



Project no. 34721

## **TAGora**

# **Semiotic Dynamics in Online Social Communities**

<http://www.tagora-project.eu>

Sixth Framework Programme (FP6)

Future and Emerging Technologies of the Information Society Technologies (IST-FET Priority)

---

### **D4.4: Review of existing recommendation strategies and systems.**

---

Period covered: from 01/06/2006 to 31/05/2007  
Start date of project: June 1<sup>st</sup>, 2006  
Due date of deliverable: May 31<sup>st</sup>, 2007  
Distribution: Public

Date of preparation: 31/05/2007  
Duration: 36 months  
Actual submission date: May 31<sup>st</sup>, 2007  
Status: Final

Project coordinator: Vittorio Loreto  
Project coordinator organisation name: "La Sapienza" Università di Roma  
Lead contractor for this deliverable: University of Southampton

# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>   | <b>3</b>  |
| <b>2</b> | <b>Recommendation Strategies</b>                                | <b>4</b>  |
| 2.1      | Collaborative Filtering . . . . .                               | 4         |
| 2.1.1    | Collaborative Filtering Algorithms . . . . .                    | 5         |
| 2.1.2    | Evaluation of Collaborative Filtering Techniques . . . . .      | 6         |
| 2.2      | Feature-Based Recommendation . . . . .                          | 6         |
| 2.2.1    | Content-Based Recommendation . . . . .                          | 6         |
| 2.2.2    | Demographic-Based Recommendation . . . . .                      | 7         |
| 2.2.3    | Folksonomy-Based Recommendation . . . . .                       | 8         |
| 2.2.4    | Evaluation of Feature-Based Recommendation Techniques . . . . . | 9         |
| 2.3      | Hybrid Recommendation . . . . .                                 | 10        |
| 2.3.1    | Evaluation of Hybrid Recommendation Techniques . . . . .        | 11        |
| <b>3</b> | <b>Recommender Systems</b>                                      | <b>12</b> |
| 3.1      | Amazon.com . . . . .  | 12        |
| 3.2      | Last.fm . . . . .   | 14        |
| <b>4</b> | <b>Conclusions</b>  | <b>16</b> |

# Chapter 1

## Introduction

Recommendation systems have evolved in recent years to support users in the discovery of new items through the construction of profiles that represent their interests, and networks that connect them to other users who share similar tastes. Many of these recommendation strategies rely on the modeling of salient features that describe each item (e.g. the keywords for a document or the genre of a CD) so that the items can be categorized, and the level of interest a user has can be expressed in terms of these features. This knowledge is usually gathered over time, by monitoring and logging various user interactions with the system (e.g. buying, browsing, bookmarking), as well as explicit ratings by users. Amazon.com, for example, provides a recommendation service that is based on collaborative filtering: if a user buys an item that has been bought by a number of other users in combination with some other items, then those other items will be recommended by Amazon.com to the user. These recommendations are entirely based on what goes on inside the system (Amazon.com in this case), ignorant of any external knowledge about the items or the users themselves.

Many different recommendation strategies have been proposed, exploiting a variety of different data resources and information retrieval techniques. Research has continued with an aim to better represent user's interests as well as ways to express their similarity to other users. In this report, a review and categorization of recommendation strategies is given, including collaborative filtering and feature-based approaches, along with a discussion of their strengths and weaknesses. This is followed by a discussion of hybrid recommendation strategies (i.e. those which combine two or more different techniques) and how they can be used to improve the quality of recommendations. Two popular e-commerce sites (Amazon.com and last.fm) are reviewed, with a description of the recommendation techniques they employ and the problems faced during their implementation. Finally, directions for future research are discussed with an emphasis on exploiting the new opportunities arising through semantic web technology and the widespread use of community driven websites.

## Chapter 2

# Recommendation Strategies

A variety of different techniques have been proposed to support the recommendation of new items to users, and to predict the interest a particular user would have in a previously unrated item. These fundamental techniques can be placed into one of two categories based on the information they acquire about users and items, and the way in which user interests are modeled. *Collaborative filtering* (Resnick et al., 1994) techniques work on the assumption that users who have similar rating habits also share the same preferences for items. This “people-to-people correlation” (Schafer et al., 1999) is frequently used in e-commerce recommender systems (Breese et al., 1998) and drives many of the popular e-commerce sites, such as Amazon.com and last.fm. Other recommendation strategies have been developed by extending research from the field of information retrieval (Belkin and Croft, 1992) and focus on the modeling of features that describe items and users, a category of systems referred to in this report as *feature-based* recommendation. As these strategies evolved, and their weaknesses were identified, *hybrid recommender* systems (Burke, 2002) arose as a method for combining both types of approaches. In this section of the report, an overview of collaborative filtering, feature-based recommendation, and hybrid recommendation strategies is given, along with a discussion of their relative merits and weaknesses.

### 2.1 Collaborative Filtering

Collaborative recommendation is probably the most widely used and extensively studied technique that is founded on one simple premise: if user  $U_1$  is interested in items  $I_1$ ,  $I_2$ , and  $I_3$ , and user  $U_2$  is interested in items  $I_1$ ,  $I_2$ ,  $I_3$ , and  $I_4$ , then it is likely that user  $U_1$  will also be interested in item  $I_4$ . In a collaborative filtering recommender system, the ratings a user assigns to items is used to measure their commonality with other users who have also rated the same items. The degree of interest for an unrated item can then be deduced for a particular user by examining the ratings of their closet neighbors.

