implemented the Advogato trust metric and incorporated the latter into our testbed. Hereby, our implementation of Advogato does not require a priori complete trust graph information, but accesses nodes "just in time", similar to Appleseed. All experiments were conducted on top of the testbed application.

5.3.3 Comparison of Advogato and Appleseed

Advogato and Appleseed are both implementations of local group trust metrics. Advogato has already been successfully deployed into the Advogato online community,

[8] See Figure 7.2 for a sample visualization.

though quantitative evaluation results have not been provided yet. In order to evaluate the fitness of Appleseed as an appropriate means for group trust computation, we relate our approach to Advogato for qualitative comparison:

(F.1) **Attack-resistance.** This property defines the behavior of trust metrics in case of malicious nodes trying to invade into the system. For evaluation of attack-resistance capabilities, we have briefly introduced the "bottleneck property" in Section 5.3.1.2, which holds for Advogato. In order to recapitulate, suppose that s and t are nodes and connected through trust edge (s, t). Node s is assumed good, while t is an attacking agent trying to make good nodes trust malevolent ones. In case the bottleneck property holds, manipulation "on the part of bad nodes does not affect the trust value" [Levien, 2004]. Clearly, Appleseed satisfies the bottleneck property, for nodes cannot raise their impact by modifying the structure of trust statements they issue. Bear in mind that the amount of trust accorded to agent t *only* depends on his predecessors and does not increase when t adds more nodes. Both, spreading factor d and normalization of trust statements, ensure that Appleseed maintains attack-resistance properties according to Levien's definition.

(F.2) **Eager truster penalization.** We have indicated before that issuing multiple trust statements dilutes trust accorded to successors. According to Guha [2003], this does not comply with real world observations, where statements of trust "do not decrease in value when the user trusts one more person [...]". The malady that Appleseed suffers from is common to many trust metrics, most notably those based upon finding principal eigenvectors [Page et al., 1998; Kamvar et al., 2003; Richardson et al., 2003]. On the other hand, the approach pursued by Advogato does *not* penalize trust relationships asserted by eager trust dispensers, for node capacities do not depend on *local* information. Remember that capacities of nodes pertaining to level l are assigned based on the capacity of level $l - 1$, as well as the *overall* outdegree of nodes part of that level. Hence, Advogato *encourages* agents issuing numerous trust statements, while Appleseed *penalizes* overly abundant trust certificates.

(F.3) **Deterministic trust computation.** Appleseed is deterministic with respect to the assignment of trust rank to agents. Hence, for any arbitrary trust graph $G = (A, E, W)$ and for every node $x \in A$, linear equations allow for characterizing the amount of trust assigned to x, as well as the quantity that x accords to successor nodes. Advogato, however, is *non-deterministic*. Though the *number* of trusted agents, and therefore the computed maximum flow size, is determined for given input parameters, the set of agents is not. Changing the order in which trust assertions are issued may yield different results. For example, suppose $C_A(s) = 1$ holds for trust seed s. Furthermore, assume s has issued trust certificates for two agents, b and c. The actual choice between b or c as trustworthy peer with maximum flow *only depends on the order* in which

nodes are accessed.

(F.4) **Model and output type.** Basically, Advogato supports non-weighted trust statements only. Appleseed is more versatile by virtue of its trust model based on *weighted* trust certificates. In addition, Advogato returns one set of trusted peers, whereas Appleseed assigns *ranks* to agents. These ranks allow to select most trustworthy agents first and relate them to each other with respect to their accorded rank. Hereby, the definition of thresholds for trustworthiness is left to the user who can thus tailor relevant parameters to fit different application scenarios. For instance, raising the application-dependent threshold for the selection of trustworthy peers, which may be either an absolute or a relative value, allows for enlarging the neighborhood of trusted peers. Appleseed is hence more adaptive and flexible than Advogato.

The afore-mentioned characteristics of Advogato and Appleseed are briefly summarized in Table 5.1.

	Feature **F.1**	Feature **F.2**	Feature **F.3**	Feature **F.4**
Advogato	yes	no	no	boolean
Appleseed	yes	yes	yes	ranking

Table 5.1. Characteristics of Advogato and Appleseed

5.4 Distrust

The notion of distrust is one of the most controversial topics when defining trust metrics and trust propagation. Most approaches completely *ignore* distrust and only consider *full* trust or *degrees of trust* [Levien and Aiken, 1998; Mui et al., 2002; Beth et al., 1994; Maurer, 1996; Reiter and Stubblebine, 1997a; Richardson et al., 2003]. Others, among those Abdul-Rahman and Hailes [1997], Chen and Yeager [2003], Aberer and Despotovic [2001], and Golbeck et al. [2003], allow for distrust ratings, though, but do not consider the subtle semantic differences that exist between those two notions, i.e., trust and distrust. Consequently, according to Gans et al. [2001], "distrust is regarded as just the other side of the coin, that is, there is generally a symmetric scale with complete trust on one end and absolute distrust on the other." Furthermore, some researchers equate the notion of distrust with *lack of trust information*. However, in his seminal work on the essence of trust, Marsh [1994a]

has already pointed out that those two concepts, i.e., lack of trust and distrust, may *not* be intermingled. For instance, in absence of trustworthy agents, one might be more prone to accept recommendations from non-trusted persons, being non-trusted probably because of lack of prior experiences [Marsh, 1994b], than from persons we explicitly *distrust*, the distrust resulting from bad past experiences or deceit. However, even Marsh pays little attention to the specifics of distrust.

Gans et al. [2001] were among the first to recognize the importance of distrust, stressing the fact that "distrust is an irreducible phenomenon that cannot be offset against any other social mechanisms", including trust. In their work, an explicit distinction between confidence, trust, and distrust is made. Moreover, the authors indicate that distrust might be highly relevant to social networks. Its impact is not inherently negative, but may also influence the network in an extremely positive fashion. However, the primary focus of this work is on methodology issues and planning, not considering trust assertion evaluations and propagation through appropriate metrics.

Guha et al. [2004] acknowledge the immense role of distrust with respect to trust propagation applications, arguing that "distrust statements are very useful for users to debug their web of trust" [Guha, 2003]. For example, suppose that agent a_i blindly trusts a_j, which again blindly trusts a_k. However, a_i completely distrusts a_k. The distrust statement hence ensures that a_i will *not* accept beliefs and ratings from a_k, irrespective of him trusting a_j trusting a_k.

5.4.1 Semantics of Distrust

The non-symmetrical nature of distrust and trust, being two dichotomies, has already been recognized by recent sociological research [Lewicki et al., 1998]. In this section, we investigate the differences between distrust and trust with respect to inference opportunities and the propagation of beliefs.

5.4.1.1 Distrust as Negated Trust

Interpreting distrust as the negation of trust has been adopted by many trust metrics, among those trust metrics proposed by Abdul-Rahman and Hailes [1997, 2000], Jøsang et al. [2003], and Chen and Yeager [2003]. Basically, these metrics compute trust values by analyzing *chains* of trust statements from source s to target t, eventually merging them to obtain an aggregate value. Each chain hereby becomes synthesized into one single number through *weighted multiplication* of trust values along trust paths. Serious implications resulting from the assumption that trust concatenation relates to multiplication [Richardson et al., 2003], and distrust to negated trust, arise when agent a_i distrusts a_j, who distrusts a_k:[9]

[9] We oversimplify by using predicate calculus expressions, supposing that trust, and hence distrust, is fully transitive.

$$\neg\,\mathrm{trust}(a_i, a_j) \wedge \neg\,\mathrm{trust}(a_j, a_k) \models \mathrm{trust}(a_i, a_k) \tag{5.11}$$

Jøsang et al. [2003] are aware of this rather unwanted effect, but do not question its correctness, arguing that "the enemy of your enemy could well be your friend". Guha [2003], on the other hand, indicates that two distrust statements cancelling out each other commonly does *not* reflect desired behavior.

5.4.1.2 Propagation of Distrust

The *conditional transitivity* of trust [Abdul-Rahman and Hailes, 1997] is commonly agreed upon and represents the foundation and principal premiss that trust metrics rely upon. However, no consensus in literature has been achieved with respect to the *degree* of transitivity and the decay rate of trust. Many approaches therefore explicitly distinguish between *recommendation* trust and *direct* trust [Jøsang et al., 2003; Abdul-Rahman and Hailes, 1997; Maurer, 1996; Beth et al., 1994; Chen and Yeager, 2003] in order to keep apart the transitive fraction of trust from the non-transitive. Hence, in these works, only the *ultimate* edge within the trust chain, i.e., the one linking to the trust target, needs to be direct, while all others are supposed to be recommendations. For the Appleseed trust metric, this distinction is made through the introduction of spreading factor d. However, the conditional transitivity property of trust does not equally extend to distrust. The case of double negation through distrust propagation has already been considered. Now suppose, for instance, that a_i distrusts a_j, who trusts a_k. Supposing distrust to propagate through the network, we come to make the following inference:

$$\mathrm{distrust}(a_i, a_j) \wedge \mathrm{trust}(a_j, a_k) \models \mathrm{distrust}(a_i, a_k) \tag{5.12}$$

The above inference is more than questionable, for a_i penalizes a_k simply for being trusted by an agent a_j that a_i distrusts. Obviously, this assumption is not sound and does not reflect expected real-world behavior. We assume that distrust does not allow for making direct inferences *of any kind*. This conservative assumption well complies with [Guha, 2003].

5.4.2 Incorporating Distrust into Appleseed

We compare our distrust model with Guha's approach, making similar assumptions. Guha computes trust by means of *one global* group trust metric, similar to PageRank [Page et al., 1998]. For distrust, he proposes two candidate approaches. The first one directly integrates distrust into the iterative eigenvector computation and comes up with one single measure combining both trust and distrust. However, in networks dominated by distrust, the iteration might not converge [Guha, 2003]. The second proposal first computes trust ranks by trying to find the dominant eigenvector, and

then computes separate distrust ranks in one single step, based upon the iterative computation of trust ranks. Suppose that D_{a_i} is the set of agents who distrust a_i:

$$\text{DistrustRank}(a_i) = \frac{\sum_{a_j \in D_{a_i}} \text{TrustRank}(a_j)}{|D_{a_i}|} \tag{5.13}$$

The problem we perceive with this approach refers to *superimposing* the computation of distrust ranks *after* trust rank computation, which may yield some strange behavior: suppose an agent a_i who is highly controversial by engendering ambiguous sentiments, i.e., on the one hand, there are numerous agents that *trust* a_i, while on the other hand, there are numerous agents who *distrust* a_i. With the approach proposed by Guha, a_i's impact for distrusting other agents is huge, resulting from his immense positive trust rank. However, this should clearly not be the case, for a_i is subject to tremendous distrust himself, thus levelling out his high trust rank.

Hence, for our own approach, we intend to *directly* incorporate distrust into the iterative process of the Appleseed trust metric computation, and not superimpose distrust afterwards. Several pitfalls have to be avoided, such as the risk of non-convergence in case of networks dominated by distrust [Guha, 2003]. Furthermore, in absence of distrust statements, we want the distrust-enhanced Appleseed algorithm, which we denote by Trust_{α^-}, to yield results identical to those engendered by the original version Trust_{α}.

5.4.2.1 Normalization and Distrust

First, the trust normalization procedure has to be adapted. We suppose normalization of weights to the power of q, as has been discussed in Section 5.3.2.6. Let $\text{in}(x)$, the trust influx for agent x, be *positive*. As usual, we denote the global spreading factor by d, and quantified trust statements from x to y by $W(x, y)$. Function $\text{sign}(x)$ returns the sign of value x. Note that from now on, we assume $W : E \to [-1, +1]$, for degrees of *distrust* need to be expressible. Then the trust quantity $e_{x \to y}$ passed from x to successor y is computed as follows:

$$e_{x \to y} = d \cdot \text{in}(x) \cdot \text{sign}(W(x, y)) \cdot w, \tag{5.14}$$

where

$$w = \frac{|W(x, y)|^q}{\sum_{(x,s) \in E} |W(x, s)|^q}$$

The accorded quantity $e_{x \to y}$ becomes *negative* if $W(x, y)$ is negative, i.e., if x distrusts y. For the relative weighting, the *absolute* values $|W(x, s)|$ of all weights are considered. Otherwise, the denominator could become negative, or positive trust statements could become boosted unduly. The latter would be the case if the sum

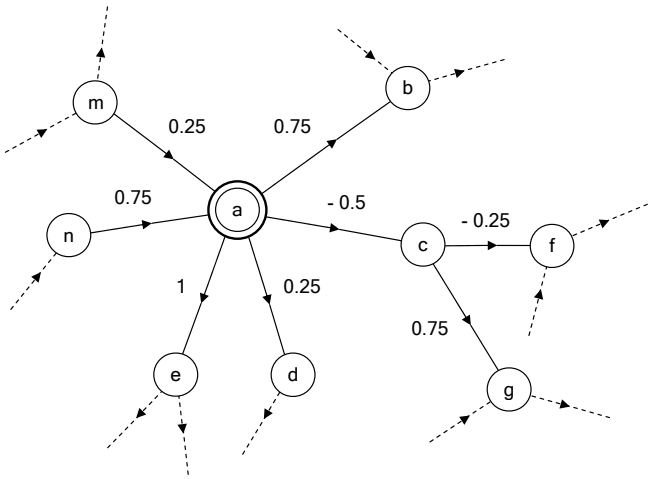

Figure 5.9. Network augmented by distrust

of positive trust ratings *only slightly* outweighed the sum of negative ones, making the denominator converge towards zero. An example demonstrates the computation process:

Example 5 (Distribution of Trust and Distrust) We assume the trust network as depicted in Figure 5.9. Let the trust energy influx into node a be $\text{in}(a) = 2$, and global spreading factor $d = 0.85$. For simplicity reasons, backward propagation of trust to the source is *not* considered. Moreover, we suppose *linear* weight normalization, thus $q = 1$. Consequently, the denominator of the normalization equation is $|0.75| + |-0.5| + |0.25| + |1| = 2.5$. The trust energy that a distributes to b hence amounts to $e_{a\to b} = 0.51$, whereas the energy accorded to the distrusted node c is $e_{a\to c} = -0.34$. Furthermore, we have $e_{a\to d} = 0.17$ and $e_{a\to e} = 0.68$.

Observe that trust energy becomes *lost* during distribution, for the sum of energy accorded along outgoing edges of a amounts to 1.02, while 1.7 was provided for distribution. The effect results from the negative trust weight $W(a, c) = -0.5$.

5.4.2.2 Distrust Allocation and Propagation

We now analyze the case where the influx $\text{in}(x)$ for agent x is *negative*. In this case, the trust allocated for x will also be negative, i.e., $\text{in}(x) \cdot (1 - d) < 0$. Moreover, the energy $\text{in}(x) \cdot d$ that x may distribute among successor nodes will be negative as well. The implications are those which have been mentioned in Section 5.4.1, i.e., distrust as negation of trust and propagation of distrust. For the first case, refer to

node f in Figure 5.9 and assume $\text{in}(c) = -0.34$, which is derived from Example 5. The trusted agent a distrusts c who distrusts f. Eventually, f would be accorded $d \cdot (-0.34) \cdot (-0.25)$, which is *positive*. For the second case, node g would be assigned the *negative* trust quantity $d \cdot (-0.34) \cdot (0.75)$, simply for being trusted by f, who is distrusted. Both unwanted effects can be avoided by not allowing distrusted nodes to distribute *any energy at all*. Hence, more formally, we introduce a novel function $\text{out}(x)$:

$$\text{out}(x) = \begin{cases} d \cdot \text{in}(x), & \text{if } \text{in}(x) \geq 0 \\ 0, & \text{else} \end{cases} \tag{5.15}$$

This function then has to replace $d \cdot \text{in}(x)$ when computing the energy distributed along edges from x to successor nodes y:

$$e_{x \to y} = \text{out}(x) \cdot \text{sign}(W(x, y)) \cdot w, \tag{5.16}$$

where

$$w = \frac{|W(x, y)|^q}{\sum_{(x,s) \in E} |W(x, s)|^q}$$

This design decision perfectly aligns with assumptions made in Section 5.4.1 and prevents the inference of unwanted side-effects mentioned before. Furthermore, one can see easily that the modifications introduced *do not affect* the behavior of Algorithm 5.3 when not considering relationships of distrust.

5.4.2.3 Convergence

In networks largely or entirely dominated by distrust, the extended version of Appleseed is still guaranteed to converge. We therefore briefly outline an informal proof, based on Proof 1:

Proof 2 (Convergence in presence of distrust) Recall that only *positive* trust influx $\text{in}(x)$ becomes propagated, which has been indicated in Section 5.4.2.2. Hence, all we need to show is that the overall quantity of *positive* trust distributed in computation step i cannot be augmented through the presence of distrust statements. In other words, suppose that $G = (A, E, W)$ defines an arbitrary trust graph, containing quantified trust statements, but *no distrust*, i.e., $W : E \to [0, 1]$. Now consider another trust graph $G' = (A, E \cup D, W')$, which contains additional edges D, and weight function $W' = W \cup (D \to [-1, 0[)$. Hence, G' augments G by additional distrust edges between nodes taken from A. We now perform two parallel computations with the extended version of Appleseed, one operating on G and the other on G'. In every step, and for every trust edge $(x, y) \in E$ for G, the distributed energy $e_{x \to y}$ is greater or equal to the respective counterpart on G', because the denominator

of the fraction given in Equation 5.16 can only become *greater* through additional distrust outedges. Second, for the computation performed on G', negative energy distributed along edge (x, y) can only *reduce* the trust influx for y and may hence even accelerate convergence. □

However, as can be observed from the proof, there exists one serious implication arising from having distrust statements in the network: the overall accorded trust quantity does *not* equal the initially injected energy anymore. Moreover, in networks dominated by distrust, the overall trust energy sum may even be *negative*.

Experiment 3 (Network impact of distrust) We observe the number of iterations until convergence is reached, and the overall accorded trust rank of 5 networks. The structures of all these graphs are identical, being composed of 623 nodes with an average indegree and outdegree of 9. The only difference applies to the assigned weights, where the first graph contains no distrust statements at all, while 25% of all weights are negative for the second, 50% for the third, and 75% for the fourth. The fifth graph contains nothing but distrust statements. The Appleseed parameters are identical for all 5 runs, having backward propagation enabled, an initial trust injection $\text{in}^0 = 200$, spreading factor $d = 0.85$, convergence threshold $T_c = 0.01$, *linear* weight normalization, and no upper bound on the number of nodes to unfold. The left-hand side of Figure 5.10 clearly demonstrates that the number of iterations until convergence, given on the vertical axis, *decreases* with the proportion of distrust increasing, observable along the horizontal axis. Likewise, the overall accorded trust rank, indicated on the vertical axis of the right-hand side of Figure 5.10, decreases rapidly with increasing distrust, eventually dropping below zero. The same experiment was repeated for another network with 329 nodes, an average indegree and outdegree of 6, yielding similar results.

The effects observable in Experiment 3 only marginally affect the ranking itself, for trust ranks are interpreted *relative* to each other. Moreover, compensation for lost trust energy may be achieved by boosting the initial trust injection in^0.

5.5 Discussion and Outlook

We provided a new classification scheme for trust metrics along three non-orthogonal feature axes. Moreover, we advocated the need for local group trust metrics, eventually presenting Appleseed, this chapter's main contribution. Appleseed's nature largely resembles Advogato, bearing similar complexity and attack-resistance properties, but offers one particular feature that makes Appleseed much more suitable for certain applications than Advogato: the ability to compute *rankings* of peers according to their trustworthiness rather than *binary* classifications into trusted and untrusted agents.

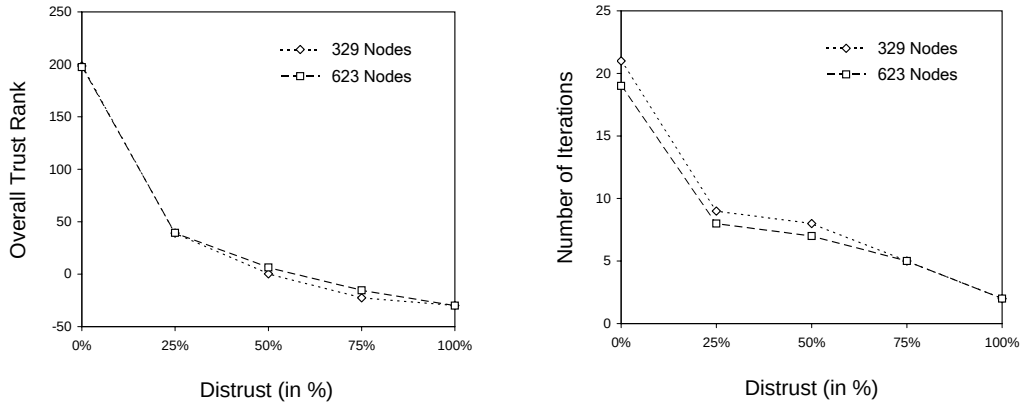

Figure 5.10. Network impact of distrust

Originally designed as an approach to social filtering within our decentralized recommender framework, Appleseed suits other application scenarios as well, such as group trust computation in online communities, open rating systems, ad-hoc and peer-to-peer networks:

For instance, Appleseed could support peer-to-peer-based file-sharing systems in reducing the spread of self-replicating inauthentic files by virtue of trust propagation [Kamvar et al., 2003]. In that case, explicit trust statements, resulting from direct interaction, would reflect belief in someone's endeavor to provide authentic files.

We strongly believe that local group trust metrics, such as Advogato and Appleseed, will become subject to substantial research for diverse computing domains within the near future, owing to their favorable time complexity and their intuitive computation scheme, as opposed to other classes of trust metrics (see Section 5.2.2.2). However, one has to bear in mind that their applicability is confined to particular problem domains only, whereas scalar metrics are more versatile.

Chapter 6

Interpersonal Trust and Similarity

"The essence of trust building is to emphasize the similarities between you and the customer."

– Thomas Watson (1874–1956)

Contents

6.1 Introduction

Recently, the integration of computational trust models [Marsh, 1994b; Mui et al., 2002; McKnight and Chervany, 1996] into recommender systems has started gaining momentum [Montaner et al., 2002; Kinateder and Rothermel, 2003; Guha, 2003; Massa and Bhattacharjee, 2004], synthesizing recommendations based upon opinions from *most trusted* peers rather than *most similar*[1] ones. Likewise, for social filtering within our decentralized recommender framework, we cannot rely upon conventional collaborative filtering methods only, owing to the neighborhood computation scheme's poor scalability (see Section 1.2). Some more natural and, most important,

[1] Requiring the explicit application of some similarity measure.

scalable neighborhood selection process becomes indispensable, e.g., based on trust networks.

However, in order to provide *meaningful* results, one should suppose trust to reflect user similarity to some extent. Clearly, recommendations only make sense when obtained from like-minded people having similar taste. For instance, Abdul-Rahman and Hailes [2000] claim that given some predefined domain and context, e.g., communities of people reading books, its members commence creating ties of friendship and trust primarily with persons resembling their *own* profile of interest. Jensen et al. [2002] make likewise assumptions, supposing similarity as a strong predictor of friendship: "If I am a classic car enthusiast, for example, my friends will likely share my interests [...]. In other words, my circle of friends is likely to either share the same values as I do, or at least tolerate them."

Reasons for that phenomenon are manyfold and mostly sociologically motivated, like people's striving for some sort of social affiliation [Given, 2002]. For instance, Pescovitz [2003] describes endeavors to identify trust networks for crime prevention and security. Hereby, its advocates operate "on the assumption that birds of a feather tend to flock together [...]", an ancient and widely-known aphorism. However, though belief in the positive relation of trust and user similarity has been widely adopted and presupposed, thus constituting the foundations for trust-based recommender and rating systems, to our best knowledge, no endeavors have been made until now to provide "real-world" empirical evidence.

Hence, we want to investigate and analyze whether the latter correlation actually holds, relying upon data mined from the All Consuming community, which has been introduced before in Section 3.4.1. Our studies involve several hundreds of members indicating which books they like and which other community members they trust. However, before presenting our framework for conducting trust-similarity correlation experiments, we provide an outline of recommender systems employing trust models, and an extensive survey giving results from socio-psychological research that bear some significant relevance for our analysis.

6.2 Trust Models in Recommender Systems

Sinha and Swearingen [2001] have found that people prefer receiving recommendations from people they *know* and *trust*, i.e., friends and family-members, rather than from online recommender systems. Some researchers have therefore commenced to focus on computational trust models as appropriate means to supplement or replace current collaborative filtering approaches.

Kautz et al. [1997] mine social network structures in order to render expertise information exchange and collaboration feasible. Olsson [1998] proposes an architecture combining trust, collaborative filtering and content-based filtering in one single framework, giving only vague information and insight, though. Another agent-based

approach has been presented by Montaner et al. [2002], who introduce the so-called "opinion-based" filtering. Montaner claims that trust should be *derived* from user similarity, implying that friends are exactly those people that resemble our very nature. However, Montaner's model only extends to the agent world and does not reflect evidence acquired from real-world social studies concerning the formation of trust. Similar agent-based systems have been devised by Kinateder and Rothermel [2003], Kinateder and Pearson [2003], and Chen and Singh [2001].

Apart from research in agent systems, online communities have also discovered opportunities through trust network leverage. Epinions (*http://www.epinions.com*) provides information filtering facilities based upon personalized webs of trust [Guha, 2003]. Hereby, Guha states that the trust-based filtering approach has been greatly approved and appreciated by Epinions' members. However, empirical and statistical justifications underpinning these findings, like indications of a correlation between trust and interest similarity, have not been subject to Guha's work. Likewise, Massa and Avesani [2004] operate on Epinions and propose superseding CF-based neighborhoods by trust networks, making use of very basic propagation schemes. Initial empirical data has been provided in their work, indicating that precision does not decrease too much when using trust-based neighborhood formation schemes instead of common CF.

Besides Epinions, All Consuming (see Section 3.4.1) represents another community combining ratings and trust networks[2]. Unlike Epinions, All Consuming only poorly exploits synergies between social filtering and trust.

6.3 Evidence from Social Psychology

Research in social psychology offers some important results for investigating interactions between trust and similarity. However, most relevant studies primarily focus on *interpersonal attraction* rather than trust, and its possible coupling with similarity. Interpersonal attraction constitutes a major field of interest of social psychology, and the positive impact of attitudinal similarity on liking has effectively become one of its most reliable findings [Berscheid, 1998]. Studies have given extensive attention to three different types of interpersonal relationships, namely same-sex friendships, primarily among college students, cross-sex romantic relationships, again primarily among college students, and marriage [Huston and Levinger, 1978]. Clearly, these three types of relationships also happen to be essential components of trust, though perfect equivalence does not hold. For instance, while friendship usually implies mutual trust, marriage does not. Moreover, the complex notion of interpersonal trust, already difficult to capture regarding the "lack of consensus" which has been pointed out by McKnight and Chervany [1996], interacts with other sociological concepts not

[2]When describing the All Consuming dataset in Chapter 3, we did not consider inter-subject trust information, owing to its irrelevance for the experiments at hand.

reflected through interpersonal attraction. These elusive components comprise reputation, skill, situational and dispositional aspects of interpersonal trust [Marsh, 1994a,b], and familiarity [Einwiller, 2003].

However, since explicit *trust* relationships have remained outside the scope of empirical analysis on the correlation with attitudinal similarity, we are forced to stick to *interpersonal* attraction instead. Clearly, results obtained must be treated with care before attributing them to interpersonal trust as well. The following paragraphs hence intend to briefly summarize relevant evidence collected from research on interpersonal attraction.

6.3.1 On Interpersonal Attraction and Similarity

Early investigations date back until 1943, when Burgess and Wallin published their work about homogamy of social attributes with respect to engaged couples [Burgess and Wallin, 1943]. Similarity could be established for nearly every characteristic examined. However, according to Berscheid [1998], these findings do not justify conclusions about positive effects of similarity on interpersonal attraction by themselves, since "part of the association between similarity and social choice undoubtedly is due not to personal preference, but to the fact that people tend to be thrown together in time and space with others similar to themselves".

First large-scale experimental studies were conducted by Newcomb [1961] and Byrne [1961, 1971]. The former work focused on friendships between American college students and nowadays counts among the seminal works on friendship formation. By means of his longitudinal study, Newcomb could reveal a positive association between attraction and attitudinal value similarity. Byrne, doing extensive research and experiments in the area of attraction, conducted similar experiments, applying the now-famous "bogus stranger technique" [Byrne, 1971]. The following section roughly outlines the original setup of this technique.

6.3.1.1 The Bogus Stranger Technique

First, all participating subjects had to indicate their preference on 26 diverse topics, ranging from more important ones, e.g., belief in super-natural beings, premarital sex, etc., to less important ones, like television programs, music taste, and so forth. Preference was expressed through 7-point likert scales. Two weeks later, the participants were falsely informed that they were now in a study on how well people can predict other people's behavior. In order to make these predictions, they were told that they would be given the attitude scales filled out by another participant. However, this was an outright lie. Actually, the scales were prepared by the experimenter, i.e., Byrnes and his assistants, in such way as to either reflect *high similarity* or *dissimilarity* with the subject's own profile. Participants were asked some questions thereafter about the respective "other participant", including personal sentiments

toward the person and how much they would appreciate working with him. Moreover, participants were requested to evaluate the "bogus stranger" with respect to his intelligence, knowledge of current events, morality, and adjustment.

6.3.1.2 Analysis of Similarity-Attraction Associations

The result of Byrne's bogus stranger experiment well aligned with Newcomb's findings and confirmed that attitude similarity is a determinant of attraction. Rather than further document this fact, which counts among the most reliable findings in social psychology today [Berscheid, 1998], researchers have ever since attempted to identify the factors that mediate and define the *limitations* of positive association between similarity and attraction.