Collaborative filtering systems are often categorized in terms of the rating strategy employed (Breese et al., 1998). In systems where users consciously rate items (often called *explicit* voting), their level of interest may be expressed using a discrete value. This could be binary (e.g. like / dislike, good / bad), or a within a range of values (e.g. between 1 and 10). In other cases, the amount of preference for a particular item is derived by examining the user’s past behavior, such as their purchasing or browsing history.

Typically, a collaborative filtering system use a vector space model to represent users, items, and how users have rated items, as illustrated in Figure 2.1. The first collaborative filtering systems, such as GroupLens (Resnick et al., 1994) and Ringo (Shardanand and Maes, 1995), used this type of representation as a foundation for algorithms that predict the rating a user would give to a previously unseen item.

|       |                | Users          |                |                |                |     |                  |
|-------|----------------|----------------|----------------|----------------|----------------|-----|------------------|
|       |                | U <sub>1</sub> | U <sub>2</sub> | U <sub>3</sub> | U <sub>4</sub> | ... | U <sub>i</sub>   |
| Items | I <sub>1</sub> | 4              |                | 5              | 2              |     |                  |
|       | I <sub>2</sub> | 3              | 5              |                | 1              |     | 1                |
|       | I <sub>3</sub> |                | 2              | 4              | 4              |     | 3                |
|       | ...            |                |                |                |                |     |                  |
|       | I <sub>j</sub> | 4              | 4              | 3              |                |     | r <sub>i,j</sub> |

Figure 2.1: Collaborative filtering algorithms typically make use of user votes (or ratings) across a set of items

### 2.1.1 Collaborative Filtering Algorithms

In general, the aim of collaborative filtering is to predict the rating a user would give to a previously unrated item based on the ratings of other users. Given a set of ratings where  $r_{i,j}$  denotes the rating a user  $i$  has given to the item  $j$ , the mean rating ( $\bar{r}_i$ ) for the user  $i$  is defined as follows:

$$\bar{r}_i = \frac{1}{|I_i|} \sum_{j \in I_i} r_{i,j} \quad (2.1)$$

where  $I_i$  is the set of items that user  $i$  has rated. The predicted rating for a user  $x$  (the active user) is then defined as a weighted sum of the ratings of other users in the database:

$$p_{x,j} = \bar{r}_a + \kappa \sum_{i=1}^n w(x,i)(r_{i,j} - \bar{r}_i) \quad (2.2)$$

where  $n$  is the total number of users and  $\kappa$  is a normalization constant. The weight  $w(x,i)$  is used to capture the similarity between users based on commonalities in their rating habits. One popular way to calculate this weighting is to use the *pearson* correlation coefficient (Resnick et al., 1994), the first statistical method defined explicitly for collaborative filtering. The Pearson correlation coefficient is defined as follows:

$$w(x,i) = \frac{\sum_j (r_{x,j} - \bar{r}_x)(r_{i,j} - \bar{r}_i)}{\sqrt{\sum_j (r_{x,j} - \bar{r}_x)^2 \sum_j (r_{i,j} - \bar{r}_i)^2}} \quad (2.3)$$

With this similarity measure, users receive the highest weighting when compared to other users who have rated the same items with the same values. This method has been used successfully in many collaborative filtering systems (Goldberg et al., 1992; Hill et al., 1995; Shardanand and Maes, 1995) and performs well in many domains.

Another common way to define a user similarity weighting is to adapt measures from the field of information retrieval (Salton and McGill, 1986). If the ratings each user has applied to items in the database is considered to be a vector, the similarity between user can be derived by calculating the cosine of the angle formed by the two rating vectors:

$$w(x,i) = \sum_j \frac{r_{x,j}}{\sqrt{\sum_{k \in I_x} r_{x,k}^2}} \frac{r_{i,j}}{\sqrt{\sum_{k \in I_i} r_{i,k}^2}} \quad (2.4)$$

where the squared terms in the denominator are used to normalize ratings so that users who have many ratings will not automatically receive high weightings.

### 2.1.2 Evaluation of Collaborative Filtering Techniques

Collaborative filtering techniques work well for complex domains, such as movie and music recommendation, because they focus on the similarities between users often provide good cross-genre recommendations. For example, a collaborative filtering system for music recommendation can identify that progressive rock listeners may also enjoy fusion jazz. A feature-based system (described below in Section 2.2) could not find this kind of connection because the items (in this case albums or artists) do not share common features (e.g. composers, performers, etc.). However, this approach does not work well for users who have unique tastes and do find themselves in a clique of other neighbors, so called “grey sheep” (Claypool et al., 1999).

A prominent problem associated with collaborative filtering systems arises when new items and new users are added to the system - commonly referred to as the *ramp-up* problem (Konstan et al., 1998). Since collaborative filtering algorithms rely on ratings to compare users, a new user with no ratings will have a neutral profile. When new items are added to the system, they will not be recommended until some users have rated them.

Another major factor affecting the success of collaborative filtering systems is the relative density of user ratings to the number of items. Because collaborative filtering depends on the overlap in ratings across users, they perform badly when ratings are sparse (i.e. few users have rated the same items) because it is hard to find similar neighbors. Solutions to these problems have been proposed (Foltz, 1990; Rosenstein and Lochbaum, 2000) using mathematical techniques such as dimensionality reduction.

To summarize, collaborative filtering techniques are good at finding cross-genre recommendations and do not require extensive domain knowledge modeling. Over time, the quality of recommendations increases as more users are added to the system and the number of ratings increases. However, the new item and new user ramp-up problem poses a significant handicap, particularly for new users because little utility can be harnessed from the system until a significant amount of ratings have been made.