For instance, along with other theorists, e.g., Festinger and Newcomb, Byrne conjectured that one's mere discovery of some other person holding similar attitudes is reinforcing in itself, arguing that "the expression of similar attitudes by a stranger serves as a positive reinforcement, because consensual validation for an individual's attitudes and opinions and beliefs is a major source of reward for the drive to be logical, consistent, and accurate in interpreting the stimulus world" [Byrne, 1971]. We suppose likewise effects when forging bonds of trust. Hence, the sheer observation that some other peer holds interests similar to our own, e.g., reading the same kinds of books, intuitively renders the latter more trustworthy in our eyes and engenders sentiments of "familiarity". In fact, automated collaborative filtering systems exploit this conjecture in order to make reliable predictions of product preference [Sinha and Swearingen, 2001].

Social psychologists have identified some other likely factors accounting for the similarity-attraction association. For example, the information that another person possesses similar attitudes may suggest his sympathy towards the individual, and "it is known that the anticipation of being liked often generates attraction in return" [Berscheid, 1998]. Jones et al. [1972] provided some large-scale empirical analysis for reciprocation of attraction from similar others.

6.3.1.3 Limitations

While positive association was attested for attitudinal similarity and interpersonal attraction, evidence could not be expanded to similarity in general. Berscheid [1998] therefore notes that despite "considerable effort to find a relationship between friendship choice and personality (as opposed to attitude) similarity, for example, the evidence for this hypothesis remains unconvincing [...]".

This inability to establish an association between personality similarity and attraction does not prove overly harmful to our hypothesis, since personal interests represent traits of *attitude* rather than *personality*. However, even attitude similarity fails to produce attraction under certain circumstances. Snyder and Fromkin [1980]

reveal that perceiving very high similarity with another individual may even evoke *negative* sentiments towards that respective person. Moreover, according to Heider [1958], "similarity can evoke disliking when the similarity carries with it disagreeable implications", which common sense anticipates, likewise. Take narcist persons as an example.

6.3.2 Conclusion

The preceding literature survey has shown that interactions between *similarity* traits and *interpersonal attraction* are difficult to capture. Even though the tight coupling between both concepts counts among social psychology's most reliable findings, there are numerous caveats to take into consideration, like subtle distinctions between various types of similarity, e.g., attitudinal similarity and personality similarity. Moreover, most studies argue that attitudinal similarity *implies* attraction, whereas the latter proposition's inversion, i.e., positing that similarity *follows* from attraction, has been subject to sparse research only. Common sense supports this thesis, though, since people tend to adopt attitudes of friends, spouses, etc.

6.4 Trust-Similarity Correlation Analysis

Even when taking reciprocal action between attitudinal similarity, and hence similarity of interests, and interpersonal attraction for granted, evidence from socio-psychological research does *not* provide sufficient support for positive interactions between *trust* and *interest similarity*. Mind that trust and interpersonal attraction, though subsuming several common aspects, e.g., friendship, familiarity, etc., are *not* fully compliant notions.

We hence intend to establish a formal framework for investigating interactions between trust and similarity, believing that given an application domain, such as, for instance, the book-reading domain, people's trusted peers are on average considerably more similar to their sources of trust than arbitrary peers. More formally, let A denote the set of all community members, $\text{trust}(a_i)$ the set of all users trusted by a_i, and $\text{sim} : A \times A \rightarrow [-1, +1]$ some similarity function:

$$\sum_{a_i \in A} \frac{\sum_{a_j \in \text{trust}(a_i)} \text{sim}(a_i, a_j)}{|\,\text{trust}(a_i)|} \gg \sum_{a_i \in A} \frac{\sum_{a_j \in A \setminus \{a_i\}} \text{sim}(a_i, a_j)}{|A| - 1} \tag{6.1}$$

For instance, given that agent a_i is interested in Science-Fiction and Artificial Intelligence, chances that a_j, trusted by a_i, also likes these two topics are much higher than for peer a_e not explicitly trusted by a_i. Various social processes are involved, such as participation in those social groups that best reflect our own interests and desires.

6.4.1 Model and Data Acquisition

In order to verify or refute our hypothesis for some specific domain, we need to define an information model, determine metrics and methods for evaluation, and apply our framework to real-world data.

6.4.1.1 Information Model

We assume the same model as the one presented in Section 3.3.1, but provide an extension for trust networks. Function trust : $A \rightarrow 2^A$ gives the set of all users that agent a_i trusts. Hence, for the scenario at hand, we assume trust as a relationship of *binary* preference, dividing the set of agents A into trusted and non-trusted ones for every $a_i \in A$. For user-user similarity computations $c(a_i, a_j) : A \times A \rightarrow [-1, +1]$, we employ our taxonomy-driven metric, presented in Section 3.3. Its huge advantage over pure CF similarity measures (see Section 2.3.2) lies in its ability to also work for *sparse* domains: when two users have no overlap in their purchased or rated products, pure CF measures become unable to make any reasonable inferences with respect to interest similarity, which is *not* the case for the taxonomy-driven method. Since All Consuming, the dataset we conduct all experiments upon, offers comparatively few ratings taken from a large product set, the ability to handle sparsity becomes an indispensable feature for eligible similarity metrics.

The following section now relates our supposed information model to an actual scenario, making use of variable and function bindings introduced above.

6.4.1.2 Data Acquisition

All Consuming is one of the few communities that allow members to express which other users they trust, as well as which items, in our case books, they appreciate. Hereby, users may *import* their list of trusted persons from other applications like FOAF [Dumbill, 2002]. All Consuming also offers to automatically compile information about books its members have read from their personal weblog. In contrast to the All Consuming dataset described in Chapter 3, the one used in this chapter has been crawled earlier, from October 13 to October 17, 2003.

Our tools mined data from about $2,074$ weblogs contributing to the All Consuming information base, and 527 users issuing 4.93 trust statements on average. These users have mentioned $6,592$ different books altogether. In order to obtain category descriptors $f(b_k)$ for all discovered books b_k, classification information from the Amazon.com online shop (*http://www.amazon.com*) was garnered. For each book, we collected an average of about 4.1 classification topics, relating books to Amazon.com's book taxonomy.

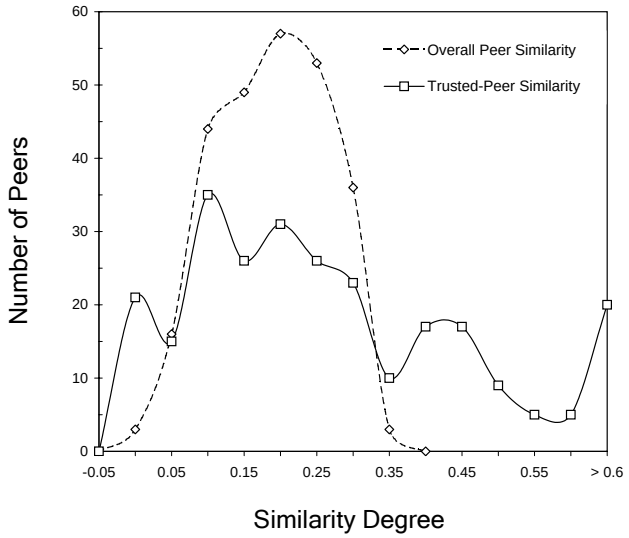

Figure 6.1. Histogram representation of the upper bound analysis

6.4.2 Experiment Setup and Analysis

This section describes the two experiments we performed in order to analyze possible positive correlations between interest similarity and interpersonal trust. In both cases, experiments were run on data obtained from All Consuming (see Section 6.4.1.2). Considering the slightly different information makeup the two experiments were based upon, we expected the first to define an *upper bound* for the analysis, and the second one a *lower bound*. Results obtained confirmed our assumption.

6.4.2.1 Upper Bound Analysis

Before conducting the two experiments, we applied extensive data cleansing and duplicate removal to the All Consuming *active user* base of 527 members[3]. First, we pruned all users a_i having fewer than 3 books mentioned, removing them from user base A and from all sets trust(a_j) where $a_i \in$ trust(a_j). Next, we discarded all users a_i who did not issue any trust assertions at all. Interestingly, some users had created *several* accounts. We discovered these "duplicates" by virtue of scanning through account names for similarity patterns and by tracking identical or highly similar profiles in terms of book mentions. Moreover, we stripped self-references, i.e., statements about users trusting themselves. Through application of data cleansing,

[3]All Consuming's crawled weblogs were *not* considered for the experiments, owing to their lack of trust web information.

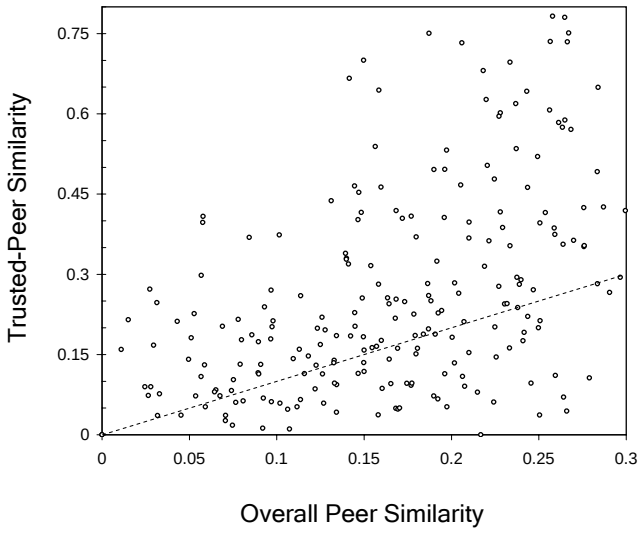

Figure 6.2. Scatter plot for the upper bound analysis

266 users were discarded from the initial test set, leaving 261 users for the upper bound experiment to run upon. We denote the reduced set of users by A' and corresponding trust functions by $\text{trust}'(a_i)$.

For every single user $a_i \in A'$, we generated his profile vector and computed the similarity score $c(a_i, a_j)$ for each *trusted* peer $a_j \in \text{trust}'(a_i)$. Next, we averaged these proximity measures, obtaining value z'_i:

$$z'_i := \frac{\sum_{a_j \in \text{trust}'(a_i)} c(a_i, a_j)}{|\,\text{trust}'(a_i)|} \tag{6.2}$$

Moreover, we computed a_i's similarity with any other user from dataset A', except a_i himself. Again, we took the average of these proximity measures and recorded the result s'_i:

$$s'_i := \frac{\sum_{a_j \in A' \setminus \{a_i\}} c(a_i, a_j)}{|A'| - 1} \tag{6.3}$$

A comparison of pairs (z'_i, s'_i) revealed that in 173 cases, users were more similar to their trusted peers than arbitrary ones. The opposite held for only 88 agents. Users had an average similarity score of 0.247 with respect to their trusted peers, while only exhibiting 0.163 with complete A'. In other words, users were *more than* 50% more similar to their trusted agents than arbitrary peers.

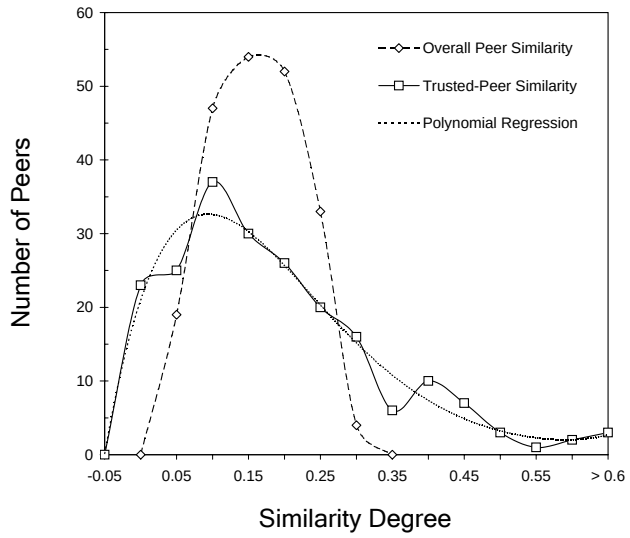

Figure 6.3. Histogram representation of the lower bound analysis

Distributions of z' and s'

Figure 6.1 gives histogram representations for z' and s', respectively. No agents have higher average similarity than 0.4, i.e., $s'_i \leq 0.4$ holds for all $a_i \in A'$. This is not the case for z', as there remains a considerable amount of users a_i exhibiting an average trusted-peer similarity z'_i larger than 0.4. About 20 agents have $z'_i > 0.6$. Interestingly, while the overall peer similarity s' shows an almost perfect Gaussian distribution curve, its counterpart z' does not feature the typical bell shape. This observation raises some serious concerns when conducting analysis of statistical significance in Section 6.4.3.

Scatter Plot

In order to directly match every user's overall similarity s'_i against his average trusted-peer similarity z'_i, Figure 6.2 provides a scatter plot for the experiment at hand. The dashed line, dividing the scheme into an upper and lower region, models an agent a_i having *identical* similarity values, i.e., $s'_i = z'_i$. Clearly, the plot exhibits a strong bias towards the upper region, which becomes particularly pronounced for agents a_i with $s'_i > 0.15$.

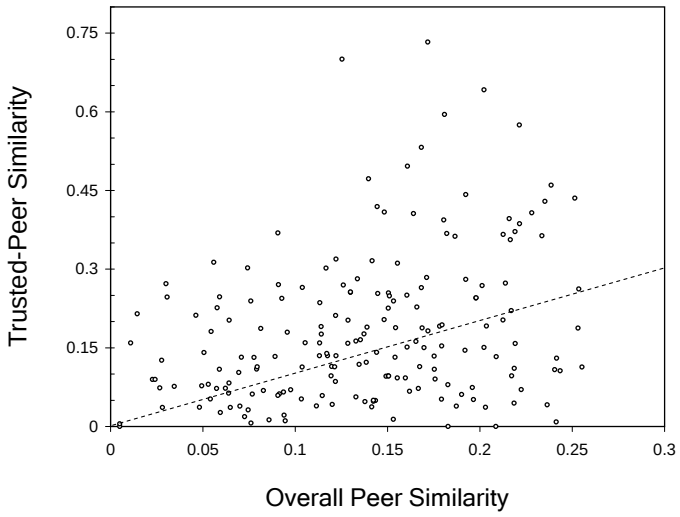

Figure 6.4. Scatter plot for the lower bound analysis

6.4.2.2 Lower Bound Analysis

The first experiment proposed that users tend to trust people that are significantly more similar to themselves than average users. However, we have to consider that All Consuming offers a feature that *suggests friends* to newbie users a_i. Hereby, All Consuming chooses users who have *at least one book in common* with a_i. Hence, we have reason to suspect that our first experiment was biased and too optimistic with respect to positive interactions between trust and similarity. Consequently, we pruned user set A' once again, eliminating trust statements whenever trusting and trusted user had at least one book in common. We denote the latter user base by A'', now reduced to 210 trusting users, and indicate its respective trust functions by $trust''(a_i)$.

Clearly, our approach to eliminate All Consuming's intrusion into the natural process of trust formation entails the removal of many "real" trust relationships between users a_i and a_j, i.e., relationships which had been forged owing to a_i actually *knowing* and *trusting* a_j, and not because of All Consuming proposing a_j as an appropriate match for a_i.

For the second experiment, we computed values s_i'' and z_i'' for every $a_i \in A''$. We supposed results to be biased to the *disadvantage* of our conjecture, i.e., unduly lowering possible positive associations between trust and user similarity. Again, one should bear in mind that for set A'', users did not have one single book in common with their trusted peers.

Results obtained from the second experiment corroborate our expectations, being less indicative for existing positive interactions between interpersonal trust and attitudinal similarity. Nevertheless, similarity values z_i'' still exceeded s_i'': in 112 cases, people were more similar to their trusted fellows than arbitrary peers. The opposite held for 98 users. Mean values of z'' and s'' amounted to 0.164 and 0.134, respectively. Hence, even for the lower bound experiment, users were still approximately 23% more similar to their trusted fellows than arbitrary agents.

Histogram Curves

The bell-shaped distribution of s'', depicted in Figure 6.3, looks more condensed with respect to s' and has its peak slightly below the latter plot's curve. The differences between z'' and z' are even more pronounced, though, e.g., the shape of z'''s histogram looks more "regular" than z''s pendant. Hence, the approximation of z'''s distribution, applying polynomial regression of degree 5, strongly resembles the Erlang-k distribution, supposing $k = 2$. For similarity degrees above 0.35, peaks of z'''s histogram are considerably less explicit than for z'' or have effectively disappeared, as is the case for degrees above 0.6.

Matching z_i'' Against s_i''

Figure 6.4 gives the scatter plot of our lower bound analysis. The strong bias towards the upper region has become less articulate, though still clearly visible. Interestingly, the increase of ratio $z_i'' : s_i''$ for $s_i'' > 0.15$ still persists.

6.4.3 Statistical Significance

We conclude our experimental analysis noticing that without exact knowledge of how much noise All Consuming's "friend recommender" adds to our obtained results, we expect the *true* correlation intensity between trust and interest similarity to reside somewhere within our computed upper and lower bound.

Moreover, we investigated whether the increase of mean values of z' with respect to s', and z'' with respect to s'', bears statistical significance or not. For the analyses at hand, common parametrical one-factor ANOVA could *not* be applied to z' and s', and z'' and s'', likewise, for diverse reasons:

Gaussian distribution. The distributions of both samples have to be *normal*, even though small departures may be accommodated. While s' and s'' exhibit the latter Gaussian distribution property, z' and z'' obviously do not.

Equal variances. Data transformation, e.g., logarithmic, probits, etc., might be an option for z'', bearing traits of Erlang-2. However, ANOVA also demands largely identical *variances* σ^2. Since z'''s variance is 5.33 times the variance of s'', this criterion cannot be satisfied.

Hence, owing to these two limitations, we opted for Kruskal-Wallis non-parametric ANOVA [Siegel and Castellan, 1988], which does not make any assumptions with respect to distribution and variance.

	n	**Rank Sum**	**Mean Rank**
z'	261	73702.0	284.56
s'	261	60719.0	234.44
		Kruskal-Wallis Statistic	14.52
		p	**0.0001**

Table 6.1. Kruskal-Wallis ANOVA test results for the upper bound experiment

Table 6.1 shows result parameters obtained from analyzing the upper bound experiment. Since value p is much smaller than 0.05, very high statistical significance holds, thus refuting the hypothesis that fluctuations between medians of s' and z' were caused by mere random.

For the lower bound experiment, on the other hand, no statistical significance was detected, indicated by large p and a low Kruskal-Wallis statistic being much smaller than 1 (see Table 6.2).

6.4.4 Conclusion

Both experiments suggest that the mean similarity of trusting and trusted peers exceeds the arbitrary user similarity. For the upper bound analysis, strong statistical significance was discovered, which was not the case for its lower bound pendant. However, assuming the true distribution curves to reside somewhere in between these bounds, and taking into account that both z' and z'' exhibit larger mean values than s' and s'', respectively, the results we obtained bear strong indications towards positive interactions between interpersonal trust and interest similarity.

6.5 Discussion and Outlook

In this chapter, we articulated our hypothesis that positive mutual interactions between interpersonal trust and user similarity exist when the community's trust network is tightly bound to some particular application, e.g., reading books. Before presenting an evaluation framework to conduct empirical analyses, we provided an extensive literature survey on relevant socio-psychological research. We then applied

	n	**Rank Sum**	**Mean Rank**
z''	210	43685.0	210.02
s''	210	43051.0	206.98
		Kruskal-Wallis Statistic	0.07
		p	**0.796**

Table 6.2. Kruskal-Wallis ANOVA test results for the lower bound experiment

our evaluation method, using the taxonomy-driven similarity computation technique presented in Chapter 3, to the All Consuming community.

We believe that our positive results will have substantial impact for ongoing research in recommender systems, where discovering similar users is of paramount importance. Decentralized approaches will especially benefit from trust network leverage. Hereby, the outstanding feature of trust networks lies in their ability to allow for sensible prefiltering of like-minded peers and to increase the *credibility* of recommendations. Arbitrary social networks, on the other hand, only allow for reducing the computational complexity when composing neighborhoods.

Though backing our experiments with information involving several hundreds of people, we believe that additional efforts studying trust-similarity interactions in domains other than book-reading are required in order to further corroborate our hypothesis. Unfortunately, at the time of this writing, the large-scale penetration of trust networks into communities, particularly those where users are given the opportunity to rate products, still has to take place.

Chapter 7

Decentralized Recommender Systems

"In nature we never see anything isolated, but everything in connection with something else [...]"

– Johann Wolfgang von Goethe (1749–1832)

Contents

7.1 Introduction

Preceding chapters, particularly Chapter 3 and Chapter 5, have presented methods and techniques that are, among other things, able to address specific issues of *decentralized* recommender systems. Moreover, Chapter 6 has shown that, to a certain extent, trust *implies* similarity and thus becomes eligible as a tool for CF neighborhood formation, which is generally performed by applying some rating-based or attribute-based similarity measure (see Section 2.3.2).

In this chapter, we integrate our prior contributions into one coherent, decentralized recommender framework, sufficing the criteria stated in Section 1.1.2, i.e., distributed data storage and centralized computation. The presented architecture illustrates one sample approach, others are likewise conceivable and may represent

equally appropriate solutions. Outstanding features of our method are the strong focus on trust and the usage of taxonomy-driven similarity measures.

Trust networks allow for circumventing the complexity issue that those recommender systems that operate in large decentralized settings are facing. Mind that similarity-based neighborhood computation impersonates a severe bottleneck, owing to the $O(|A|^2)$ time complexity when composing neighborhoods for all $|A|$ members part of the system. Hence, these approaches do not scale. On the other hand, network-based propagation approaches, e.g., Appleseed or Advogato (see Chapter 5), necessitate partial graph exploration only and scale to arbitrary network sizes. Second, the low rating profile overlap issue [Sarwar et al., 2000b] that large communities with effectively unconstrained product sets are confronted with, investigated in detail by Massa and Bhattacharjee [2004] for the well-known Epinions community (*http://www.epinions.com*), is addressed through taxonomy-driven similarity computations (see Chapter 3).

Besides describing the make-up of our decentralized recommender framework, we conduct empirical analyses and compare results with benchmarks from two centralized architectures, namely one purely content-based system, and the taxonomy-driven filtering system proposed in Chapter 3.

Advocacy for Trust-based Neighborhood Formation

As stated above, we investigate social network structures in order to easily assemble personalized neighborhoods for active users a_i. To give an example of network-based neighborhood formation, a_i's neighborhood may comprise exactly those peers being closest in terms of *link distance*, necessitating simple breadth-first search instead of $O(|A|)$ complexity, which is required for computing similarity measures between one single a_i and all other individuals in the system. More specifically, we exclusively focus on *trust* relationships, motivated by reasons given below:

- **Security and attack-resistance.** Closed communities generally possess efficient means to control the user's identity and penalize malevolent behavior. However, decentralized systems cannot prevent deception and insincerity. Spoofing and identity forging thus become facile to achieve and allow for luring people into purchasing products which may provide some benefit for attackers a_o [Lam and Riedl, 2004; O'Mahony et al., 2004]. For instance, to accomplish such attacks, agents a_o simply have to copy victim a_v's rating profile and add excellent ratings for products b_k they want to trick a_v into buying. Owing to high similarities between rating profiles of a_o and a_v, b_k's probability of being proposed to a_v quickly soars beyond competing products' recommendation likelihood. On the other hand, only proposing products from people the active user deems most trustworthy inherently solves this issue, hence excluding perturbations from unknown and malevolent agents from the outset.

- **Recommendation transparency.** One of the major disadvantages of recommender systems relates to their lacking transparency, i.e., users would like to understand *why* they were recommended particular goods [Herlocker et al., 2000]. However, algorithmic clockworks of recommenders effectively resemble black boxes. Hence, when proposing products from users based upon complex similarity measures, most of these "neighbors" probably being unknown to the active user, recommendations become difficult to follow. On the other hand, recommendations from trustworthy people clearly exhibit higher acceptance probability. Recall that trust metrics operate on naturally grown social network structures while neighborhoods based upon interest similarity represent pure artifacts, computed according to some obscure scheme.
- **Correlation of trust and similarity.** Sinha and Swearingen [2001] have found that people tend to prefer receiving recommendations from people they *know* and *trust*, i.e., friends and family-members, rather than from online recommender systems. Moreover, positive mutual impact of attitudinal similarity on interpersonal attraction counts among one of the most reliable findings of modern social psychology [Berscheid, 1998], backing the proverbial saying that "birds of a feather flock together". In Chapter 6, we have provided first empirical evidence confirming the positive correlation between trust and interest similarity.
- **Mitigating the new-user cold-start problem.** One major weakness that CF systems are faced with is the so-called new-user cold-start problem [Middleton et al., 2002]: newbie members generally have issued few product ratings only. Consequently, owing to common product-user matrix sparseness and low profile overlap, appropriate similarity-based neighbors are difficult to find, entailing poor recommendations. The whole process is self-destructive, for users discontinue to use the recommender system before the latter reaches acceptable performance. Trust networks alleviate cold-start issues by virtue of comparatively high network connectivity. Neighborhood formation hence becomes practicable even for users that explicitly trust one person only, taking into account an abundant transitive trust closure (see Section 7.3.1.1 for details).

Note that when computing neighborhoods based upon types of social relationships other than trust, e.g., geographical proximity, acquaintanceship, etc., the benefits given above may not become fully exploited.

7.2 Related Work

Decentralized recommenders have been proposed in the past already. Foner [1997, 1999] has been the first to conceive of decentralized recommender systems, pursuing an agent-based approach to the matchmaking of like-minded peers. Similarity

is determined by means of feature extraction in documents, e.g., electronic mails, articles, and so forth. Olsson [1998, 2003] builds upon Foner's work and proposes another multi-agent system that addresses the issue of peers self-organizing into similarity-based clusters without central control. In both systems, the amount of messages routed through the network may prove problematic, creating a severe bottleneck. Kinateder and Rothermel [2003] propose an architecture for reputation-based matchmaking that is similar to Foner's and Olsson's approach. However, no empirical evaluations have been provided so far.

Miller [2003] explores various matchmaking techniques known from research in peer-to-peer networks, e.g., the Gnutella protocol for random discovery, Chord, and transitive traversal. Similarities between peers are computed based upon item-based and user-based CF. Hence, the sparsity issue is not part of Miller's investigations.

A taxonomization of decentralized recommender architectures along various dimensions is given by Sarwar [2001].

Systems using trust for recommendation making are still rare. Mui et al. [2001] propose *collaborative sanctioning* as suitable surrogate for collaborative filtering. The idea is to make recommendations based on user *reputation* rather than user similarity. Montaner [2003] uses *trust* rather than *reputation*[1] in his multi-agent framework. Trust is regarded as direct, non-propagatable and a mere consequence of similarity. Massa and Avesani [2004] present a trust network-based recommender system, making use of simple propagation schemes. Initial results appear promising, though, at the time of this writing, their architecture still undergoes several adaptations for decentralized scenarios.

7.3 Framework Components

The following subsections and paragraphs briefly outline our decentralized, trust-based recommender system's core constituents. The information model assumed represents the union of the filtering model given in Section 3.3.1, and the trust model from Section 5.2.2.1, renaming trust functions $W_i(a_j)$ to $t_i(a_j)$ for convenience and ease of reading. Consequently, the underlying model features agents $a_i \in A$, products $b_k \in B$, implicit ratings $R_i \subseteq B$, taxonomy C and descriptors $f : B \rightarrow 2^D$, and explicit trust functions $t_i : A \rightarrow [-1, +1]^{\perp}$.

7.3.1 Trust-based Neighborhood Formation

The computation of trust-based neighborhoods is 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.

[1] See Section 5.2.1 for distinguishing both concepts from each other.

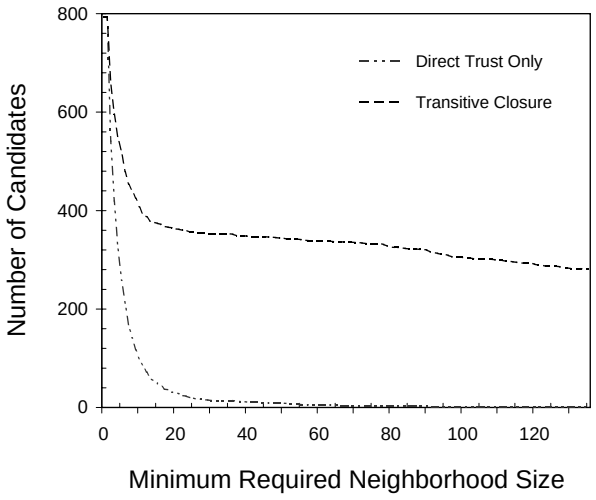

Figure 7.1. Reach of direct trust versus transitive closure

7.3.1.1 Network Connectivity

However, as has been indicated before, trust functions t_i assigning *explicit* trust ratings are generally sparse. When also taking into account *indirect* trust relationships, thus exploiting the "conditional transitivity" property of trust [Abdul-Rahman and Hailes, 1997], the assembly of neighborhoods that contain M most trustworthy peers becomes possible even for larger M, e.g., $M \geq 50$. Figure 7.1 backs our hypothesis, analyzing the connectivity of 793 users from the All Consuming community (see Section 3.4.1). The chart shows the number of users, indicated on the y-axis, who satisfy the minimum neighborhood size criterion given along the x-axis. For instance, while 49 people have issued 15 or more *direct* trust statements, 374 users are able to reach 15 or more peers when also considering the *transitive closure* of trust relationships. While the trust outdegree curve decays rapidly, the transitive closure curve's fallout decelerates drastically as the number of candidate persons drops below 400, thus revealing the presence of one highly connected trust cluster (see Figure 7.2)[2].

The above result relates to the classical theorem on random graphs [Erdős and Rényi, 1959].[3] Therein, Erdős and Rényi have proved that in large graphs $G =$

[2]The network has been visualized with our trust testbed viewer, presented in Section 5.3.2.9.

[3]Watts and Strogatz [1998] have shown that social networks exhibit diverse "small-world" properties that make them different from random graphs, such as high clustering coefficients $C(p)$. Barabási and Albert [1999] have investigated further distinctive features, such as the scale-free nature of social networks, which is not present in random graphs. Even though, the aforementioned theorem holds for random graphs and social networks alike.