## 2.2 Feature-Based Recommendation

Feature-Based recommendation refers to a category of recommender systems that evolved out of information retrieval research. Rather than focusing on the correlation of ratings between users, feature-based approaches attempt to model the attributes that define and categorize items so that user interests can be modeled and the most relevant items matched.

### 2.2.1 Content-Based Recommendation

When considering a recommendation system that operates over a set of items (such as documents, books, or movies), each item can be described by a number of *features*. For example, documents can be described by a set of keywords, or movies by their genre. This concept is illustrated in Figure 2.2 where a matrix of users and their ratings of items is shown with an additional space reserved to express the features the item. With this kind of meta data in place, recommender systems can build profiles for each user that express their interests in terms of the features associated with the items they rate most highly.

NewsWeeder (Lang, 1995) was developed in the mid 1990's to provide a filtering system for articles posted on Usenet (a distributed Internet discussion system), and is one of the first recommender systems to employ a content-based approach. As the Internet was gaining momentum as a communication platform, mechanisms to sift through the information overload became an important research interest. The NewsWeeder Web interface provides a method for users to view articles from particular topics. In addition to this conventional newsgroups interface, virtual newsgroups

|       |                | Users          |                |                |                |     |                  | Item Features  |                |                |                |                |
|-------|----------------|----------------|----------------|----------------|----------------|-----|------------------|----------------|----------------|----------------|----------------|----------------|
|       |                | U <sub>1</sub> | U <sub>2</sub> | U <sub>3</sub> | U <sub>4</sub> | ... | U <sub>i</sub>   | F <sub>1</sub> | F <sub>2</sub> | F <sub>3</sub> | F <sub>4</sub> | F <sub>5</sub> |
| Items | I <sub>1</sub> | 4              |                | 5              | 2              |     |                  | 0.7            | 0.3            | 0.1            | 0.2            | 0.3            |
|       | I <sub>2</sub> | 3              | 5              |                | 1              |     | 1                | 0.5            | 0.3            | 0.8            | 0.9            | 0.4            |
|       | I <sub>3</sub> |                | 2              | 4              | 4              |     | 3                | 0.5            | 0.1            | 0.6            | 0.9            | 0.8            |
|       | ...            |                |                |                |                |     |                  |                |                |                |                |                |
|       | I <sub>j</sub> | 4              | 4              | 3              |                |     | r <sub>i,j</sub> | 0.3            | 0.9            | 0.6            | 0.9            | 0.4            |

Figure 2.2: Content-Based algorithms use features of the items to generate user profiles that express users preference in terms of item features.

are created for each user containing 50 recommended articles based on the preferences already learned by the system. After reading an entry, a NewsWeeder user rates an article from 1 to 5 to denote their interest in the document and the system records their ratings.

To automatically build a set of features that describe the content of articles posted on UseNet, a text mining method called *term frequency - inverse document frequency* (TF-IDF) (Salton, 1988) was used. TF-IDF is a statistical measure used to express how important terms are in document with respect to the whole collection of documents (or *corpus*). The measure of importance for term within a document increases proportionally to the number of times the term appears, but is offset against the total number of times that term appears in the whole corpus. This means that common words (such as “the”, “of”, and “to”) do not receive high importance because they are used frequently in all documents, and less frequently used words have a higher degree of importance. The rationale behind this approach is that less frequently used words that appear often in a particular set of documents are likely to reflect a strong correlation in their content. For a given document  $j$ , the weighting of a term  $t$  is defined as follows:

$$w(j, t) = tf_{j,t} \times \log_2 \frac{N}{n} \quad (2.5)$$

where  $tf_{j,t}$  is the frequency of term  $t$  in Document  $j$ ,  $N$  is the number of documents in the corpus, and  $n$  is the number of documents where term  $t$  occurs at least once.

By using a TF-IDF approach, text from the UseNet articles is analyzed and used to create a vector describing the salient terms in the article. As the user rates more articles, their profile is modified to reflect the terms in the documents they rate highest. When a new document is added to the system, a user's profile can be compared to feature vector of the new article (using a method such as the cosine correlation defined earlier in Equation 2.4) to predict their rating. This method of comparison allows the NewsWeeder system to periodically build a list of the articles it believes to be most interesting for each user.

## 2.2.2 Demographic-Based Recommendation

Demographic-Based recommendation techniques evolved as a way to direct users to the items they are most likely to be interested in based on their personal attributes. By recording information about the users, such as their age, gender, and location, recommendation strategies have been developed to categorize users under the assumption that users who share demographic attributes also share similar tastes. The Grundy system (Rich, 1998) is an book recommendation system that illustrates the importance of demographic-based recommendation by proposing the following scenario:

|               |       | Users |       |       |       |     |           |
|---------------|-------|-------|-------|-------|-------|-----|-----------|
|               |       | $U_1$ | $U_2$ | $U_3$ | $U_4$ | ... | $U_i$     |
| Items         | $I_1$ | 4     |       | 5     | 2     |     |           |
|               | $I_2$ | 3     | 5     |       | 1     |     | 1         |
|               | $I_3$ |       | 2     | 4     | 4     |     | 3         |
|               | ...   |       |       |       |       |     |           |
|               | $I_j$ | 4     | 4     | 3     |       |     | $r_{i,j}$ |
| User Features | $D_1$ | 0.7   | 0.3   | 0.1   | 0.2   |     | 0.3       |
|               | $D_2$ | 0.5   | 0.3   | 0.8   | 0.9   |     | 0.9       |
|               | $D_3$ | 0.5   | 0.1   | 0.6   | 0.9   |     | 0.3       |
|               | $D_4$ | 0.3   | 0.4   | 0.8   | 0.8   |     | 0.8       |
|               | $D_5$ | 0.3   | 0.9   | 0.6   | 0.9   |     | 0.1       |

Figure 2.3: Demographic-Based algorithms use features of the users (e.g. demographic information) to identify other users who share similar tastes.