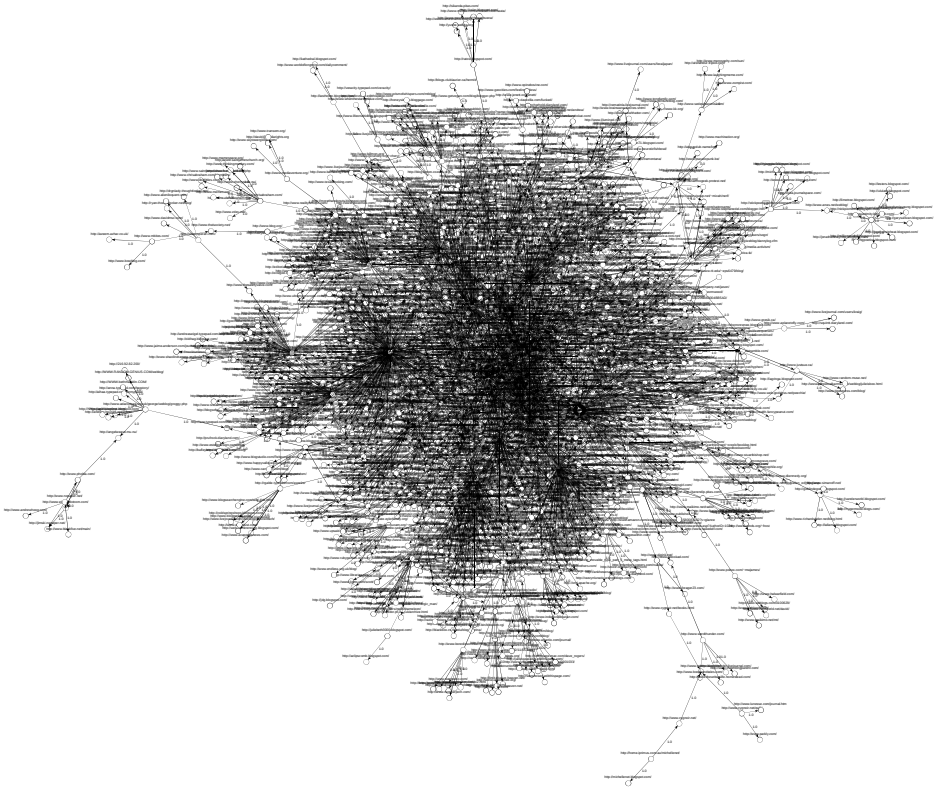

Figure 7.2. All Consuming's largest trust cluster

(V, E), assuming E randomly assigned, the probability of getting a single gigantic component jumps from zero to one as E/V increases beyond the critical value 0.5. However, Erdős and Rényi have supposed *undirected* graphs, in contrast to our assumption of *directed* trust relationships.

Massa and Bhattacharjee [2004] have conducted experiments on top of the well-known Epinions rating community, revealing that "trust-aware techniques can produce trust scores for very high numbers of peers". Neighborhood formation thus becomes facile to achieve when considering reachability of nodes via trust paths.

7.3.1.2 Trust Propagation Models

Trust-based neighborhood computation 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. The trust network's high connectivity allows assembling top-M trusted neighborhoods with

potentially large M.

Numerous scalar metrics [Beth et al., 1994; Levien and Aiken, 1998] 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 top-M *nearest trust neighbors*, and not evaluate trust values for any two given agents. We hence opt for *local group trust metrics* (see Chapter 5), e.g., Advogato and Appleseed. Advogato, Levien's well-known local group trust metric, can only make *boolean* decisions with respect to trustworthiness, simply classifying agents into trusted and non-trusted ones.

Appleseed, on the other hand, allows more fine-grained analysis, assigning continuous trust weights for peers within trust computation range. Rankings thus become feasible. The latter metric operates on partial trust graph information, exploring the social network within predefined ranges only and allowing the neighborhood formation process to retain scalability. High ranks are accorded to trustworthy peers, i.e., those agents which are largely trusted by others with high trustworthiness. These ranks are used later on for selecting agents deemed suitable for making recommendations.

7.3.2 Measuring User Similarity and Product-User Relevance

Trust allows selecting peers with overall above-average interest similarity (see Chapter 6). However, for each active user a_i, some highly trusted peers a_j having completely opposed interests generally exist. The proposition that interpersonal attraction, and hence trust, implies attitudinal similarity does not always hold true. Supplementary filtering, preferably content-based, e.g., considering a_i's major fields of interest, thus becomes indispensable.

For this purpose, we apply taxonomy-driven methods to likewise compute user similarity $c(a_i, a_j)$ and product-user relevance $c_b(a_i, b_k)$, according to the computational schemes given in Section 3.3.3.1 and 3.3.4, respectively. These metrics have been designed with decentralized scenarios in mind, for common filtering metrics based upon rating vector similarity (see Section 2.3.2) tend to fail in these settings [Massa and Bhattacharjee, 2004], owing to information sparseness implied by virtually unconstrained product sets and sparse, largely implicit, rating information.

7.3.3 Recommendation Generation

Candidate recommendations of products b_k for the active user a_i are taken from the set of products that a_i's top-M neighbors have implicitly rated, discounting those products that a_i already knows. We obtain set B_i of candidate products. Next, all $b_k \in B_i$ need to be weighted according to their *relevance* for a_i. Relevance $w_i(b_k)$ hereby depends on two factors:

- **Computed trust weights $t_i^c(a_j)$ of peers a_j mentioning b_k.** Trust-based neighborhood formation supersedes finding nearest neighbors based upon interest similarity. Likewise, similarity ranks $c(a_i, a_j)$ become substituted by trust weights $t_i^c(a_j)$ for computing the predicted relevance of a_j 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. For example, one may consider the situation where even close friends recommend products not fitting our interest profile at all.

We then define relevance $w_i(b_k)$ of product b_k for the active user a_i as follows, borrowing from Equation 3.6:

$$w_i(b_k) = \frac{q \cdot c_b(a_i, b_k) \cdot \sum_{a_j \in A_i(b_k)} \rho(a_i, a_j)}{|A_i(b_k)| + \Upsilon_R}, \tag{7.1}$$

where

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

and

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

In accordance with Section 3.3.4, $\text{prox}(a_i)$ denotes a_i's neighborhood, Γ_T and Υ_R represent those fine-tuning parameters introduced therein. Function $\rho(a_i, a_j)$ gives a_j's *significance* for a_i. In Equation 3.6, the latter parameter has been instantiated with the taxonomy-based user-user similarity weight $c(a_i, a_j)$.

Since we now suppose trust-based neighborhoods, $\rho(a_i, a_j) := t_i^c(a_j)$ holds.

7.4 Offline Experiments and Evaluation

The following sections present empirical results obtained from evaluating our trust-based approach for decentralized social filtering. Again, we gathered information from the All Consuming online community featuring both trust network information and product rating data. Our analysis mainly focused on pinpointing the impact that latent information kept within the trust network, namely positive interactions between interpersonal trust and attitudinal similarity (see Chapter 6), may have on recommendation quality. We performed empirical offline evaluations, applying metrics introduced and used before, e.g., precision/recall according to Sarwar's definition [Sarwar et al., 2000b], and Breese score (see Section 2.4.1.2).

7.4.1 Dataset Acquisition

Currently, few online communities provide both trust *and* product rating information. To our best knowledge, Epinions and All Consuming count among the only prospective candidates. Unfortunately, Epinions has two major drawbacks that are highly pernicious for our purposes. First, owing to an immense product range diversity, most ratable products lack content meta-information. Taxonomy-based filtering thus becomes unfeasible. Second, rating information sparseness is beyond measure. For instance, Massa and Bhattacharjee [2004] have pointed out that only 8.34% of all ratable products have 10 or more reviews.

We therefore opted for the All Consuming community, which has its product range thoroughly confined to the domain of books. For the experiments at hand, we performed a third crawl, following those described in Chapter 6 and 3. Launched on May 10, 2004, the community crawl garnered information about $3,441$ users, mentioning $10,031$ distinct book titles in $15,862$ implicit book ratings. The accompanying trust network consisted of $4,282$ links. For $9,374$ of all $10,031$ books, $31,157$ descriptors pointing to Amazon.com's book taxonomy were found. Book ratings referring to one of those $6,55\%$ of books not having valid taxonomic content descriptors were discarded.

One can see that using the All Consuming dataset only partially exploits functionalities our trust-based recommender system is able to unfold. For instance, the Appleseed trust metric has been conceived with *continuous* trust and distrust statements in mind, whereas All Consuming only offers statements of *full trust*.

7.4.2 Evaluation Framework

The principal objective of our evaluation was to match the trust-based neighborhood formation scheme against other, more common approaches. Hereby, all benchmark systems were devised according to the same algorithmic clockwork, i.e., based upon the recommendation generation framework defined in Equation 7.1. Their only difference refers to the kind of neighborhood formation, depending on function $\rho(a_i, a_j)$, which identifies the *relevance* of peers a_j for the active user a_i. Besides the trust-based recommender described in Section 7.3.3, the following two recommender setups have been used for experimentation:

- **Advanced hybrid approach.** Hybrid filtering likewise exploits content-based and collaborative filtering facilities. Designed to eliminate intrinsic drawbacks of both mentioned types, this approach currently represents the most promising paradigm for crafting state-of-the-art recommender systems. The hybrid recommender we propose features *similarity-based* neighborhood formation, requiring $\rho(a_i, a_j) := c(a_i, a_j)$. Consequently, aside from diversification factor Θ_F, the approach is identical to the taxonomy-driven filtering framework

presented in Chapter 3. Therein, we have also substantiated its superior performance over common benchmark recommender systems (see Section 3.4.2.4). However, note that this recommender's applicability is largely restricted to *centralized* scenarios only, necessitating similarity computations $c(a_i, a_j)$ for *all pairs* $(a_i, a_j) \in A \times A$.

- **Purely content-based filter.** Purely content-driven recommender systems *ignore* aspects of collaboration among peers and focus on content-based information only. We simulate one such recommender by supposing $\rho(a_i, a_j) := \text{rnd}_{[0,1]}(a_i, a_j)$, where function $\text{rnd}_{[0,1]} : A \times A \rightarrow [0, 1]$ randomly assigns relevance weights to pairs of agents. Neighborhood formation thus amounts to an *arbitrary sampling* of users, devoid of meaningful similarity criteria. Discarding collaboration, generated recommendations are not subject to mere random, though. They rather depend on product features, i.e., measure $c_b(a_i, b_k)$, only. Hence this recommender's purely content-based nature.

Past efforts have shown that intelligent hybrid approaches tend to outperform purely content-based ones [Huang et al., 2002; Pazzani, 1999]. We are particularly interested in beneficial ramifications resulting from trust-based neighborhood formation as opposed to random neighborhoods. Supposing that latent semantic information about interpersonal trust and its positive association with attitudinal similarity, endogenous to the very network, has forged sufficiently strong bonds, we conjecture that the overall recommendation quality of our trust-based approach surpasses filtering based upon content only.

7.4.2.1 Experiment Setup

The evaluation framework we established intends to compare the *utility* of recommendation lists generated by all three recommenders and roughly complies with the framework proposed in Chapter 3. Measurement is achieved by applying precision, recall, and Breese's half-life utility metric (see Section 2.4.1.2).

For cross-validation, we selected all users a_i with more than five ratings and discarded those having fewer, owing to the fact that reasonable recommendations are beyond feasibility for these cases. Moreover, users having low trust connectivity were likewise discounted. Next, we applied 5-folding, performing 80/20 splits of every user a_i's implicit ratings R_i into five pairs of *training sets* R_i^x and *test sets* T_i^x.

7.4.2.2 Parameterization

For our first experiment, neighborhood formation size was set to $M = 20$, and we provided top-20 recommendations for each active user's training set R_i^x. Proximity between profiles, based upon R_i^x and the original ratings R_j of all other agents a_j, was computed anew for each training set R_i^x of a_i.

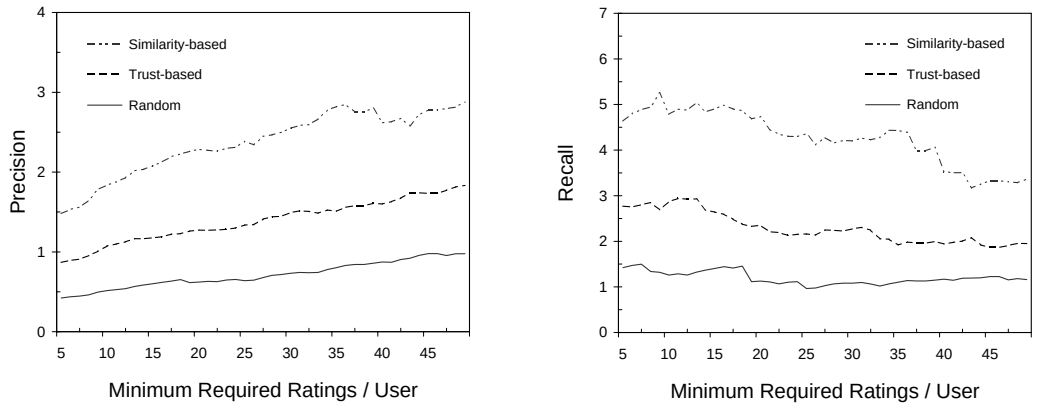

Figure 7.3. Precision and recall, investigating neighborhood formation

In order to promote the impact that collaboration may have on eventual recommendations, we adopted $\Upsilon_R = 2.25$, thus rewarding books occurring frequently in ratings R_j of the active user a_i's immediate neighborhood. For content-based filtering, this parameter exerts marginal influence only. Moreover, we assumed propagation factor $\kappa = 0.75$, and topic reward $\Gamma_T = 0.1$.

7.4.3 Experiments

We conducted three diverse experiments. The first compares the effects of neighborhood formation on recommendation quality when assuming raters with varying numbers of ratings. The second investigates neighborhood size sensitivity for all three candidate schemes, while the third measures overlap of neighborhoods.

7.4.3.1 Neighborhood Formation Impact

For the first experiment, performance was analyzed by computing unweighted precision and recall (see Figure 7.3), and Breese score with half-life $\alpha = 5$ and $\alpha = 10$ (see Figure 7.4). For each indicated chart, the *minimum numbers* of ratings that users were required to have issued in order to be considered for recommendation generation and evaluation are expressed by the horizontal axis. Since all users with fewer than five ratings were ignored from the outset, performance evaluations start with all users having at least five ratings. Clearly, larger x-coordinates imply less agents considered for measurement.