*Someone walks into a large library, tells the librarian that he is interested in China, and asks for some books. What sort of books does the librarian recommend? That depends. Is the person a small child who just saw a TV show about China and wants to see more pictures of such an exotic place? Is the person a high school student doing a term paper? Or maybe a prospective tourist? Or a scholar interested in Eastern thought? Can the person read Chinese? The librarian needs to know these things before he can point the reader to the right books. Some of what he needs to know he'll know before he even thinks about it, such as the approximate age of the person. Some things he'll assume until he has evidence to the contrary, such as that the person does not read Chinese. To find out other things, he'll ask a few specific questions. Only after he has a rough model of the person he's talking to can he answer the question.*

In terms of modeling these user features, vector space models can be expanded, as Figure 2.3 depicts, to include a demographic dimension. The Lifestyle Finder (Krulwich, 1997) is an example of another demographic-based recommender system that is used to recommend a range of products and service by surveying users according to demographic groups from marketing research.

### 2.2.3 Folksonomy-Based Recommendation

Traditional feature-based recommender systems often rely on pre-defined attributes specified by the system architect, but these may not be sufficient or fully encapsulate the features that reflect user interest. Therefore, folksonomies (Vander Wal, 2005), which are organic structures that arise through collaborative tagging (Gruber, 2006), have been proposed (Szomszor et al., 2007) as a method for capturing the perceptions users have of resources, and therefore provide a foundation for the expression of users' interests. The ITNG '06 Submissions recommender system (Niwa et al., 2006) investigated folksonomy based recommendation in the context of social bookmarking



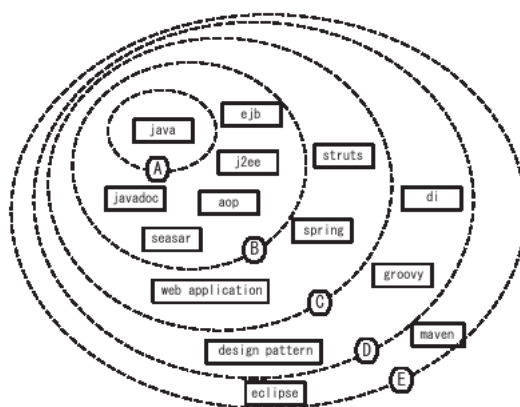


Figure 2.4: Tag Clusters from del.icio.us

to build a Web Page recommender system. Niwa *et al* cluster terms from the del.icio.us<sup>1</sup> folksonomy so the affinity between a user and a particular tag can be calculated and used to express their areas of interest. Figure 2.4 is an illustration of the terms associated with the `java` tag. Levels **A** to **E** denote different levels of abstraction, highlighting the fact that different users with different levels of expertise in a domain (such as programming in this case) use different terms to annotate a resource.

## 2.2.4 Evaluation of Feature-Based Recommendation Techniques

Content-Based recommendation systems also suffer from the new user ramp-up problem because they must accumulate enough ratings for each user to generate an accurate profile that fully represents their areas of interest. However, new items are not subject to a ramp-up problem because the features to describe can be automatically generated using techniques such as latent semantic indexing (Foltz, 1990), and text mining (van Meteren and van Someren, 2000). In cases where a content-based approach is based on a set of pre-conceived features, the accuracy of recommendations is proportional to quality of domain modeling: if the system designer does not choose to capture the appropriate features of the items, they may not reflect the concepts that distinguish users' preferences. Folksonomy-Based recommendation has been proposed as a way to alleviate this problem because the tags that describe items are created by users and therefore reflect their perceptions. Clustering of the terms used in folksonomies has also revealed interesting hierarchies that mirror the level of understanding a user has. For example, the tags used to describe a web page about programming may be very general (e.g. `java`), or more specific (e.g. `J2EE` and `EJB`). These levels of granularity reflect the level of understanding the user has, and therefore, provide a good measure for the suitability of other articles.

Both content-based and collaborative filtering system must consider the compromise between stability and plasticity. Once the system has learned the preference of a particular user, it is hard to unlearn them. Sometimes, important features about a users' preferences change, such as becoming a vegetarian, that would drastically effect the quality of recommendations made. Time-based discounting of ratings (Billsus and Pazzani, 2000; Schwab et al., 2001) has been proposed as a may to address this problem, but these method can loose information about long-term, but sporadically exercised preferences.

<sup>1</sup>[www.del.icio.us](http://www.del.icio.us)

## 2.3 Hybrid Recommendation

Collaborative filtering and content-based recommendation systems are not without their deficiencies. Since both strategies rely on ratings to build a user's profile of interest, new users with no ratings have neutral profiles and cannot receive personalized recommendations. When new items are added to a collaborative filtering recommender system, they will not be recommended until some users have rated them. Collaborative systems also depend on the overlap in ratings across users and perform badly when ratings are sparse (i.e. few users have rated the same items) because it is hard to find similar neighbors.

Hybrid recommender systems, i.e. those which make use of collaborative and feature-based approaches, have been developed to overcome some of these problems. For example, collaborative recommender systems do not perform well with respect to items that have not been rated, but content-based methods can be used to understand their relationship to other items through a set of common attributes. Hence, a mixture of the two methods can be used to provide more robust recommender systems.