Remarkably, by looking at the *differences* between the curves, more important for our analysis than the very shapes, all four charts confirm our principal hypothesis

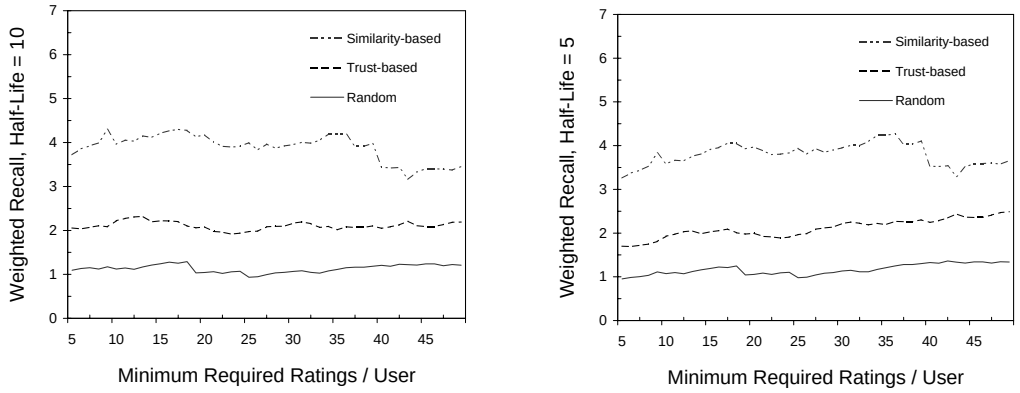

Figure 7.4. Breese score with $\alpha \in \{5, 10\}$, investigating neighborhood formation

that hybrid approaches outperform purely content-based ones. Hence, promoting products that like-minded agents have voted for increases recommendation quality considerably. Next, we observe that our trust-based recommender significantly exceeds its purely content-based counterpart, but cannot reach the hybrid method's superior score. These results again corroborate our assumption that trust networks contain latent knowledge that reflects interest similarity between trusted agents. Clearly, trust-based neighborhood formation can only *approximate* neighborhoods assembled by means of similarity, which therefore serves as upper bound definition. However, recall that similarity-based neighborhood formation exhibits poor scalability, owing to its $O(|A|^2)$ complexity that arises from computing proximity measures $c(a_i, a_j)$ for all pairs $(a_i, a_j) \in A \times A$. Hence, this neighborhood formation scheme is not an option for decentralized recommender system infrastructures.

Trust-based clique formation, on the other hand, *does* scale and lends itself well to decentralized settings. Moreover, it bears several welcome features that similarity-based neighborhood formation does not (see Section 7.1).

The following few paragraphs investigate the *shapes* of the curves we obtained in a more fine-grained fashion. As a matter of fact, the experiment at hand corroborates our hypothesis that trust networks, in contrast to arbitrary connections between agents, bear inherent information about similarity that improves recommendation quality.

Precision

Interestingly, precision (see Figure 7.3) steadily increases even for content-based filtering. The reason for this phenomenon lies in the very nature of precision: for users a_i with test sets T_i^x smaller than the number $|P_i^x|$ of recommendations received,

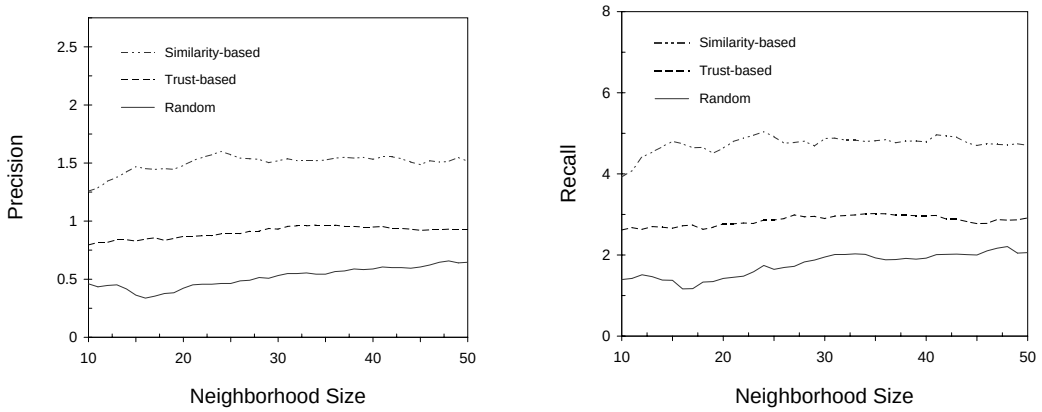

Figure 7.5. Precision and recall for varying neighborhood sizes

there is not even a chance of achieving 100% precision (see Section 3.4.2.4).

Recall

Degradation takes place for all curves when increasing x, an effect that is particularly pronounced for the hybrid recommender. Sample inspections of the All Consuming dataset suggest that infrequent raters favor bestsellers and popular books. Consequently, recommending popular books, promoted by large factor $\Upsilon_R = 2.25$, represents an appropriate guess for that particular type of users. However, when considering users possessing more refined profiles, simple "cherry picking" [Herlocker et al., 2004] does not apply anymore.

Breese Score

Scores for half-life $\alpha = 5$ and $\alpha = 10$ (see Figure 7.4) only exhibit marginal variances with respect to unweighted recall. However, degradation for increasing x becomes less pronounced when supposing lower α[4], i.e., $\alpha = 10$ and eventually $\alpha = 5$.

7.4.3.2 Neighborhood Size Sensitivity

The second experiment analyzes the impact of the neighborhood's *size* on evaluation metrics. Note that we omitted charts for weighted recall, owing to minor deviations

[4]Recall that unweighted recall equates Breese score with $\alpha = \infty$.

from unweighted recall only. Figure 7.5 indicates scores for precision and recall for increasing neighborhood size $|M|$ along the horizontal axis.

Both charts exhibit similar tendencies for each neighborhood formation scheme. As it comes to similarity-based neighborhood formation, the performance of the hybrid approach steadily increases at first. Upon reaching its peak at $|M| = 25$, further increasing neighborhood size $|M|$ does not entail any gains in precision and recall anymore. This result well aligns with Sarwar's investigations for baseline collaborative filtering techniques [Sarwar et al., 2001]. Undergoing slight downward movements between $|M| = 10$ and $|M| = 15$, the content-based scheme's performance curve catches up softly. Basically, increasing the neighborhood size for the content-based filter equates to offering more candidate products[5] and easing "cherry-picking" [Herlocker et al., 2004] by virtue of large $\Upsilon_R = 2.25$.

In contrast to both other techniques, the trust-based approach shows comparatively *insensitive* to increasing neighborhood size $|M|$. As a matter of fact, its performance only marginally improves. We attribute this observation to trust's "conditional transitivity" [Abdul-Rahman and Hailes, 1997] property and Huang's investigations on transitive associations for collaborative filtering [Huang et al., 2004]: exploitation of *transitive* trust relationships, i.e., opinions of friends of friends, only works to a certain extent. However, with increasing network distance from the trust source, these peers do not satisfactorily reflect interest similarity anymore and thus represent weak predictors only. Besides empirical evidence of a positive correlation between interpersonal trust and interest similarity, as well as its positive impact on recommendation quality, we regard this aspect as one of the most important findings of our study at hand.

7.4.3.3 Neighborhood Overlap Analysis

Eventually, we compared neighborhoods formed by those three techniques. For any unordered pair $\{p, q\}$ of the three neighborhood formation techniques, we measured the number of agents a_j occurring in *both* x-sized neighborhoods of every active user $a_i \in A$, and normalized the figure by clique size x and the number of agents $|A|$:

$$s^x(\{p,q\}) = \frac{\sum_{a_i \in A} |\operatorname{prox}_p^x(a_i) \cap \operatorname{prox}_q^x(a_i)|}{|A| \cdot x} \tag{7.2}$$

Figure 7.6 shows all three plots of $s^x(\{p,q\}), x \in [0, 50]$. All curves exhibit tendencies of approximatively linear rise for increasing neighborhood size x, for the probability of overlap rises when neighborhoods become larger. Consequently, supposing clique size $x = |A|$, 100% overlap holds.

As expected, both curves displaying overlap with randomly formed neighborhoods *only marginally* differ from each other. On the other hand, the overlap between

[5]Note that only products rated by neighbors are considered for recommendation.

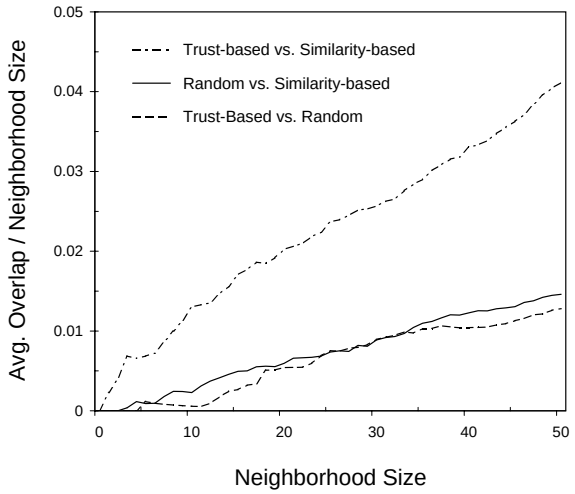

Figure 7.6. Overlap of x-sized neighborhoods for all formation scheme pairs

trust-based and similarity-based cliques exceeds these two baseline plots, showing that trust-based and similarity-based neighborhoods are considerably more similar to each other than pure random would allow. The above experiment again strongly corroborates our hypothesis that interpersonal trust and attitudinal similarity correlate.

7.5 Conclusion and Outlook

This chapter introduced an approach to exploit trust networks for product recommending, making use of techniques and evidence stemming from preceding chapters of this work. Superseding common collaborative approaches with trust-based filtering becomes vital when envisaging *decentralized* recommender system infrastructures, lacking central authorities.

We devised a new hybrid recommender architecture, based on the framework presented in Chapter 3, that makes use of trust-based neighborhood formation and taxonomy-driven selection of suitable products. Moreover, we provided empirical evidence to show that network structures emanating from relationships of interpersonal trust, in contrast to random associations between users, exhibit traits of interest similarity which significantly improve recommendation quality. However, we also found that trust's tight coupling with similarity becomes lost when overly exploiting *transitive* relationships.

For our experiments, we used data from the All Consuming book reading commu-

nity which offers both rating *and* trust information about its users. Note that most reputation and rating systems operating upon trust models only use *synthesized* rather than *real* trust data, therefore allowing largely limited analysis of trust semantics only. However, we would like to base our investigations upon richer datasets in order to make our results more reliable. Unfortunately, few communities currently exist that offer accessible bulk information about both trust relationships *and* product rating data of its users. We expect this situation to change within the next years to come, owing to the steadily increasing public interest in trust networks, which is particularly promoted by the advent of weblogs and the Semantic Web. In this area, i.e., weblogs and the Semantic Web, we also see the main applicability of our proposed architecture. As the below paragraph demonstrates, current developments and trends already point into the right direction, providing an infrastructure that allows to easily leverage personal, decentralized product recommendation services into the Web.

Deployment Scenario

Referring to the information model the envisioned decentralized recommender operates upon, the model's single components can be instantiated as follows:

- **Trust networks.** FOAF (see Section 1.2) defines machine-readable homepages based upon RDF [6] and allows weaving acquaintance networks. Golbeck et al. [2003] have proposed some modifications making FOAF support "real" trust relationships instead of mere acquaintanceship.
- **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. Those links that refer to product pages from large catalogs like 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.
- **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

[6] See *http://www.w3.org/RDF/* for specifications of the RDF standard.

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, videos, and apparel, to name some.
Standardization efforts for the classification of diverse kinds of consumer goods 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.

Eventually, we come to conclude that the information infrastructure required for the decentralized recommender approach described in this chapter may soon turn into reality, fostering future research on information filtering through social networks and yielding valuable large-scale evidence.

Chapter 8

Conclusion

"The end we aim at must be known, before the way can be made."

– Jean Paul (1763-1825)

Contents

8.1 Summary

Undoubtedly, recommender systems are becoming increasingly popular, owing to their versatility and their ability to reduce complexity for the human user. Current research greatly benefits from cross-fertilization, including results from other disciplines like economics (see, e.g., Sénécal [2003]), and behavioral sciences on the verge of HCI (see, e.g., [Swearingen and Sinha, 2001; Jensen et al., 2002]). The integration of interdisciplinary evidence also represents an important ingredient of this thesis, and our research on trust propagation in social networks expands the current research focus into new directions, namely that of the emerging science of social networks [Newman, 2003], and social psychology, investigating semantics of interpersonal trust.

Our research and contributions made derive from issues that appear when transplanting centralized recommender systems, serving well-defined, closed communities, into anarchical large-scale networks, e.g., the Semantic Web, the Grid, peer-to-peer and ad-hoc networks. These issues were outlined in Chapter 1. We then devised approaches to tackle these single issues, e.g., neighborhood formation based on propagation of trust in social networks in order to address the credibility and scalability problem, etc., amalgamating them into one sample framework for decentralized recommendation making. These contributions fall into mainly two categories, both being substantially different from each other:

- **Information Filtering.** The procedure of taxonomy-driven profile creation, presented in Chapter 3, lies at the heart of our advanced filtering approach and has been designed with information sparseness in mind. We integrated taxonomy-driven similarity metrics into a new framework for making product recommendations and provided comparisons with benchmark approaches.
  Moreover, we distilled the topic diversification technique, an integral part of the afore-mentioned framework, and applied this particular procedure on top of conventional collaborative filtering systems in order to make top-N recommendation lists more meaningful (see Chapter 4). An extensive large-scale study involving more than 2,100 human subjects and offline analyses were conducted, investigating the effects of incrementally applying topic diversification to lists generated by item-based CF and user-based CF. In addition, the study delivered a first, empirically backed argument supporting the hypothesis that "accuracy does not tell the whole story" [Cosley et al., 2002] and that there are more components to user satisfaction than pure accuracy. [1]
- **Computational Trust.** Our contributions in the field of trust and trust networks are twofold. First we introduced a new trust metric, Appleseed, which is based on spreading activation models [Quillian, 1968] known from cognitive psychology, and blends traits of PageRank [Page et al., 1998] and maximum network flow [Ford and Fulkerson, 1962]. Appleseed makes inferences in an intuitive fashion, respecting subtle semantic differences between trust and distrust, and scales to any network size. We devised Appleseed with neighborhood formation for CF in decentralized scenarios in mind.
  To this end, so that trust-based neighborhoods are *meaningful* for CF applications, we conceived an evaluation framework to investigate whether interpersonal trust and interest similarity correlate, i.e., if users trusting each other were on average more similar than mere random would foretell. Again, similarity was measured by applying our taxonomy-driven similarity metric (see Chapter 3). Results obtained from an offline study on All Consuming (*http://www.allconsuming.net*) indicated that positive interactions exist (see Chapter 6), supporting the proverbial saying that "birds of a feather flock together" and levelling the ground for the application of trust-based neighborhood formation in CF systems.

Eventually, those single building bricks were put together to build a trust-based, decentralized recommender system (see Chapter 7) able to address those issues outlined in Chapter 1. Note that the presented decentralized recommender's design constitutes *one possible option* among various others, giving opportunities for future research.

[1] Some researchers, e.g., Herlocker et al. [2004] and Hayes et al. [2002], have raised this concern before, but have not provided any evidence whatsoever to substantiate their assertion.

8.2 Discussion and Outlook

As a matter of fact, we see the underlying thesis' foremost strength in its versatility and variety, making contributions in diverse fields. These single contributions are not necessarily confined to the recommender system universe, but also extend to other research fields. For instance, Brosowski [2004] investigates the application of Appleseed for trust-based spam filtering in electronic mails, Nejdl [2004] considers our trust metric for distributed Web search infrastructures, and Chirita et al. [2004] discuss the utility of Appleseed for personalized reputation management in peer-to-peer networks. On the other hand, the integration of all these diverse mosaic stones into a coherent framework for decentralized information filtering, exploiting mutual synergies, gives the broader *context* and provides the *component glue* of this work's various facets. The framework itself is by no means complete, though, which is indicated by the adverb "towards" in the underlying thesis' title.