The way in which a hybrid recommender system combines different recommendation strategies can be used as a basis for their categorization and comparison (Burke, 2002). These categories are:

- **Weighted**

A weighted approach describes a class of recommendation systems that sum the output from two or more different recommendation techniques to produce a single prediction. The P-Tango system (Claypool et al., 1999) is an example of a weighted hybrid recommender system where collaborative filtering and content-based recommendations are given equal weighting. Pazzani (Pazzani, 1999) also proposes a weighted hybrid system making use of collaborative filtering, content-based, and demographic-based recommendation. The output from each strategy is used as a vote for the final prediction based on the confidence behind the rating.

- **Switching**

A recommender system can choose to use one particular recommendation technique over another given the current state of the system. For example, the DailyLearner system (Pazzani and Billsus, 2000) attempts content-based recommendation first. If few recommendations are found, a collaborative filtering algorithm is used.

- **Mixed**

The output from multiple recommendation strategies may be presented to the user simultaneously. The Personalized TV recommender system (PTV) (Smyth and Cotter, 2000) uses content-based and collaborative filtering techniques to construct a personalized television viewing timetable. ProfBuilding (Wasfi, 1999) and Pick A Flick (Burke et al., 1997) are also examples of a mixed hybrid recommender system.

- **Cascade**

Different recommendation strategies can be combined by taking a set of candidate recommendations from one algorithm and passing them to as input to another for further refinement. The EntreeC system (Burke, 2002) is an example of such a system where the output from a feature-based algorithm passed as input to a collaborative filtering algorithm.

- **Feature Augmentation**

Some recommendation systems produce an output that provides additional information about a particular item. For example, Amazon.com produce a list of related authors and related titles for a given book. These additional features can then be used as input to another recommendation algorithm. The Libra system (Mooney and Roy, 2000) and GroupLens (Sarwar

et al., 1998) have both used this technique to enhance collaborative filtering recommendation.

- **Meta-Level**

Another way to combine two recommendation strategies is to use the model generated by one technique as the input to another. While similar to the feature augmentation method described above, meta-level combination uses the entire model produced and not just the additional features it generates. For example, a content-based technique will usually create a user profile that lists the terms or features that a user is most interested in. This profile can then be passed as input to a collaborative filtering algorithm to find other users who share similar interests, an approach first used by the Fab system (Balabanovic and Shoham, 1997) to recommend online articles.

It is often the case that hybrid recommendation systems also make use of some inferencing technology to aid in the recommendation process (Middleton et al., 2004; Towle and Quinn, 2000). For example, the EntreeC system provides recommendations for restaurants according to a set of preferences expressed by the user. If they specify a desire for a restaurant in a romantic location, this preference would be expanded to a set of requirements such as a quiet location, intimate atmosphere, and good views.

### 2.3.1 Evaluation of Hybrid Recommendation Techniques

While a weighted hybridization is straight forward to implement, it makes a simplifying assumption that the output from each technique is given equal credence. Since collaborative filtering systems are sensitive to the amount of user ratings and the density of user ratings to items, it will not always produce predictions with equal confidence to that of another approach. A switching based combination of recommendation techniques can overcome this problem because the most appropriate method is used in any given context. However, switching is difficult to implement because complex parameters must be analyzed in order to choose the most appropriate recommendation strategy.

A mixed hybridization is good when many different predictions are required, such as the PTV system, where an entire television viewing schedule must be produced. In this context, a recommender system must cater for situations where conflicting suggestions are generated for a particular time. A cascading method can be more computationally efficient than a weighted approach because each recommendation algorithm does not need to be run over a full set of users and items. This is an important consideration for a recommender system that operates over a very large set of users and items.

In situations where different recommender systems are integrated, a feature augmentation approach is often the most practical, particularly in scenarios where the inner working of one system are not known and cannot be adjusted. Like the cascading method, meta-level hybridization provides a good way to improve the computational efficiency of collaborative filtering methods because rather than trying to reason on the ratings of all users and all items, user similarity can be computed by comparing the user profile generated by a feature-based approach.

Whenever inferencing mechanisms are introduced, for example through knowledge based approaches, additional and often expensive domain modeling is required. However, this additional cost does give a high reward with respect to the quality of recommendations and can reduce the amount of previous rating behavior required to build user profiles.

## Chapter 3

# Recommender Systems

Prior to the advent of the internet, and its subsequent success as a market place to trade goods and services, the variety of products made available to users was limited by physical constraints: the number of items offered to shoppers was dictated by the size of a shop or the volume of a product catalog. As the internet matured and connectivity became more widespread, it was quickly realized that virtual shops could provide consumers with a larger range of products. Even though the largest book store may hold around 200,000 items, an online book store can exceed this limit easily, potentially supplying millions of titles. As the range of products made available increases, so to does the amount of information presented to consumers, requiring more ingenious ways to point customers to the products they desire.

Recommendation strategies, such as collaborative filtering, have proven themselves to be a good solution to the problem of information overload, supporting customers in the location of products and the discovery of new content while also considering their own personal requirements. However, deployment of such technologies in the real world reveals other problems that effect their feasibility, such as the stability of the system, computational complexity, and privacy of data. In this section of the report, an overview of two popular online recommendation systems is given; one that provides customers with a method of finding books, and another for the recommendation and delivery of music content.