In fact, research on trust network-based recommender systems has just begun, and now starts to attract increased research interest [Massa and Avesani, 2004; Bonhard, 2004, 2005; O'Donovan and Smyth, 2005; Papagelis et al., 2005]. Except for [Massa and Avesani, 2004], all of these works are still in their infancies and still have to prove their viability. An aggravating factor for decentralized, trust-based recommender systems, at the time of this writing, is certainly the fact that little data is currently available to base experiments upon. Most datasets *either* feature product ratings *or* trust networks. We expect this situation to drastically change in the near future, owing to the steadily increasing popularity of social networking applications [Pescovitz, 2003; Fitzgerald, 2004].

Hence, the road ahead remains vague and the direction unclear. But the journey will be an interesting and revealing one, full of marvels and curiosities our minds did not anticipate. This thesis has already set some important landmarks and made bold strides into the direction of more social and network-oriented recommender systems. Numerous other landmarks will follow in the near future and shape an exciting new landscape.

Bibliography

Abdul-Rahman, A. and Hailes, S. 1997. A distributed trust model. In *New Security Paradigms Workshop*. Cumbria, UK, 48–60.

Abdul-Rahman, A. and Hailes, S. 2000. Supporting trust in virtual communities. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*. Maui, HI, USA.

Aberer, K. and Despotovic, Z. 2001. Managing trust in a peer-2-peer information system. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, H. Paques, L. Liu, and D. Grossman, Eds. ACM Press, Atlanta, GA, USA, 310–317.

Aggarwal, C., Wolf, J., Wu, K.-L., and Yu, P. 1999. Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, San Diego, CA, USA, 201–212.

Ali, K. and van Stam, W. 2004. TiVo: Making show recommendations using a distributed collaborative filtering architecture. In *Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, Seattle, WA, USA, 394–401.

Alspector, J., Kolcz, A., and Karunanithi, N. 1998. Comparing feature-based and clique-based user models for movie selection. In *Proceedings of the Third ACM Conference on Digital Libraries*. ACM Press, Pittsburgh, PE, USA, 11–18.

Armitage, P. and Berry, G. 2001. *Statistical Methods in Medical Research*, 3rd ed. Blackwell Science, Oxford, UK.

Avery, C. and Zeckhauser, R. 1997. Recommender systems for evaluating computer messages. *Communications of the ACM 40,* 3 (March), 88–89.

Baeza-Yates, R. and Ribeiro-Neto, B. 1999. *Modern Information Retrieval*. Addison-Wesley, Reading, MA, USA.

Balabanović, M. and Shoham, Y. 1997. Fab: Content-based, collaborative recommendation. *Communications of the ACM 40,* 3 (March), 66–72.

Barabási, A.-L. and Albert, R. 1999. Emergence of scaling in random networks. *Science 286*, 509–512.

Basilico, J. and Hofmann, T. 2004. Unifying collaborative and content-based filtering. In *Proceedings of the 21st International Conference on Machine Learning*. ACM Press, Banff, Canada.

Baudisch, P. 2001. Dynamic information filtering. Ph.D. thesis, Technische Universität Darmstadt, Darmstadt, Germany.

Berscheid, E. 1998. Interpersonal attraction. In *The Handbook of Social Psychology*, 4th ed., D. Gilbert, S. Fiske, and G. Lindzey, Eds. Vol. II. McGraw-Hill, New York, NY, USA.

Beth, T., Borcherding, M., and Klein, B. 1994. Valuation of trust in open networks. In *Proceedings of the 1994 European Symposium on Research in Computer Security*. Brighton, UK, 3–18.

Blaze, M., Feigenbaum, J., and Lacy, J. 1996. Decentralized trust management. In *Proceedings of the 17th Symposium on Security and Privacy*. IEEE Computer Society Press, Oakland, CA, USA, 164–173.

Bonhard, P. 2004. Improving recommender systems with social networking. In *Proceedings Addendum of the 2004 ACM Conference on Computer-Supported Cooperative Work*. Chicago, IL, USA.

Bonhard, P. 2005. Who do trust? combining recommender systems and social networking for better advice. In *Proceedings of the IUI 2005 Beyond Personalization Workhop*. San Diego, CA, USA. Position paper.

Brafman, R., Heckerman, D., and Shani, G. 2003. Recommendation as a stochastic sequential decision problem. In *Proceedings of ICAPS 2003*. Trento, Italy.

Breese, J., Heckerman, D., and Kadie, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, Madison, WI, USA, 43–52.

Brosowski, M. 2004. *Webs of Trust in Distributed Environments: Bringing Trust to Email Communication*. B.S. thesis, Information Systems Institute, University of Hannover.

Budanitsky, A. and Hirst, G. 2000. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Proceedings of the Workshop on WordNet and Other Lexical Resources*. Pittsburgh, PA, USA.

Burgess, E. and Wallin, P. 1943. Homogamy in social characteristics. *American Journal of Sociology 2,* 49, 109–124.

Burke, R. 2002. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction 12,* 4, 331–370.

Byrne, D. 1961. Interpersonal attraction and attitude similarity. *Journal of Abnormal and Social Psychology* 62, 713–715.

Byrne, D. 1971. *The Attraction Paradigm.* Academic Press, New York, NY, USA.

Ceglowski, M., Coburn, A., and Cuadrado, J. 2003. Semantic search of unstructured data using contextual network graphs.

Chen, M. and Singh, J. 2001. Computing and using reputations for internet ratings. In *Proceedings of the 3rd ACM Conference on Electronic Commerce.* ACM Press, Tampa, FL, USA, 154–162.

Chen, R. and Yeager, W. 2003. Poblano: A distributed trust model for peer-to-peer networks. Tech. rep., Sun Microsystems, Santa Clara, CA, USA. February.

Chirita, P.-A., Nejdl, W., Schlosser, M., and Scurtu, O. 2004. Personalized reputation management in P2P networks. In *Proceedings of the ISWC 2004 Workshop on Trust, Security and Reputation.* Hiroshima, Japan.

Cosley, D., Lawrence, S., and Pennock, D. 2002. REFEREE: An open framework for practical testing of recommender systems using ResearchIndex. In *28th International Conference on Very Large Databases.* Morgan Kaufmann, Hong Kong, China, 35–46.

Deshpande, M. and Karypis, G. 2004. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems 22,* 1, 143–177.

Dumbill, E. 2002. Finding friends with XML and RDF. IBM's XML Watch.

Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. 2001. Rank aggregation methods for the Web. In *Proceedings of the Tenth International Conference on World Wide Web.* ACM Press, Hong Kong, China, 613–622.

Einwiller, S. 2003. The significance of reputation and brand in creating trust between an online vendor and its customers. In *Trust in the Network Economy*, O. Petrovic, M. Fallenböck, and C. Kittl, Eds. Springer-Verlag, Heidelberg, Germany, 113–127.

Erdős, P. and Rényi, A. 1959. On random graphs. *Publicationes Mathematicae 5*, 290–297.

Eschenauer, L., Gligor, V., and Baras, J. 2002. On trust establishment in mobile ad-hoc networks. Tech. Rep. MS 2002-10, Institute for Systems Research, University of Maryland, MD, USA. October.

Fagin, R., Kumar, R., and Sivakumar, D. 2003. Comparing top-k lists. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms.* SIAM, Baltimore, MD, USA, 28–36.

Ferman, M., Errico, J., van Beek, P., and Sezan, I. 2002. Content-based filtering and personalization using structured metadata. In *Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries*. ACM Press, Portland, OR, USA, 393–393.

Fitzgerald, M. 2004. Internetworking. *M.I.T. Technology Review*, 44–49. Technology Review, Inc.

Foner, L. 1997. Yenta: A multi-agent, referral-based matchmaking system. In *Proceedings of the First International Conference on Autonomous Agents*. ACM Press, Marina del Rey, CA, USA, 301–307.

Foner, L. 1999. Political artifacts and personal privacy: The Yenta multi-agent distributed matchmaking system. Ph.D. thesis, Massachusetts Institute of Technology, Boston, MA, USA.

Ford, L. and Fulkerson, R. 1962. *Flows in Networks*. Princeton University Press, Princeton, NJ, USA.

Gans, G., Jarke, M., Kethers, S., and Lakemeyer, G. 2001. Modeling the impact of trust and distrust in agent networks. In *Proceedings of the Third International Bi-Conference Workshop on Agent-oriented Information Systems*. Montreal, Canada.

Gaul, W. and Schmidt-Thieme, L. 2002. Recommender systems based on user navigational behavior in the internet. *Behaviormetrika 29,* 1, 1–22.

Ghani, R. and Fano, A. 2002. Building recommender systems using a knowledge base of product semantics. In *Proceedings of the Workshop on Recommendation and Personalization in E-Commerce (RPEC)*. Springer-Verlag, Malaga, Spain.

Given, B. 2002. *Teaching to the Brain's Natural Learning Systems*. Association for Supervision and Curriculum Development, Alexandria, VA, USA.

Golbeck, J. and Hendler, J. 2004. Accuracy of metrics for inferring trust and reputation in Semantic Web-based social networks. In *Proceedings of the 14th International Conference on Knowledge Engineering and Knowledge Management*. Northamptonshire, UK.

Golbeck, J., Parsia, B., and Hendler, J. 2003. Trust networks on the Semantic Web. In *Proceedings of Cooperative Intelligent Agents*. Helsinki, Finland.

Goldberg, D., Nichols, D., Oki, B., and Terry, D. 1992. Using collaborative filtering to weave an information tapestry. *Communications of the ACM 35,* 12, 61–70.

Goldberg, K., Roeder, T., Gupta, D., and Perkins, C. 2001. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval 4,* 2, 133–151.

Good, N., Schafer, B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., and Riedl, J. 1999. Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the 16th National Conference on Artificial Intelligence and Innovative Applications of Artificial Intelligence*. American Association for Artificial Intelligence, Orlando, FL, USA, 439–446.

Guha, R. 2003. Open rating systems. Tech. rep., Stanford Knowledge Systems Laboratory, Stanford, CA, USA.

Guha, R., Kumar, R., Raghavan, P., and Tomkins, A. 2004. Propagation of trust and distrust. In *Proceedings of the Thirteenth International World Wide Web Conference*. ACM Press, New York, NY, USA.

Hartmann, K. and Strothotte, T. 2002. A spreading activation approach to text illustration. In *Proceedings of the 2nd International Symposium on Smart Graphics*. ACM Press, Hawthorne, NY, USA, 39–46.

Hayes, C. and Cunningham, P. 2004. Context-boosting collaborative recommendations. *Knowledge-Based Systems 17,* 2-4 (May), 131–138.

Hayes, C., Massa, P., Avesani, P., and Cunningham, P. 2002. An online evaluation framework for recommender systems. In *Proceedings of the Workshop on Personalization and Recommendation in E-Commerce (RPEC)*. Springer-Verlag, Malaga, Spain.

Heider, F. 1958. *The Psychology of Interpersonal Relations*. Wiley, New York, NY, USA.

Herlocker, J., Konstan, J., Borchers, A., and Riedl, J. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, Berkeley, CA, USA, 230–237.

Herlocker, J., Konstan, J., and Riedl, J. 2000. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer-Supported Cooperative Work*. Philadelphia, PA, USA, 241–250.

Herlocker, J., Konstan, J., and Riedl, J. 2002. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval 5,* 4, 287–310.

Herlocker, J., Konstan, J., Terveen, L., and Riedl, J. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems 22,* 1, 5–53.

Holland, P. and Leinhardt, S. 1972. Some evidence on the transitivity of positive interpersonal sentiment. *American Journal of Sociology 77*, 1205–1209.

Housely, R., Ford, W., Polk, W., and Solo, D. 1999. Internet X.509 public key infrastructure. Internet Engineering Task Force RFC 2459.

Huang, Z., Chen, H., and Zeng, D. 2004. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems 22,* 1, 116–142.

Huang, Z., Chung, W., Ong, T.-H., and Chen, H. 2002. A graph-based recommender system for digital library. In *Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries*. ACM Press, Portland, OR, USA, 65–73.

Huston, T. and Levinger, G. 1978. Interpersonal attraction and relationships. *Annual Review of Psychology 29*, 115–156.

Jensen, C., Davis, J., and Farnham, S. 2002. Finding others online: Reputation systems for social online spaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM Press, Minneapolis, MN, USA, 447–454.

Jones, E., Bell, L., and Aronson, E. 1972. The reciprocation of attraction from similar and dissimilar others. In *Experimental Social Psychology*, C. McClintock, Ed. Holt, Rinehart, and Winston, New York, NY, USA.

Jøsang, A., Gray, E., and Kinateder, M. 2003. Analysing topologies of transitive trust. In *Proceedings of the Workshop of Formal Aspects of Security and Trust*. Pisa, Italy.

Kamvar, S., Schlosser, M., and Garcia-Molina, H. 2003. The EigenTrust algorithm for reputation management in P2P networks. In *Proceedings of the Twelfth International World Wide Web Conference*. Budapest, Hungary.

Karypis, G. 2001. Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the Tenth ACM CIKM International Conference on Information and Knowledge Management*. ACM Press, Atlanta, GA, USA, 247–254.

Kautz, H., Selman, B., and Shah, M. 1997. Referral Web: Combining social networks and collaborative filtering. *Communications of the ACM 40,* 3 (March), 63–65.

Kinateder, M. and Pearson, S. 2003. A privacy-enhanced peer-to-peer reputation system. In *Proceedings of the 4th International Conference on Electronic Commerce and Web Technologies*. LNCS, vol. 2378. Springer-Verlag, Prague, Czech Republic.

Kinateder, M. and Rothermel, K. 2003. Architecture and algorithms for a distributed reputation system. In *Proceedings of the First International Conference on Trust Management*, P. Nixon and S. Terzis, Eds. LNCS, vol. 2692. Springer-Verlag, Crete, Greece, 1–16.

Kleinberg, J. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM 46,* 5, 604–632.

Konstan, J. 2004. Introduction to recommender systems: Algorithms and evaluation. *ACM Transactions on Information Systems 22,* 1, 1–4.

Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J. 1997. GroupLens: Applying collaborative filtering to usenet news. *Communications of the ACM 40,* 3, 77–87.

Kummamuru, K., Lotlikar, R., Roy, S., Singal, K., and Krishnapuram, R. 2004. A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In *Proceedings of the 13th International Conference on World Wide Web*. ACM Press, New York, NY, USA, 658–665.

Lam, S. and Riedl, J. 2004. Shilling recommender systems for fun and profit. In *Proceedings of the 13th International Conference on World Wide Web*. ACM Press, New York, NY, USA, 393–402.

Lam, W., Mukhopadhyay, S., Mostafa, J., and Palakal, M. 1996. Detection of shifts in user interests for personalized information filtering. In *Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, Zürich, Switzerland, 317–325.

Lang, K. 1995. NewsWeeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*. Morgan Kaufmann, San Mateo, CA, USA, 331–339.