### 3.1 Amazon.com

Amazon.com were one of the first companies to sell products over the internet, supplying much of the momentum that drove the dot-com boom in the late 1990s. As their product catalog grew and the range of items offered became more diverse, new tools and techniques were developed to help customers find the products they want. By using recommendation algorithms, Amazon.com create personalized shopping interfaces for each customer based on their past buying behavior and the buying behavior of other customers. One key feature of this interface is the ability to take a product a customer is interested in purchasing and recommend similar products to them. Figure 3.1 provides an example of this feature where the book *War and Peace* is shown along with a selection of related books (*Crime and Punishment*, *Anna Karenina*, *The Count of Monte Cristo*, etc). This recommendation is generated by listing the products that previous customers have bought in conjunction with *War and Peace*, a technique referred to my Amazon.com as item-to-item collaborative filtering.

While traditional collaborative filtering techniques compute a product-to-product similarity matrix by iterating through all item pairs, this technique is not practical for Amazon.com because many items do not have common customers. Instead, the customer centric algorithm (Linden et al., 2003) listed in Figure 3.2 is used. To compute the similarity between two items, a cosine measure is used across each item vector, where the item vector contains a list of other products that have

**People who bought this item...**



**War and Peace (Wordsworth Classics) (Paperback)**  
by L.N. Tolstoy (Author), Louise Maude (Translator), Aylmer Maude (Translator)  
Average Customer Review: ★★★★★ (35)  
Usually dispatched within 4 to 6 weeks  
Eligible for **FREE UK delivery** on orders over £15 with Super Saver Delivery. [See details and conditions](#)

**Book Description**  
In Russia's struggle with Napoleon, Tolstoy saw a tragedy that involved all mankind. Greater than a historical chronicle, War and Peace is an affirmation of life itself, 'a complete picture', as a contemporary reviewer put it, 'of everything in which people find their happiness and greatness, their... [Read More](#)

**£1.99**

[Add to Basket](#) [Add to Wish List](#)

65 used & new from £0.01

**Also bought these items...**



**Crime and Punishment (Penguin Popular Classics) Paperback** by Fyodor Dostoyevsky  
[More like this](#)



**Anna Karenina (Wordsworth Classics) Paperback** by L.N. Tolstoy  
[More like this](#)



**Crime and Punishment (Wordsworth Classics) Paperback** by F.M. Dostoyevsky  
[More like this](#)



**The Idiot (Wordsworth Classics) Paperback** by F.M. Dostoyevsky  
[More like this](#)



**The Count of Monte Cristo (Wordsworth Classics) Paperback** by Alexandre Dumas  
[More like this](#)



**Don Quixote (Wordsworth Classics) Paperback** by Miguel De Cervantes Saavedra  
[More like this](#)

Figure 3.1: The Amazon Similar products interface

been purchased by customers at the same time. Given that the Amazon catalog holds several million items, and the number of Amazon customers is in the region of 29 million, calculating item-to-item similarity is an expensive and time consuming process. However, this process can be performed offline (for example overnight): the algorithm's online component (i.e. looking up similar items) runs efficiently in real-time and scales independently of the catalog size or the total number of customers.

With this recommendation technique, customer purchasing history is used as input to the recommendation algorithm, but it is hidden from other users. Therefore, the recommendation system benefits from understanding the relationship between different customers buying habits while retaining a level of privacy from the customer's perspective.

```

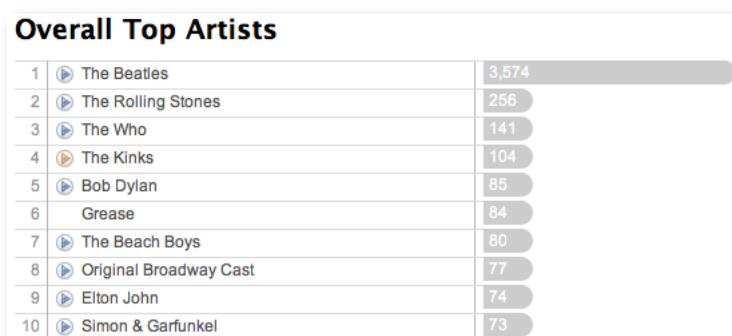
For each item in product catalog,  $I_1$ 
  For each customer  $C$  who purchased  $I_1$ 
    For each item  $I_2$  purchased by customer  $C$ 
      Record that a customer purchased  $I_1$  and  $I_2$ 
    For each item  $I_2$ 
      Compute the similarity between  $I_1$  and  $I_2$ 

```

Figure 3.2: The algorithm used by Amazon to calculate item-item distance.

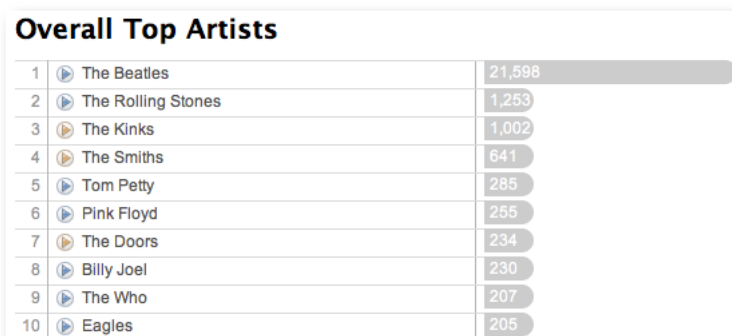
## 3.2 Last.fm

Last.fm is an internet radio station and music recommendation system headlined with the slogan *the social music revolution*. By building up a profile of musical interest for each user by recording the tracks they have listened to, either through the Last.fm radio player or from their own music collection (using plugins from the Audioscrobbler<sup>1</sup> project), users can be connected to their musical neighbors - people who share similar music tastes. For example, Figure 3.3 and Figure 3.4 show the top 10 artists for two users (denoted as User A and User B) and the number of times tracks by each artist have been played. In this example, User A and User B share many of the same favorite artists (namely The Beatles, The Rolling Stones, The Who, and The Kinks). By using a collaborative filtering algorithm, Last.fm can exploit this relationship between users to provide recommendations for new artists by examining the top artists of a user's musical neighbors.



| Overall Top Artists |                        |       |
|---------------------|------------------------|-------|
| 1                   | The Beatles            | 3,574 |
| 2                   | The Rolling Stones     | 256   |
| 3                   | The Who                | 141   |
| 4                   | The Kinks              | 104   |
| 5                   | Bob Dylan              | 85    |
| 6                   | Grease                 | 84    |
| 7                   | The Beach Boys         | 80    |
| 8                   | Original Broadway Cast | 77    |
| 9                   | Elton John             | 74    |
| 10                  | Simon & Garfunkel      | 73    |

Figure 3.3: Last.fm Top Artists for User A



| Overall Top Artists |                    |        |
|---------------------|--------------------|--------|
| 1                   | The Beatles        | 21,598 |
| 2                   | The Rolling Stones | 1,253  |
| 3                   | The Kinks          | 1,002  |
| 4                   | The Smiths         | 641    |
| 5                   | Tom Petty          | 285    |
| 6                   | Pink Floyd         | 255    |
| 7                   | The Doors          | 234    |
| 8                   | Billy Joel         | 230    |
| 9                   | The Who            | 207    |
| 10                  | Eagles             | 205    |

Figure 3.4: Last.fm Top Artists for User B

Like Amazon.com, last.fm face a problem of computational feasibility when implementing a collaborative filtering algorithm: it is not practical to calculate a user's neighbors in real-time. Instead, the collaborative filtering algorithm is run offline on a weekly basis and the results are cached for the upcoming week. In terms of privacy, last.fm is an open system: the listening habits of users are exposed for other users to view freely. However, much of the success of last.fm has been attributed to fact that users can browse the profiles of the neighbors manually to see what they listen to.

<sup>1</sup>audioscrobbler.net



## Chapter 4

# Conclusions

This report has demonstrated that recommender systems employ a variety of different techniques to provide users with personalized recommendations. Collaborative filtering and feature-based methods both provide successful mechanisms to deliver new content and assist users in the discovery of resources they will find the most interesting. Hybrid recommender systems that combine both these approaches have furthered the development of recommendation strategies and provided ways to overcome the problems of computational feasibility associated with collaborative filtering algorithms and the weaknesses that occur when new items and new users are added to the system.

Through this investigation, it is apparent that most recommender systems are limited to operating over the information they have collected. For example, Amazon.com only considers the buying behavior of its own customers even though other resources, such as Wikipedia<sup>1</sup>, provide extensive information on authors, book titles, and how they are related. With the new levels of data interoperability offered by semantic web technology, and the free exchange of data associated with Web 2.0 sites such as last.fm, new opportunities have arisen to extend the personalization of recommendations through the collection of data about items and individuals from multiple resources.

In the future, traditional information resources, such as database, could be collected and used in conjunction with other data structures, such as folksonomies, to investigate how the two can coexist and be used to better understand both the items, and how users perceive them. With this kind of data, it will also be possible to research the correlation in user tastes across different domains. For example, music interests may be useful when recommending movies to customers, and viceversa.

---

<sup>1</sup>[www.wikipedia.org](http://www.wikipedia.org)



## Bibliography

Marko Balabanovic and Yoav Shoham. Fab: content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, 1997. ISSN 0001-0782. doi: <http://doi.acm.org/10.1145/245108.245124>.

Nicholas J. Belkin and Bruce B. Croft. Information filtering and information retrieval: two sides of the same coin? *Commun. ACM*, 35(12):29–38, December 1992. ISSN 0001-0782. URL <http://portal.acm.org/citation.cfm?id=138861>.

Daniel Billsus and Michael J. Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2-3):147–180, 2000. ISSN 0924-1868. URL <http://www.ics.uci.edu/~pazzani/Publications/BillsusA.pdf>.

John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. pages 43–52, 1998. URL [citeseer.ist.psu.edu/breese98empirical.html](http://citeseer.ist.psu.edu/breese98empirical.html).

Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002. ISSN 0924-1868. doi: <http://dx.doi.org/10.1023/A:1021240730564>.

Robin D. Burke, Kristian J. Hammond, and Benjamin C. Young. The findme approach to assisted browsing. *IEEE Expert*, 12(4):32–40, 1997. URL [citeseer.ist.psu.edu/burke97findme.html](http://citeseer.ist.psu.edu/burke97findme.html).

M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. Combining content-based and collaborative filters in an online newspaper. In *In Proceedings of ACM SIGIR Workshop on Recommender Systems*, 1999. URL [citeseer.ist.psu.edu/claypool99combining.html](http://citeseer.ist.psu.edu/claypool99combining.html).

P. W. Foltz. Using latent semantic indexing for information filtering. In *Proceedings of the conference on Office information systems*, pages 40–47, New York, NY, USA, 1990. ACM Press. URL <http://portal.acm.org/citation.cfm?id=91486>.

David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, December 1992. ISSN 0001-0782. URL <http://dx.doi.org/10.1145/138859.138867>.

Thomas Gruber. Ontology of folksonomy: A mash-up of apples and oranges, 2006. URL <http://tomgruber.org/writing/ontology-of-folksonomy.htm>.

Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. Recommending and evaluating choices in a virtual community of use. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 194–201, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co. ISBN 0201847051. URL <http://portal.acm.org/citation.cfm?id=223904.223929>.