Levien, R. 2004. Attack-resistant trust metrics. Ph.D. thesis, University of California at Berkeley, Berkeley, CA, USA. To appear.

Levien, R. and Aiken, A. 1998. Attack-resistant trust metrics for public key certification. In *Proceedings of the 7th USENIX Security Symposium*. San Antonio, TX, USA.

Levien, R. and Aiken, A. 2000. An attack-resistant, scalable name service. Draft submission to the Fourth International Conference on Financial Cryptography.

Lewicki, R., McAllister, D., and Bies, R. 1998. Trust and distrust: New relationships and realities. *Academy of Management Review 23,* 12, 438–458.

Linden, G., Smith, B., and York, J. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing 4,* 1 (January).

Luhmann, N. 1979. *Trust and Power*. Wiley, Chichester, UK.

Malone, T., Grant, K., Turbak, F., Brobst, S., and Cohen, M. 1987. Intelligent information-sharing systems. *Communications of the ACM 30,* 5, 390–402.

Marsh, S. 1994a. Formalising trust as a computational concept. Ph.D. thesis, Department of Mathematics and Computer Science, University of Stirling, Stirling, UK.

Marsh, S. 1994b. Optimism and pessimism in trust. In *Proceedings of the Ibero-American Conference on Artificial Intelligence*, J. Ramirez, Ed. McGraw-Hill, Caracas, Venezuela.

Massa, P. and Avesani, P. 2004. Trust-aware collaborative filtering for recommender systems. In *Proceedings of the DOA/CoopIS/ODBASE Confederated International Conferences (1)*, R. Meersman and Z. Tari, Eds. LNCS, vol. 3290. Springer-Verlag, Larnaca, Cyprus, 492–508.

Massa, P. and Bhattacharjee, B. 2004. Using trust in recommender systems: an experimental analysis. In *Proceedings of the 2nd International Conference on Trust Management*, C. Jensen, S. Poslad, and T. Dimitrakos, Eds. LNCS, vol. 2995. Springer-Verlag, Oxford, UK.

Maurer, U. 1996. Modelling a public key infrastructure. In *Proceedings of the 1996 European Symposium on Research in Computer Security*, E. Bertino, Ed. LNCS, vol. 1146. Springer-Verlag, Rome, Italy, 325–350.

McKnight, H. and Chervany, N. 1996. The meaning of trust. Tech. Rep. MISRC 96-04, Management Information Systems Research Center, University of Minnesota, MN, USA.

McNee, S., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S., Rashid, A., Konstan, J., and Riedl, J. 2002. On the recommending of citations for research papers. In *Proceedings of the 2002 ACM Conference on Computer-Supported Cooperative Work*. ACM Press, New Orleans, LA, USA, 116–125.

Melville, P., Mooney, R., and Nagarajan, R. 2002. Content-boosted collaborative filtering for improved recommendations. In *Eighteenth National Conference on Artificial Intelligence*. American Association for Artificial Intelligence, Edmonton, Canada, 187–192.

Middleton, S., Alani, H., Shadbolt, N., and De Roure, D. 2002. Exploiting synergy between ontologies and recommender systems. In *Proceedings of the WWW2002 International Workshop on the Semantic Web*. CEUR Workshop Proceedings, vol. 55. Maui, HI, USA.

Middleton, S., De Roure, D., and Shadbolt, N. 2001. Capturing knowledge of user preferences: Ontologies in recommender systems. In *Proceedings of the First International Conference on Knowledge Capture*. Victoria, Canada.

Middleton, S., Shadbolt, N., and De Roure, D. 2004. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems 22,* 1, 54–88.

Milgram, S. 1992. The small world problem. In *The Individual in a Social World - Essays and Experiments*, 2nd ed., J. Sabini and M. Silver, Eds. McGraw-Hill, New York, NY, USA.

Miller, B. 2003. Toward a personalized recommender system. Ph.D. thesis, University of Minnesota, Minneapolis, MA, USA.

Miyahara, K. and Pazzani, M. 2000. Collaborative filtering with the simple bayesian classifier. In *Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence*. Melbourne, Australia, 679–689.

Montaner, M. 2003. Collaborative recommender agents based on case-based reasoning and trust. Ph.D. thesis, Universitat de Girona, Girona, Spain.

Montaner, M., López, B., and de la Rosa, J. 2002. Opinion-based filtering through trust. In *Proceedings of the Sixth International Workshop on Cooperative Information Agents*, S. Ossowski and O. Shehory, Eds. LNAI, vol. 2446. Springer-Verlag, Madrid, Spain, 164–178.

Mui, L., Mohtashemi, M., and Halberstadt, A. 2002. A computational model of trust and reputation. In *Proceedings of the 35th Hawaii International Conference on System Sciences*. Big Island, HI, USA, 188–196.

Mui, L., Szolovits, P., and Ang, C. 2001. Collaborative sanctioning: Applications in restaurant recommendations based on reputation. In *Proceedings of the Fifth International Conference on Autonomous Agents*. ACM Press, Montreal, Canada, 118–119.

Mukherjee, R., Dutta, P., and Sen, S. 2001. MOVIES2GO: A new approach to online movie recommendation. In *Proceedings of the IJCAI Workshop on Intelligent Techniques for Web Personalization*. Seattle, WA, USA.

Nejdl, W. 2004. How to build Google2Google: An incomplete receipe. Invited talk, 3rd International Semantic Web Conference, Hiroshima, Japan.

Newcomb, T. 1961. *The Acquaintance Process*. Holt, Rinehart, and Winston, New York, NY, USA.

Newman, M. 2003. The structure and function of complex networks. *SIAM Review 45,* 2, 167–256.

Nichols, D. 1998. Implicit rating and filtering. In *Proceedings of the Fifth DELOS Workshop on Filtering and Collaborative Filtering*. ERCIM, Budapest, Hungary, 31–36.

Obrst, L., Liu, H., and Wray, R. 2003. Ontologies for corporate Web applications. *AI Magazine 24,* 3, 49–62.

O'Donovan, J. and Smyth, B. 2005. Trust in recommender systems. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*. ACM Press, San Diego, CA, USA, 167–174.

Olsson, T. 1998. Decentralized social filtering based on trust. In *Working Notes of the AAAI-98 Recommender Systems Workshop*. Madison, WI, USA.

Olsson, T. 2003. Bootstrapping and decentralizing recommender systems. Ph.D. thesis, Uppsala University, Uppsala, Sweden.

O'Mahony, M., Hurley, N., Kushmerick, N., and Silvestre, G. 2004. Collaborative recommendation: A robustness analysis. *ACM Transactions on Internet Technology 4,* 3 (August).

Oztekin, U., Karypis, G., and Kumar, V. 2002. Expert agreement and content-based reranking in a meta search environment using Mearf. In *Proceedings of the Eleventh International Conference on World Wide Web*. ACM Press, Honolulu, HI, USA, 333–344.

Page, L., Brin, S., Motwani, R., and Winograd, T. 1998. The PageRank citation ranking: Bringing order to the Web. Tech. rep., Stanford Digital Library Technologies Project.

Papagelis, M., Plexousakis, D., and Kutsuras, T. 2005. Alleviating the sparsity problem of collaborative filtering using trust inferences. In *Proceedings of the 3rd International Conference on Trust Management*. LNCS. Springer-Verlag, Rocquencourt, France.

Pazzani, M. 1999. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review 13,* 5-6, 393–408.

Pescovitz, D. 2003. The best new technologies of 2003. *Business 2.0* 11 (November). Time Inc. Publishing.

Pretschner, A. and Gauch, S. 1999. Ontology-based personalized search. In *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence*. Chicago, IL, USA, 391–398.

Quillian, R. 1968. Semantic memory. In *Semantic Information Processing*, M. Minsky, Ed. MIT Press, Boston, MA, USA, 227–270.

Rapoport, A. 1963. Mathematical models of social interaction. In *Handbook of Mathematical Psychology*, D. Luce, R. Bush, and E. Galanter, Eds. Vol. 2. Wiley, New York, NY, USA.

Reiter, M. and Stubblebine, S. 1997a. Path independence for authentication in large-scale systems. In *Proceedings of the ACM Conference on Computer and Communications Security*. ACM Press, Zürich, Switzerland, 57–66.

Reiter, M. and Stubblebine, S. 1997b. Toward acceptable metrics of authentication. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, Oakland, CA, USA, 10–20.

Resnick, P., Iacovou, N., Suchak, M., Bergstorm, P., and Riedl, J. 1994. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the ACM 1994 Conference on Computer-Supported Cooperative Work*. ACM, Chapel Hill, NC, USA, 175–186.

Resnick, P. and Varian, H. 1997. Recommender systems. *Communications of the ACM 40,* 3, 56–58.

Resnik, P. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*. Montreal, Canada, 448–453.

Resnik, P. 1999. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research 11*, 95–130.

Richardson, M., Agrawal, R., and Domingos, P. 2003. Trust management for the Semantic Web. In *Proceedings of the Second International Semantic Web Conference*. Sanibel Island, FL, USA.

Sankaralingam, K., Sethumadhavan, S., and Browne, J. 2003. Distributed PageRank for P2P systems. In *Proceedings of the Twelfth International Symposium on High Performance Distributed Computing*. Seattle, WA, USA.

Sarwar, B. 2001. Sparsity, scalability, and distribution in recommender systems. Ph.D. thesis, University of Minnesota, Minneapolis, MA, USA.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. 2000a. Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*. ACM Press, Minneapolis, MN, USA, 158–167.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. 2000b. Application of dimensionality reduction in recommender systems. In *ACM WebKDD Workshop*. Boston, MA, USA.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference*. Hong Kong, China.

Sarwar, B., Konstan, J., Borchers, A., Herlocker, J., Miller, B., and Riedl, J. 1998. Using filtering agents to improve prediction quality in the GroupLens research collaborative filtering system. In *Proceedings of the 1998 ACM Conference on Computer-Supported Cooperative Work*. ACM Press, Seattle, WA, USA, 345–354.

Schafer, B., Konstan, J., and Riedl, J. 1999. Recommender systems in e-commerce. In *Proceedings of the First ACM Conference on Electronic Commerce*. ACM Press, Denver, CO, USA, 158–166.

Schein, A., Popescul, A., Ungar, L., and Pennock, D. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press, Tampere, Finland, 253–260.

Sénécal, S. 2003. Essays on the influence of online relevant others on consumers' online product choices. Ph.D. thesis, École des Hautes Études Commerciales, Université de Montréal, Montreal, Canada.

Shardanand, U. and Maes, P. 1995. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*. ACM Press, Denver, CO, USA, 210–217.

Siegel, S. and Castellan, J. 1988. *Non-parametric Statistics for the Behavioral Sciences*, 2nd ed. McGraw-Hill, New York, NY, USA.

Sinha, R. and Swearingen, K. 2001. 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.

Smith, E., Nolen-Hoeksema, S., Fredrickson, B., and Loftus, G. 2003. *Atkinson and Hilgards's Introduction to Psychology*. Thomson Learning, Boston, MA, USA.

Snyder, C. and Fromkin, H. 1980. *Uniqueness: The Human Pursuit of Difference*. Plenum, New York, NY, USA.

Sollenborn, M. and Funk, P. 2002. 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*. LNCS, vol. 2416. Springer-Verlag, Aberdeen, UK, 395–405.

Srikumar, K. and Bhasker, B. 2004. Personalized recommendations in e-commerce. In *Proceedings of the 5th World Congress on Management of Electronic Business*. Hamilton, Canada.

Swearingen, K. and Sinha, R. 2001. Beyond algorithms: An HCI perspective on recommender systems. In *Proceedings of the ACM SIGIR 2001 Workshop on Recommender Systems*. New Orleans, LA, USA.

Terveen, L. and Hill, W. 2001. Beyond recommender systems: Helping people help each other. In *Human-Computer Interaction in the New Millenium*, J. Carroll, Ed. Addison-Wesley, Reading, MA, USA.

Tombs, M. 1997. *Osmotic Pressure of Biological Macromolecules*. Oxford University Press, New York, NY, USA.

Torres, R., McNee, S., Abel, M., Konstan, J., and Riedl, J. 2004. Enhancing digital libraries with TechLens+. In *Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries*. ACM Press, Tuscon, AZ, USA, 228–236.

Twigg, A. and Dimmock, N. 2003. Attack-resistance of computational trust models. In *Proceedings of the Twelfth IEEE International Workshop on Enabling Technologies*. Linz, Austria, 275–280.

van Rijsbergen, K. 1975. *Information Retrieval*. Butterworths, London, UK.

Vogt, C. and Cottrell, G. 1999. Fusion via a linear combination of scores. *Information Retrieval 1,* 3, 151–173.

Watts, D. and Strogatz, S. 1998. Collective dynamics of "small-world" networks. *Nature 393*, 440–442.

Wiese, R., Eiglsperger, M., and Kaufmann, M. 2001. yFiles: Visualization and automatic layout of graphs. In *Proceedings of the 9th International Symposium on Graph Drawing*. LNCS, vol. 2265. Springer-Verlag, Heidelberg, Germany, 453–454.

Ziegler, C.-N. 2004a. Semantic Web recommender systems. In *Proceedings of the Joint ICDE/EDBT Ph.D. Workshop 2004*, W. Lindner and A. Perego, Eds. Crete University Press, Heraklion, Greece.

Ziegler, C.-N. 2004b. Semantic Web recommender systems. In *EDBT 2004 Workshops (PhD, DataX, PIM, P2P&DB, and ClustWeb)*, W. Lindner, M. Mesiti, C. Türker, Y. Tzitzikas, and A. Vakali, Eds. LNCS, vol. 3268. Springer-Verlag, Heraklion, Greece, 78–89. Revised selected papers.

Ziegler, C.-N. and Lausen, G. 2004a. Analyzing correlation between trust and user similarity in online communities. In *Proceedings of the 2nd International Conference on Trust Management*, C. Jensen, S. Poslad, and T. Dimitrakos, Eds. LNCS, vol. 2995. Springer-Verlag, Oxford, UK, 251–265.

Ziegler, C.-N. and Lausen, G. 2004b. Paradigms for decentralized social filtering exploiting trust network structure. In *Proceedings of the DOA/CoopIS/ODBASE Confederated International Conferences (2)*, R. Meersman and Z. Tari, Eds. LNCS, vol. 3291. Springer-Verlag, Larnaca, Cyprus, 840–858.

Ziegler, C.-N. and Lausen, G. 2004c. Spreading activation models for trust propagation. In *Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service*. IEEE Computer Society Press, Taipei, Taiwan.

Ziegler, C.-N. and Lausen, G. 2005. Propagation models for trust and distrust in social networks. *Information Systems Frontiers*. Kluwer Academic Publishers. To appear.

Ziegler, C.-N., Lausen, G., and Schmidt-Thieme, L. 2004. Taxonomy-driven computation of product recommendations. In *Proceedings of the 2004 ACM CIKM Conference on Information and Knowledge Management*. ACM Press, Washington, D.C., USA, 406–415.

Ziegler, C.-N., McNee, S., Konstan, J., and Lausen, G. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International World Wide Web Conference*. ACM Press, Chiba, Japan.

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

Zimmermann, P. 1995. *The Official PGP User's Guide*. MIT Press, Boston, MA, USA.