- J. A. Konstan, J. Reidl, A. Borchers, and J.L. Herlocker. Recommender systems: A groupLens perspective. In *Recommender Systems: Papers from the 1998 Workshop (AAAI Technical Report WS-98-08)*, pages 60–64. AAAI Press, 1998.
- B Krulwich. Lifestyle finder: intelligent user profiling using large-scale demographic data. *Artificial Intelligence Magazine*, 18(2):37–45, 1997.
- Ken Lang. NewsWeeder: learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1995. URL [citeseer.ist.psu.edu/lang95newsweeder.html](http://citeseer.ist.psu.edu/lang95newsweeder.html).
- Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003. ISSN 1089-7801. doi: <http://dx.doi.org/10.1109/MIC.2003.1167344>.
- Stuart E. Middleton, Nigel R. Shadbolt, and David C. De Roure. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.*, 22(1):54–88, 2004. ISSN 1046-8188. doi: <http://doi.acm.org/10.1145/963770.963773>.
- Raymond J. Mooney and Lorie Roy. Content-based book recommending using learning for text categorization. In *Proceedings of DL-00, 5th ACM Conference on Digital Libraries*, pages 195–204, San Antonio, US, 2000. ACM Press, New York, US. URL <http://citeseer.ist.psu.edu/222628.html>.
- Satoshi Niwa, Takuo Doi, and Shinichi Honiden. Web page recommender system based on folksonomy mining for itng '06 submissions. In *ITNG '06: Proceedings of the Third International Conference on Information Technology: New Generations (ITNG'06)*, pages 388–393, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2497-4. doi: <http://dx.doi.org/10.1109/ITNG.2006.140>.
- Michael J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408, 1999. URL [citeseer.ist.psu.edu/pazzani99framework.html](http://citeseer.ist.psu.edu/pazzani99framework.html).
- Michael J. Pazzani and Daniel Billsus. Adaptive personalization for the mobile web. In *9th International World Wide Web Conference, Mobile Web Track*, Amsterdam, 2000.
- Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, New York, NY, USA, 1994. ACM Press. ISBN 0897916891. URL <http://portal.acm.org/citation.cfm?id=192905>.
- Elaine Rich. User modeling via stereotypes. In *Readings in intelligent user interfaces*, pages 329–342. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998. ISBN 1-55860-444-8.
- Mark Rosenstein and Carol Lochbaum. Recommending from content: preliminary results from an e-commerce experiment. In *CHI '00: CHI '00 extended abstracts on Human factors in computing systems*, pages 291–292, New York, NY, USA, 2000. ACM Press. ISBN 1-58113-248-4.
- G. Salton and J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc, New York, NY, USA, 1986.
- Gerald Salton, editor. *Automatic text processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1988. ISBN 0-2-1-1227-8.

- Badrul M. Sarwar, Joseph A. Konstan, Al Borchers, Jonathan L. Herlocker, Bradley N. Miller, and John Riedl. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In *Computer Supported Cooperative Work*, pages 345–354, 1998. URL <http://citeseer.ist.psu.edu/sarwar98using.html>.
- J. Ben Schafer, Joseph A. Konstan, and John Riedl. Recommender systems in e-commerce. In *ACM Conference on Electronic Commerce*, pages 158–166, 1999. URL [citeseer.ist.psu.edu/benschafer99recommender.html](http://citeseer.ist.psu.edu/benschafer99recommender.html).
- I. Schwab, A. Kobsa, and I. Koychev. Learning user interests through positive examples using content analysis and collaborative filtering. Technical report, Fraunhofer Institute for Applied Information Technology, Germany, 2001. URL [citeseer.ist.psu.edu/schwab01learning.html](http://citeseer.ist.psu.edu/schwab01learning.html).
- Upendra Shardanand and Patti Maes. Social information filtering: Algorithms for automating “word of mouth”. In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1, pages 210–217, 1995. URL <http://citeseer.ist.psu.edu/shardanand95social.html>.
- Barry Smyth and Paul Cotter. A personalized television listings service. *Commun. ACM*, 43(8): 107–111, August 2000. ISSN 0001-0782. URL <http://portal.acm.org/citation.cfm?id=345161>.
- Martin Szomszor, Ciro Cattuto, Harith Alani, Kieron O'Hara, Andrea Baldassarri, Vittorio Loreto, and Vito D.P. Servedio. Folksonomies, the semantic web, and movie recommendation. In *Workshop on Bridging the Gap between Semantic Web and Web 2.0, European Semantic Web Conference (ESWC)*, Innsbruck, Austria, 2007.
- Brendon Towle and Clark Quinn. Knowledge based recommender systems using explicit user models. In *Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, AAAI Technical Report*, pages 74–77, Menlo Park, CA:AAAI Press, 2000. URL [citeseer.ist.psu.edu/327756.html](http://citeseer.ist.psu.edu/327756.html).
- Robin van Meteren and Maarten van Someren. Using content-based filtering for recommendation. In *Proceedings of MLnet / ECML2000 Workshop*, Barcelona, Spain, May 2000. URL <http://citeseer.ist.psu.edu/499652.html>.
- T. Vander Wal. Folksonomy definition and wikipedia, November 2005. URL <http://vanderwal.net/random/entrysel.php?blog=1750>.
- A. M. Wasfi. Collecting user access patterns for building user profiles and collaborative filtering. In *IUI '99: Proceedings of the 4th international conference on Intelligent user interfaces*, pages 57–64, New York, NY, USA, 1999. ACM Press. ISBN 1581130988. doi: 10.1145/291080.291091. URL <http://portal.acm.org/citation.cfm?id=291091>.